Date: September 2007

Semantics of Business Vocabulary and Business Rules (SBVR)

Beta 3 Document

(with change bars against the first SBVR Interim Specification)

dtc/07-06-05

OBJECT MANAGEMENT GROUP

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

LICENSES

PATENTS

GENERAL USE RESTRICTIONS

DISCLAIMER OF WARRANTY

RESTRICTED RIGHTS LEGEND

TRADEMARKS

COMPLIANCE

OMG’s Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http://www.omg.org*, under Documents, Report a Bug/ Issue (http://www.omg.org/technology/agreement.htm).

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A catalog of all OMG Specifications Catalog is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

OMG Modeling Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications.

OMG Middleware Specifications

- CORBA/IIOP
- IDL/Language Mappings
- Specialized CORBA specifications
- CORBA Component Model (CCM).

Platform Specific Model and Interface Specifications

- CORBAservices
- CORBAfacilities

- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications.

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. (as of January 16, 2006) at:

OMG Headquarters
140 Kendrick Street
Building A, Suite 300
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/technology/agreement.htm*.

Part I - Introduction

This part includes Scope, Conformance, Normative References, Terms and Definitions, Symbols, and Additional Information.

1 Scope

Issue 9955 Replace text

This specification defines the vocabulary and rules for documenting the semantics of business vocabularies, business facts, and business rules; as well as an XMI schema for the interchange of business vocabularies and business rules among organizations and between software tools.

This specification is interpretable in predicate logic with a small extension in modal logic. This specification supports linguistic analysis of text for business vocabularies and rules, with the linguistic analysis itself being outside the scope of this specification.

This specification is applicable to the domain of business vocabularies and business rules of all kinds of business activities of all kinds of organizations. It is conceptualized optimally for business people rather than automated rules processing, and is designed to be used for business purposes, independent of information systems designs.

This specification is applicable as input to transformations by IT staff into information system designs, using a combination of decisions from system architects and Platform Independent Model designers together with software tool function.

2 Conformance

Issue 9957 Replace Clause 2

This specification defines conformance for an SBVR exchange document, for software that produces SBVR exchange documents, and for software that processes SBVR exchange documents.

Conformance of software is defined in terms of:

- the nature of its use of SBVR
- its support for SBVR concepts that are defined in Clauses 8, 9, 11, and 12 of this specification.

2.1 Support for an SBVR Concept

A software tool supports an SBVR concept if and only if all of the following hold:

- The software tool uses the representations specified in Clause 15 for that concept in any SBVR exchange document it produces. It may use other representations of the same concept for other purposes, including other forms of exchange documents.
- The software tool interprets the specified representation of the concept as having the meaning given by the Definition of that concept in this specification, and interprets instances of the concept as having the associated characteristics.
- No Necessity concerning that concept that is given in this specification is violated by any fact in any fact model maintained by the software tool nor in any SBVR exchange document it produces.

NOTE: The requirement to interpret an instance as having the associated characteristics should not be interpreted to require a conforming processor to use any elaborate reasoning to determine characteristics that may be implied by the facts provided, even when those implications are stated as Necessities in SBVR. The intent of the requirement is that what the tool does with the instance is consistent with the SBVR interpretation of the facts provided.

Use of Reference Schemes given in this specification is recommended, but not required.

Note, Example, and Dictionary Basis elements of the “glossary entry” for the concept in this specification are purely informative. All other elements are to be understood as giving the meaning and required characteristics of the concept. The glossary entry also specifies the representation of the concept that is used in this specification, while Clauses 13 and 15 specify the representation of the concept in exchange documents conforming to this specification.

NOTE: A concept is a meaning. Support for an SBVR concept is about using that meaning appropriately in the operation of the tool, and representing that meaning using the corresponding SBVR designator in SBVR exchange documents. The internal designations and other representations for the meaning, and the representation of that meaning in other exchange documents are not concerns of this specification.

2.2 Compliance Points

For conforming software, this specification defines four compliance points. A conforming software tool may conform to the compliance points as specified in 2.4 and 2.5. For every conforming software tool, a claim of conformance shall specify the compliance points to which conformance of the tool is claimed. The subclauses of this clause define the compliance points. Figure 2-1 shows the relationship of the compliance points in terms of the UML packages to which they correspond.

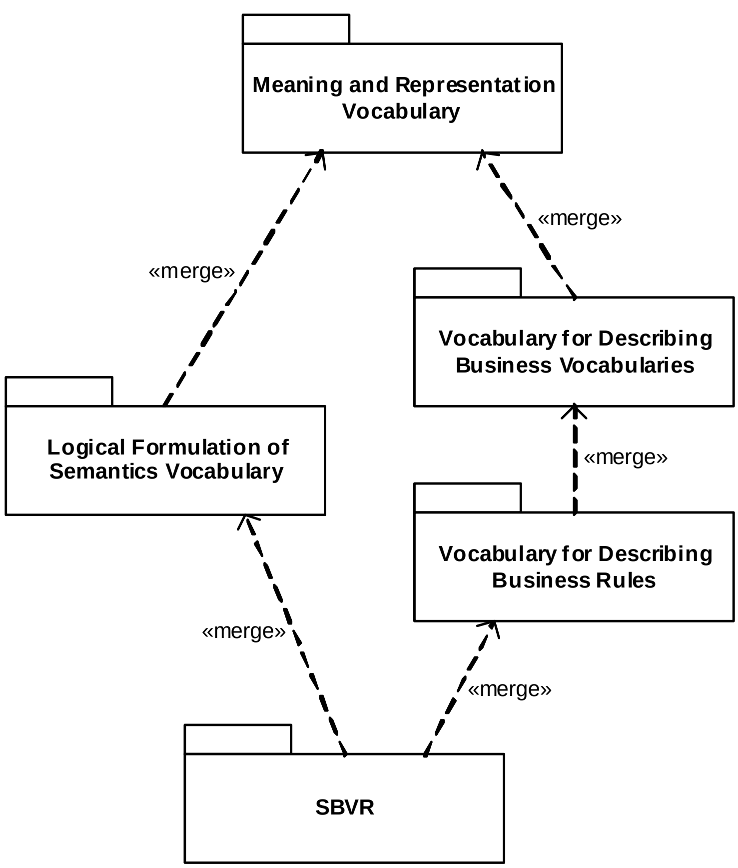


Figure 2.1

2.2.1 Meaning and Representation

A software tool that conforms to this compliance point shall support all of the concepts in the Meaning And Representation Vocabulary specified in Clause 8. This corresponds to support for UML Package “Meaning and Representation Vocabulary.”

2.2.2 Logical Formulation of Semantics

A software tool that conforms to this compliance point shall support (as defined in 2.1) all of the concepts in the Logical Formulation of Semantics Vocabulary specified in Clause 9. This corresponds to support for UML Package “Logical Formulation of Semantics Vocabulary.”

2.2.3 Business Vocabulary

A software tool that conforms to this compliance point shall support (as defined in 2.1) all of the concepts in the Business Vocabulary specified in Clause 11. This corresponds to support for UML Package “Vocabulary for Describing Business Vocabularies.”

2.2.4 Business Rules

A software tool that conforms to this compliance point shall support (as defined in 2.1) all of the concepts in the Business Rules Vocabulary specified in Clause 12 and all of the concepts in the Business Vocabulary specified in Clause 11. This corresponds to support for UML Package “Vocabulary for Describing Business Rules.”

Issue 9959 Add text

2.2.5 Restricted Higher Order Logic (Additional Conformance)

An SBVR exchange document that conforms to this compliance point shall satisfy the requirement stated in clause 10.3.1 and 10.3.2.

A software tool that conforms to this compliance point shall conform as an SBVR producer (see 2.4) and shall produce no exchange file that does not conform to this compliance point, as defined above.

2.2.6 First Order Logic (Additional Conformance)

An SBVR exchange document that conforms to this compliance point shall satisfy the requirement stated in clause 10.3.1 and 10.3.3.

A software tool that conforms to this compliance point shall conform as an SBVR producer (see 2.4) and shall produce no exchange file that does not conform to this compliance point, as defined above.

2.3 Conformance of an SBVR exchange document

An exchange document that conforms to this specification (an "SBVR exchange document") shall be an XML document that represents a 'fact model' as defined in subclause 8.5.

The fact model shall be based on the conceptual schema specified in subclause 13.5 - the "SBVR model of SBVR." The exchange document shall identify its document type as one of the XML Schemas specified in subclause 15.2, using the URI for that schema specified in 15.3.

NOTE: A business vocabulary or a business conceptual schema can be stated as a fact model that conforms to one of the conceptual schemas in Clause 15. The conformance of a fact model to a business conceptual schema so defined could be specified by the business that owns it, following the pattern of this specification. But this specification only defines conformance rules and Necessities for the concepts defined in the SBVR conceptual schema. Specifying the real requirements for conformance to a business-defined schema is beyond the scope of SBVR.

The body of facts represented in the fact model shall not contradict any Necessity in the SBVR conceptual schema. However, no concept is closed in the SBVR conceptual schema. A conforming fact model need not identify all things that necessarily exist, and a conforming fact model need not include a fact that expresses every necessary property of a thing that is referenced in the fact model. No Necessity should be interpreted as a requirement for inclusion of a fact in the fact model.

EXAMPLE

There is a rule that every statement expresses exactly one proposition. A fact model that includes that a given statement expresses two different propositions is not conformant. But a conforming document can include a statement without relating the statement to a proposition, even though the proposition necessarily exists.

NOTE: If a use of SBVR for exchange between tools requires that certain kinds of things or facts be fully represented in the exchange document, the SBVR conceptual schema can be extended for that purpose by adding the facts that particular concepts are closed or particular fact types are internally closed (see 8.5).

An exchange document that conforms to this specification may include representations of instances of any class (noun concept) or association (fact type) that is defined in Clauses 8, 9, 11, or 12.

NOTE: Not every conforming processor will support all of the concepts that can appear in a conforming SBVR document. Every conforming processor, however, is required to accept every conforming document. See 2.5.

For an XML exchange document that involves multiple namespaces, conformance to this specification is only defined for that part of the exchange document that uses the SBVR namespaces defined in this specification.

NOTE: The document type of a conforming XML exchange document need not be one of the XML schemas defined in Clause 15. For example, the document schema may include an SBVR schema as a subordinate namespace. Similarly, the SBVR schemas permit items like 'definitions' to have formal representations defined by other XML schemas.

2.4 Conformance of an SBVR Producer

A software tool that conforms as an SBVR producer shall produce exchange documents that conform to this specification as specified in 2.3.

An SBVR producer may be able to produce representations of instances of any concepts specified in Clauses 8, 9, 11, and 12. An SBVR producer is not required to be able to produce a representation of instances of any specific concept defined in this specification.

For a conforming SBVR producer, a claim of conformance shall identify the SBVR concepts for which it can produce representations of instances. It is recommended, but not required, that an SBVR producer be able to produce representations of instances of all of the concepts for one or more of the compliance points specified in 2.2.

NOTE: A conforming SBVR producer may be able to produce representations of instances of some but not all of the concepts defined for a compliance point. For such a software tool, support for the entire compliance point cannot be claimed, but its ability to produce representations of instances of the specific concepts it supports should be documented.

NOTE: As indicated in 2.3, an SBVR producer may produce instances of concepts not defined in SBVR as well. In such a case, the SBVR fact model would be only a part of the exchange document.

An SBVR producer shall support (as defined in 2.1) all of the SBVR concepts for which it is able to produce representations of instances.

An SBVR producer shall not convey in the exchange document the intent of an SBVR concept by using a representation that is not specified herein.

2.5 Conformance of an SBVR Processor

A software tool that conforms as an SBVR processor shall accept any exchange document that conforms to this specification as specified in 2.3. The interpretation it makes of any fact contained in the exchange document depends on whether the software tool supports the concepts associated with that fact (see below).

NOTE: Accepting a valid exchange document is distinguished from rejecting the document as not processable and using none of the information in it. A tool can accept a document and nonetheless discard much of the information in it. Accepting is also distinguished from supporting instances of concepts found in the exchange document, which refers to interpreting all facts about instances of the concept properly into the internal models and functions of the tool (See 2.1).

For an SBVR processor, the SBVR compliance points (see 2.2) to which it claims conformance shall be documented.

Every SBVR processor shall be able to accept representations of facts about instances of all SBVR concepts, whether they are associated with a compliance point for which conformance is claimed or not.

Every SBVR processor shall conform to the Meaning and Representation compliance point, as specified in 2.2.1. That is, it shall support (as defined in 2.1) instances of all concepts specified in the Meaning and Representation Vocabulary.

An SBVR processor for which conformance to any other compliance point specified in subclause 2.2 is claimed shall support instances of all concepts specified in the SBVR vocabulary associated with that compliance point.

NOTE: Depending on what the SBVR processor actually does with the SBVR fact model, there may be SBVR concepts for which there is no valid use in the function of the tool. For example, a tool that converts an SBVR fact model to some other modeling language or rules language may find that there are SBVR concepts that have no image in the target language. In such a case, the proper support for the SBVR concept may be to do nothing with it.

When an SBVR processor encounters a representation of an instance of a concept for which conformance is not claimed (including concepts that are not SBVR concepts), the processor may choose to do any of the following:

- ignore the instance;
- support the instance, and the SBVR concept it instantiates;
- interpret the instance via internal concepts that are not SBVR concepts per se.

An SBVR processor may, but need not, provide a warning when it encounters a representation of an instance it does not support.

~~Software that conforms to this specification will be able to import and export XMI documents that conform with the XMI rules applied to the normative metamodel contained in documents listed below under each conformance point. To be conformant, software must import and export.~~

~~There are five conformance points listed below. Software can be conformant with one or more conformance points and not with others, but conformance with any conformance point requires complete satisfaction of all of requirements of that conformance point. Any statement of conformance should specify the conformance points in which the implementation is conformant.~~

~~2.1 Logical Formulation of Semantics~~

~~A conformant software correctly consumes and produces XML documents that conform to the SBVR Logical Formulation of Semantics XML Schema. The conformant software also detects and reports when XML input violates necessary conditions stated by this specification.~~

~~2.2 Business Vocabulary~~

~~The software correctly consumes and produces XML documents conveying vocabulary information conforming to the SBVR Business Vocabulary XML Schema. The conformant software also detects and reports when XML input violates necessary conditions stated by this specification.~~

~~2.3 Business Vocabulary and Business Rules~~

~~The software correctly consumes and produces XML documents conveying information conforming to the SBVR Business Vocabulary and Business Rules XML Schema. The conformant software also detects and reports when XML input violates necessary conditions stated by this specification.~~

~~2.4 MOF 2 Generation from Vocabulary~~

~~The software correctly generates an XML document conforming to OMG's MOF 2 XML Schema from any XML document that conforms to the SBVR Logical Formulation of Semantics XML Schema. Production follows the rules of the Vocabulary-to-MOF/XMI Mapping Rule Set.~~

~~2.5 XMI Generation from Vocabulary~~

~~The software correctly generates an XML schema following OMG's XMI for MOF 2 Specification from any XML document that conforms to the SBVR Logical Formulation of Semantics XML Schema. Production follows the rules of the Vocabulary-to-MOF/XMI Mapping Rule Set.~~

3 Normative References

Issue 9753 Add text
Issue 10568 Add entry for ISO 6093
Issue 10569 Add MOF/ XMI text and remove bullet

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- Berners-Lee, T., R. Fielding, L. Masinter. IETF RFC 2396: *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998.
- International Organization for Standardization (ISO) : ISO 639-2. *Codes for the Representation of Names of Languages, Part 2*: Alpha-3 Code. Library of Congress, 2002.
- International Organization for Standardization (ISO) : 1087-1. *Terminology work — Vocabulary — Part 1: Theory and Application*
- *Meta Object Facility (MOF) Core Specification, v2.0* (http://www.omg.org/docs/formal/06-01-01.pdf).
- *MOF 2.0/XMI Mapping Specification, v2.1* (http://www.omg.org/docs/formal/05-09-01.pdf).
- International Organization for Standardization (ISO) : ISO 6093. *Information processing - Representation of numerical values in character strings for information interchange*. 1985.
- *OMG UML 2 Infrastructure, v2.1.1* (http://www.omg.org/docs/formal/07-02-04.pdf).
- *The Cambridge Dictionary of Philosophy*, 2nd ed. Cambridge University Press, 1999.
- *The New Oxford Dictionary of English*.
- *The Oxford Dictionary of English*.
- *Unicode 4.0.0 specification* : Glossary (http://www.unicode.org/versions/Unicode4.0.0/b1.pdf).
- ~~XMI 2.1 Tags~~

4 Terms and Definitions

Issue 9930: Remove text
Issue 9955 Add / replace text

For the purposes of this specification, the terms and definitions given in the normative reference and the following apply.

SBVR
shorthand for Semantics of Business Vocabulary and Business Rules.

SBVR Vocabularies
the vocabularies that make up SBVR itself, for talking about semantics, vocabulary, and rules.

Business Vocabulary
~~a structured set of terms and other symbols together with their meanings and relationships among them, for use by a business community.~~
A vocabulary that is under business jurisdiction.

Business Rule
a rule that is under business jurisdiction.

Business Vocabulary+Rules
a business vocabulary plus a set of business rules specified in terms of that business vocabulary.

SBVR Metamodel

the MOF model created from the combination of SBVR's Logical Formulation of Semantics Vocabulary, Vocabulary for Describing Business Vocabularies, and Vocabulary for Describing Business Rules. ~~guided by the Vocabulary-to-MOF/XMI Mapping Rule Set~~.

Terminological Dictionary
a collection of representations including at least one designation or definition of each of a set of concepts from one or more specific subject fields, together with other representations of facts related to those concepts.

Vocabulary
a set of designations (such as terms and names) and fact type forms primarily drawn from a single language to express concepts within a body of shared meanings. Note that this specification does not use the word "vocabulary" to refer to a dictionary or to any other sort of collection of terminological data.

5 Symbols

Issue 9930: Replace text

FL The indicated term is to be interpreted in formal logic. Terms without this symbol are not interpreted in formal logic.

~~The non-normative notation used in Part II and Annex E is specified in Annexes C, F, and H.~~

Figures in Clauses 8, 9, 11, and 12 depict the SBVR Metamodel using notational conventions described in Clause 13. For the purpose of visualizing vocabularies, Annex H describes a non-normative interpretation of those same figures and of figures in Annex E. Other non-normative notations used in Clauses 7 through 12 are explained in Annexes C and F.

6 Additional Information

6.1 Changes to Adopted OMG Specifications

This specification does not require or request any change to any other OMG specification.

Issue 9955: Replace text

6.2 How to Read this Specification

~~SBVR is a vocabulary, or actually a set of sub-vocabularies, each consisting of a set of terminological entries.~~

This specification describes a vocabulary, or actually a set of vocabularies, using terminological entries. Each entry includes a definition, along with other specifications such as notes and examples. Often, the entries include rules (necessities) about the particular item being defined.

The sequencing of the ~~sub-vocabularies~~ clauses in this specification reflects the inherent logical order of the subject matter itself. Later ~~sub-vocabularies~~ clauses build semantically on the earlier ones. The initial ~~sub-vocabularies~~ clauses are therefore rather 'deep' in terms of SBVR's grounding in formal logics and linguistics. Only after these ~~sub-vocabularies~~ clauses are presented do ~~sub-vocabularies~~ clauses more relevant to day-to-day business communication and business rules emerge.

This overall form of presentation, essential for a vocabulary standard, unfortunately means the material is rather difficult to approach. A ~~diagram~~ figure presented for each sub-vocabulary does help illustrate its structure; however, no continuous 'narrative' or explanation is appropriate.

6.2.1 About the Annexes

For that reason, the first-time general reader is urged to start with some of the non-normative Annexes, which do provide full explanation of the material, as well as context and purpose.

- Annex A, Overview of the Approach, is strongly recommended in that regard. It provides a general introduction to the fundamental concepts and approach of SBVR.
- Annex B, The Business Rules Approach, explains the core ideas and principles of business rules, which underpin SBVR's origin and focus. This short Annex is strongly recommended for readers who are unfamiliar with this area.

Good preparation for reading the specification is becoming familiar with the notation (non-normative) used to present the entries.

- Annex C, SBVR Structured English, provides comprehensive explanation in that regard.
- Annex D, SBVR Structured English Patterns, explains how to verbalize ~~vocabulary structure~~ terminological entries.

General practitioners will find the following sections of significant interest.

- Annex E, EU-Rent Example, provides a comprehensive case study, with a robust vocabulary and set of business rules fully worked through. Examples from EU-Rent are used widely in both the specification and Annexes to provide on-going commonality.
- Annex F, The RuleSpeak[R] Business Rule Notation, presents a widely-used, business-friendly syntax for expressing business rules.
- Annex G, Concept Diagram Graphic Notation, offers suggestions for how an SBVR vocabulary can be diagrammed.
- Annex H, Use of UML Notation in a Business Context to Represent SBVR-style Vocabularies, is of special interest to practitioners familiar with UML diagramming.

Object-Role Modeling (ORM)-related Annexes:

- Annex I, The ORM Notation for Verbalizing Facts and Business Rules, provides an introduction to the ORM approach. ORM contributes heavily to the theoretical underpinnings of SBVR, and represents some of the best practices in fact-based vocabulary and rule development.
- Annex J, ORM Discussion and Diagrams Related to the Logical Foundations for SBVR, provides supplemental ORM material further clarifying the normative material, Logical Foundations for SBVR.

Issue 9930: Remove text

~~For specialists in software engineering and tooling, especially regarding MOF and XMI, the following Annexes are of particular interest.~~

- ~~Annex K, Design Rational Details for the Use of MOF and XMI, explains the SBVR approach to implementation using MOF and XMI.~~
- ~~Annex L, Examples of SBVR's Use of MOF, illustrates the SBVR Approach described in Annex A.~~

For those specialists and researchers interested in standards and/or in the formal logics underpinning of SBVR, the following material is of special interest.

- Annex K, Mappings and Relationships to Other Initiatives, addresses where and how SBVR fits with other software and standards initiatives.

Issue 11263 Replace text

- ~~Annex L, Additional References, provides supplemental sources relevant to the formal logics underpinnings of SBVR.~~

For practitioners interested in a methodology supporting SBVR, used productively in industry for over 30 years, the fact-oriented approach NIAM2007 offers interesting advice.

- Annex L - a conceptual Overview of SBVR and the NIAM2007 Procedure to Specify a Conceptual Schema.
- Annex M, Additional References, provides supplemental sources relevant to the formal underpinnings of SBVR.

6.2.2 About the Normative Specification

Issue 9930: Remove text
Issue 9955 Replace text

The rest of this document contains the technical content of this specification. As background for this specification, readers are encouraged to first read:

Clauses 7-15 contain clauses for the SBVR vocabularies and rules that are the foundation for the SBVR Metamodel. ~~as well as a definition of the Essential SBVR Package, which is part of all MOF models created following the Vocabulary-to-MOF/XMI Rule Set.~~

Clauses 7-15 address different audiences. Four of the clauses are directly tied to conformance points, which are listed in Clause 2. Clause 7 gives names to the SBVR Vocabularies and to some other vocabularies and namespaces used by SBVR. Clause 8 provides the Meaning and Representation Vocabulary, which covers different kinds of meaning and representations. It is the foundation for the rest of the specification. Clause 9 provides the Logical Formulation of Semantics Vocabulary, which is the SBVR way to formulate semantics. It is not a vocabulary for business people but, rather, for detailed descriptions of the meanings of business words and statements. Clause 10 shows the formal logics and mathematical underpinnings of SBVR. Numerous concepts in clauses 8 and 9 are marked with the symbol ‘FL’ indicating that they are mapped to formal logics concepts in 10.

Clauses 11 and 12 provide (respectively) the Vocabulary for Describing Business Vocabularies and the Vocabulary for Describing Business Rules, which are for use in business to describe vocabularies and terminological dictionaries (11) and business rules (12).

Clause 13 specifies how SBVR uses MOF and XMI. Clause 14 is an index of vocabulary entries in Clauses 7-13. Clause 15 lists supporting documents, such as an XMI-based XML schema for the SBVR Metamodel.

Clauses 7-15 use SBVR Structured English to define the SBVR vocabularies and rules. Annex C describes how the Structured English is interpreted such that SBVR is specified in terms of itself.

Much of the material in Part II is illustrated by examples in the annexes, especially Annex E.

Although the clauses are organized in a logical manner and can be read sequentially, this is a reference specification and is intended to be read in a non-sequential manner. Consequently, extensive cross-references are provided to facilitate browsing and search.

6.3 Acknowledgements

Issue 11263: Add text

The following companies submitted and/or supported parts of this specification:

- Adaptive
- Automated Reasoning Corporation
- Business Rule Solutions, LLC
- Business Rules Group
- Business Semantics Ltd
- Fujitsu Ltd
- Hendryx & Associates
- Hewlett-Packard Company
- InConcept
- LibRT
- KnowGravity Inc
- MEGA
- Model Systems
- Neumont University
- Perpetual Data Systems
- PNA Group
- Sandia National Laboratories
- The Rule Markup Initiative
- Unisys Corporation
- X-Change Technologies Group

Part II - Business Vocabulary+Rules for Business Vocabulary+Rules

Issue 10442 Add text
Issue 9930 Remove text
Issue 9955 Rplace text
Issue 9950 Add text

This part contains sections for the SBVR vocabularies and rules that are the foundation for the SBVR Metamodel. ~~as well as a definition of the Essential SBVR Package, which is part of all MOF models created following the Vocabulary-to-MOF/XMI Rule Set.~~

The clauses of Part II address different audiences. Clause 7 gives names to the SBVR Vocabularies and to some other vocabularies and namespaces used by SBVR. Clause 8 provides the Meaning and Representation Vocabulary, which covers different kinds of meaning and representations. It is the foundation for the rest of the specification. Clause 9 provides the Logical Formulation of Semantics Vocabulary, which is the SBVR way to formulate semantics. It is not a vocabulary for business people, but rather, for detailed descriptions of the meanings of business words and statements. Clause 10 shows the formal logics and mathematical underpinnings of SBVR. Numerous concepts in clauses 8 and 9 are marked with the symbol 'FL' indicating that they are mapped to formal logics concepts in Clause 10.

Clauses 11 and 12 provide (respectively) the vocabulary for Describing Business Vocabularies and the Vocabulary for Describing Business Rules, which are for use in business to describe vocabularies and terminological dictionaries (11) and business rules (12).

Clause 13 specifies how SBVR uses MOF and XMI. Clause 14 is an index of vocabulary entries in Part II. Clause 15 lists supporting documents, such as an XMI-based XML schema for the SBVR Metamodel.

Part II uses SBVR Structured English to define the SBVR vocabularies and rules. Annex C describes how the Structured English is interpreted such that SBVR is specified in terms of itself. Although the Structured English is nonnormative, its use in Clauses 7 through 12 has a normative interpretation described in subclause 13.6. Examples are in natural language and use no particular notation except where noted.

Much of the material in Part II is illustrated by examples in the annexes, especially Annex E.

7 Vocabulary Registration Vocabulary

Issue 11283 Change URL throughout document

7.1 Vocabulary Registration Vocabulary

This subclause gives names of vocabularies, rule sets and namespaces. Each one is either provided by SBVR or is external to SBVR but formally referenced.

Vocabulary Registration Vocabulary
Language: English

7.1.1 Vocabularies Presented in this Document

Issue 9468 Add text
Issue 9930 Change text, remove text, add .xml to URLs

Vocabulary Registration Vocabulary
General Concept: vocabulary
Note: This clause.
Namespace URI: http://www.omg.org/spec/SBVR/1.0/VocabularyRegistration.xml

Meaning and Representation Vocabulary
General Concept: vocabulary
Note: See Clause 8 - Meaning and Representation Vocabulary.
Namespace URI: http://www.omg.org/spec/SBVR/1.0/MeaningAndRepresentation.xml

Logical Formulation of Semantics Vocabulary
General Concept: vocabulary
Note: See Clause 9 - Logical Formulation of Semantics Vocabulary.
Namespace URI: http://www.omg.org/spec/SBVR/1.0/LogicalFormulationOfSemantics.xml

Formal Logic and Mathematics Vocabulary
General Concept: vocabulary
Note: See Clause 10 - Providing Semantic and Logical Foundations for Business Vocabulary and Rules.
Namespace URI: http://www.omg.org/spec/SBVR/1.0/FormalLogicAndMathematics.xml

Vocabulary for Describing Business Vocabularies

General Concept: vocabulary

Note: See Clause 11 - Business Vocabulary.

Namespace URI: http://www.omg.org/spec/SBVR/1.0/DescribingBusinessVocabularies.xml

Vocabulary for Describing Business Rules

General Concept: vocabulary

Note: See Clause 12 - Business Rules.

Namespace URI: http://www.omg.org/spec/SBVR/1.0/DescribingBusinessRules.xml

SBVR Vocabulary

Definition: vocabulary that is a combination of the following: Meaning and Representation Vocabulary, Logical Formulation of Semantics Vocabulary, Vocabulary for Describing Business Vocabularies, and Vocabulary for Describing Business Rules

Namespace URI: http://www.omg.org/spec/SBVR/1.0/SBVR.xml

~~Vocabulary-to-MOF/XMI Vocabulary~~

~~Definition:~~ ~~the vocabulary that is used to describe transformation of any vocabulary defined in terms of SBVR into a MOF/XMI implementation that supports repository services and data interchange of facts in terms of atomic formulations using the SBVR vocabulary~~

~~Note:~~ ~~See Section 13.1 - Vocabulary-to-MOF/XMI Vocabulary.~~

~~Namespace URI:~~ ~~http://schema.omg.org/specs/SBVR/1.0/VocabularyToMOFXMI~~

~~Essential SBVR Vocabulary~~

~~General Concept:~~ ~~vocabulary~~

~~Note:~~ ~~See Section 13.2 - Essential SBVR.~~

~~Namespace URI:~~ ~~http://schema.omg.org/specs/SBVR/1.0/EssentialSBVRVocabulary~~

Issue 10568: Remove text from heading
Issue 9930 Remove entire 7.1.2 entry

~~7.1.2 Rule Sets Presented in this Document~~

~~Vocabulary-to-MOF/XMI Mapping Rule Set~~

~~General Concept:~~ ~~set~~

~~Note:~~ ~~See Section 13.3 - Vocabulary-to-MOF/XMI Mapping Rule Set.~~

Issue 10568: Remove text

~~Integer Namespace~~

~~Definition:~~ ~~the vocabulary namespace that has designations for all integers, each designation representing an individual concept of a particular integer using a sequence of one or more decimal numerals, optionally preceded by a minus sign ("-")~~

~~Note:~~ ~~The Integer Namespace includes designations using every possible sequence of decimal numerals, with and without a leading minus sign.~~

~~Namespace URI:~~ ~~http://schema.omg.org/specs/SBVR/1.0/Integers~~

7.1.3 External Vocabularies and Namespaces

Issue 10568 Add text
Issue 9930 Remove text

ISO 1087-1 (English)

Definition: the vocabulary for the English language specified in [ISO1087-1]

ISO 6093 Number Namespace

Definition: the namespace of designations for decimal numbers specified in [ISO6093]

Namespace URI: urn:iso:std:iso:6093:clause:8

ISO 639-2 (English)

Definition: the vocabulary of English language names of languages specified in [ISO639-2] available at http://www.loc.gov/standards/iso639-2/englangn.html

Namespace URI: http://www.loc.gov/standards/iso639-2/php/English_list.php

ISO 639-2 (Alpha-3 Code)

Definition: the vocabulary of 3-letter codes for languages specified in [ISO639-2] available at http://www.loc.gov/standards/iso639-2/langcodes.html

Namespace URI: http://www.loc.gov/standards/iso639-2/php/code_list.php

UML 2 Infrastructure

Definition: the namespace of designations for UML 2 Infrastructure concepts as defined by [UML2infr].

Unicode Glossary

Definition: the vocabulary presented in [Unicode4].

Uniform Resource Identifiers Vocabulary

Definition: the vocabulary presented in [IETF RFC 2396].

~~XMI 2.1 Tags~~

~~Definition:~~ ~~the vocabulary namespace of tagged values of [XMI2.1].~~

8 Meaning and Representation Vocabulary

The primary subjects of the Meaning and Representation Vocabulary fit between two other relevant subject areas described below.

1. **Expression** – things used to communicate (e.g., sounds, text, diagrams, gestures), but apart from their meaning — one expression can have many meanings.

2. **Representation** – the connection between expression and a meaning. Each representation ties one expression to one meaning.

3. **Meaning** – what is meant by a word (a concept) or by a statement (a proposition) – how we think about things.

4. **Extension** – the things to which meanings refer, which can be anything (even expressions, representations, and meanings when they are the subjects of our discourse).

Issue 9958 Change text

Following are examples of how some things, like "driver," cross through each subject area.

Extension	**Meaning**	**Representation**	**Expression**
The actual drivers of motor vehicles	Concept 'driver' — how we think of drivers, what characterizes them	Designation of the concept 'driver' by the signifier "driver"	The character sequence "driver"
		Definition of the concept 'driver' as "operator of a motor vehicle"	The character sequence "operator of a motor vehicle"
The actual City of Los Angeles, California – a real place	Individual concept 'Los Angeles' — how we think of that city, what distinguishes it from other places	'Los Angeles' as a designation for the individual concept of 'Los Angeles'	The character sequence "Los Angeles"
For each car that is out of service, its actually being out of service	Characteristic applicable to a car, what is meant by a car being out of service	Fact type form ~~form of expression~~ 'car is out of service' as a template for the characteristic with 'car' being a placeholder	The text "car is out of service"
The actual state of affairs of it being obligatory in the EU-Rent business that it not rent to a barred driver	Proposition — the meaning of the statement "EU-Rent must not rent to a barred driver"	The statement, "EU-Rent must not rent to a barred driver," having the proposition as its meaning	The character sequence "EU-Rent must not rent to a barred driver"

Another subject area of this vocabulary is reference schemes, which are ways people use information about something to identify it. For example, a city in the United States is identified by a name combined with the state it is in. The state is identified by its name or by a two-letter state code.

Representations provide a reference scheme for concepts and propositions because they are always tied to exactly one expression and to exactly one meaning. On the other hand, a single expression can have multiple meanings, a concept can have multiple expressions, a thing can be an instance of many concepts, and a proposition can be meant by many equivalent expressions.

A single representation can be tied to many speech acts, or to a single speech act, depending on how its expression is identified. For example, if the expression is a text or a sequence of words independent of any particular act of writing or speaking, the representation is independent in the same way. Conversely, if the expression is identified as belonging to a specific speech act, then the representation is tied to that speech act also.

Issue 9948 Add text

Note: in the glossary entries below, the words "Concept Type: role" indicate that an object type being defined is a role. Because it is an object type, it is necessarily a situational role and is not a fact type role.

The Meaning and Representation Vocabulary is not presented alphabetically. It is organized by subjects presented in the following order.

1. Meanings
   a. Concepts
   b. Propositions
   c. Questions
2. Expressions
3. Representations
4. Reference Schemes
5. Conceptual Schemas and Models
6. Extensions
7. Elementary Concepts

Meaning and Representation Vocabulary

Language: English

8.1 Meanings

Issue 9882 Replace figure (see also Issue 9451)
Issue 9960 Add “See Annex H” to all figures in this clause.
Issue 9721 Replace figure
Issue 9930 Replace figure caption to all figures in this Clause
Issue 9948 Replace figure

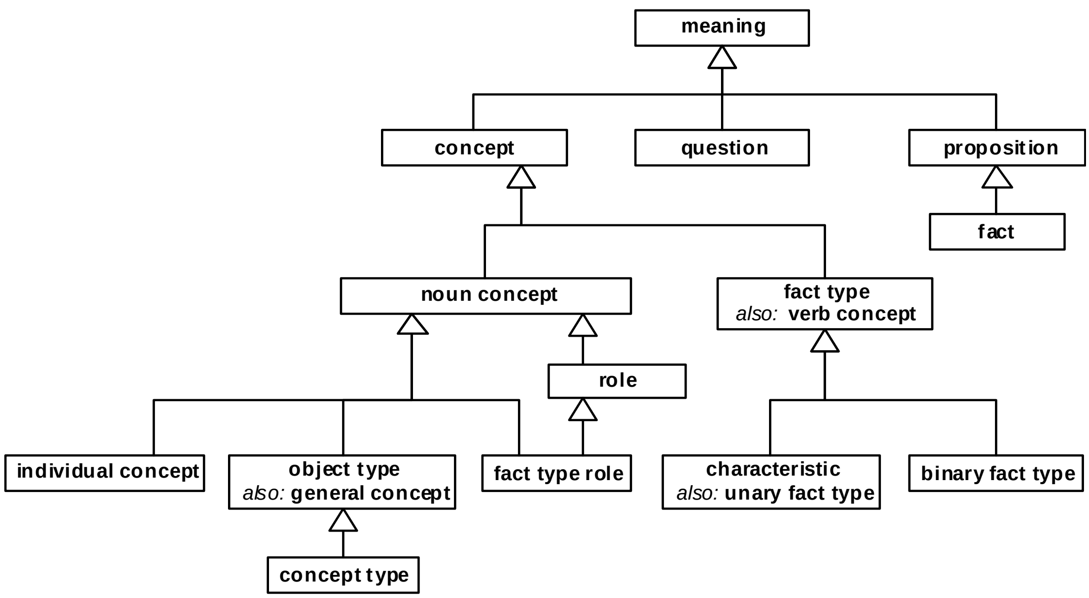


Figure 8.1

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

meaning

Definition: what is meant by a word, sign, statement, or description; what someone intends to express or what someone understands

8.1.1 Concepts

Issue 10573 Add text

concept FL

Source: ISO 1087-1 (English) (3.2.1) ['concept']

Definition: unit of knowledge created by a unique combination of characteristics

General Concept: meaning

Reference Scheme: a designation *of* the concept

Issue 10573 Revise text (Necessity)
Issue 9948 Remove text, replace text
Issue 10801 Replace definition

noun concept

FL

Definition: concept that is the meaning of a noun or noun phrase

~~Synonym:~~ ~~object type~~

Concept Type: concept type

Reference Scheme: a closed projection that *defines* the noun concept

~~Necessity:~~ ~~The set of characteristics that *are incorporated by* a noun concept *is* not the set of characteristics that *are incorporated by* another noun concept.~~

~~Note:~~ ~~A noun concept incorporates a set of characteristics which are a unique combination that distinguishes the concept from other noun concepts. See 'concept *incorporates* characteristic'. If a concept A and a concept B have the very same incorporated characteristics, they are the same concept. If they have the very same necessary characteristics, they are logically equivalent and they denote the same things in all possible worlds.~~

~~Example:~~ ~~the concept 'car', the concept 'number', the concept 'person'~~

object type

Definition: noun concept that classifies things on the basis of their common properties

Source: based on ISO 1087-1 (English) (3.2.3) ['general concept']

Concept Type: concept type

Synonym: general concept

Necessity: The set *of* characteristics that *are incorporated by* an object type *is* not the set *of* characteristics that *are incorporated by* another object type.

Note: An object type incorporates a set of characteristics which are a unique combination that distinguishes that object type from all other object types. See 'concept *incorporates* characteristic'. If an object type A and an object type B have the very same incorporated characteristics, they are the same concept. If they have the very same necessary characteristics, they are logically equivalent and they denote the same things in all possible worlds.

Example: the concept 'rental car' corresponding to cars that are rented

Example: the concept 'car', the concept 'number', the concept 'person'

concept type

FL

Definition: ~~noun concept~~ object type that *specializes* the concept 'concept'

Note: A concept is related to a concept type by being an instance of the concept type.

Example: fact type, role, concept type

role

FL

Definition: noun concept that corresponds to things based on their playing a part, assuming a function or being used in some situation

~~Necessity:~~ ~~Each role *is of* at most one fact type.~~

Concept Type:	concept type
~~Reference Scheme:~~	~~a placeholder that *represents* the role~~
~~Reference Scheme:~~	~~a variable that *maps to* the role~~
~~Reference Scheme:~~	~~a characteristic that *has* the role~~
Example:	the role 'drop-off location' of the fact type 'shipment has drop-off location'
Example:	the role 'shipment' of the fact type 'shipment has drop-off location', which should not be confused with the general concept 'shipment' (which generalizes the role)
Example:	the role 'sum' – a role of a number in relation to a set of numbers
~~Note:~~	~~If a role is defined generally, either with respect to a very general fact type or without defining a corresponding fact type, then that role can be specialized with respect to specific fact types as in the example below.~~
~~Example:~~	~~a role 'pick-up date' could be defined generally as a date for picking up something. The role 'pick-up date' of the fact type 'rental has pick-up date' is a separate role that specializes the generally defined one by being limited to pick-up dates of rentals and not other things. The role 'pick-up date' of the fact type 'shipment has pick-up date' is yet another separate role that specializes the generally defined one.~~
Note:	A role can be an object type or a fact type role. A role is always understood with respect to actualities of a particular fact type or to other particular situations.

fact type role

Definition:	role that specifically characterizes its instances by their involvement in an actuality that is an instance of a given fact type
Concept Type:	concept type
Reference Scheme:	a placeholder that *represents* the fact type role
Reference Scheme:	a variable that *maps to* the fact type role
Reference Scheme:	a characteristic that *has* the fact type role
Necessity:	Each fact type role *is in* exactly one fact type.
Necessity:	No fact type role *is* an object type.
Note:	A fact type role is fundamentally understood as a point of involvement in actualities that correspond to a fact type. Its incorporated characteristics come from the fact type - what the fact type requires of instances of the role. It is possible that two fact type roles incorporate the same characteristics, such as when a binary fact type means the same thing when roles are reversed, as in 'person is married to person'.

Issue 9958 Change form of expression to fact type form
Issue 10573 Add text
Issue 9948 Revise text
Issue 10801 Replace definition

---

fact type

FL

Definition:	concept that is the meaning of a verb phrase that involves one or more noun concepts and whose instances are all actualities
Synonym:	verb concept

Note: For each instance of a fact type, each role of the fact type is one point of involvement of something in that instance.

Note: Two fact type definitions define the same fact type if they reveal the same incorporated characteristics and the same fact type roles.

Concept Type: concept type

Necessity: Each fact type *has* at least one role.

Reference Scheme: a fact type form ~~form of expression~~ *of* the fact type

Reference Scheme: a closed projection that *defines* the fact type

characteristic

FL

Definition: fact type that *has* exactly one role

Source: ISO 1087-1 (English) (3.2.4) ['characteristic']

Definition: abstraction of a property of an object [thing] or of a set of objects

Synonym: unary fact type

~~Reference Scheme:~~ ~~a role *of* the fact type~~

Example: The fact type 'shipment is late' whose instances are actualities of shipments being late. There is one instance of the fact type for each shipment that is late.

Note: A characteristic always has exactly one role, but it can be defined using fact types having multiple roles.

Example: The characteristic 'driver is of age' with this definition: "the age of the driver is at least the EU-Rent Minimum Driving Age." The semantic formulation of this definition appears in the introduction to Clause 9 - Logical Formulation of Semantics Vocabulary.

binary fact type

FL

Definition: fact type that *has* exactly 2 roles

Example: The fact type 'shipment has drop-off location' whose instances are actualities of shipments having drop-off locations.

Example: The fact type 'number is greater than number' whose instances are actualities of numbers being greater than other numbers, there being one instance for every pair of numbers where one is greater than the other.

Note: A fact type can have two roles that seem to be identical (e.g., 'person is married to person' where each role can be called 'spouse'). ~~But the two roles are distinct within the fact type, each one specializing a more general role (e.g., the role 'spouse' defined as a participant in a marriage)~~. Even though they incorporate the same characterstics, they are distinct in that they indicate two distinct points of involvement in each actuality the fact type corresponds to.

Issue 9451 Remove text
Issue 9948 Restore text, add text
Issue 10790 Replace text

---

individual concept

FL

Source: ISO 1087-1 (English) (3.2.2) ['individual concept']

Definition: concept that corresponds to only one object [thing]

General Concept: noun concept

Concept Type: concept type

Necessity: No individual concept *is* an object type.

Necessity: No individual concept *is* a fact type role.

~~Note:~~ ~~An individual concept always has one instance, but not necessarily the same instance in all possible worlds.~~

Note: While each referring individual concept has exactly one and the same instance in all possible worlds, there can be multiple individual concepts that correspond to the same thing. Different definite descriptions of the same individual thing can represent different individual concepts that correspond to that thing.

Example: The individual concept 'California' whose one instance is an individual state in the United States of America

8.1.1.1 About Concepts

Issue 9948 Replace figure

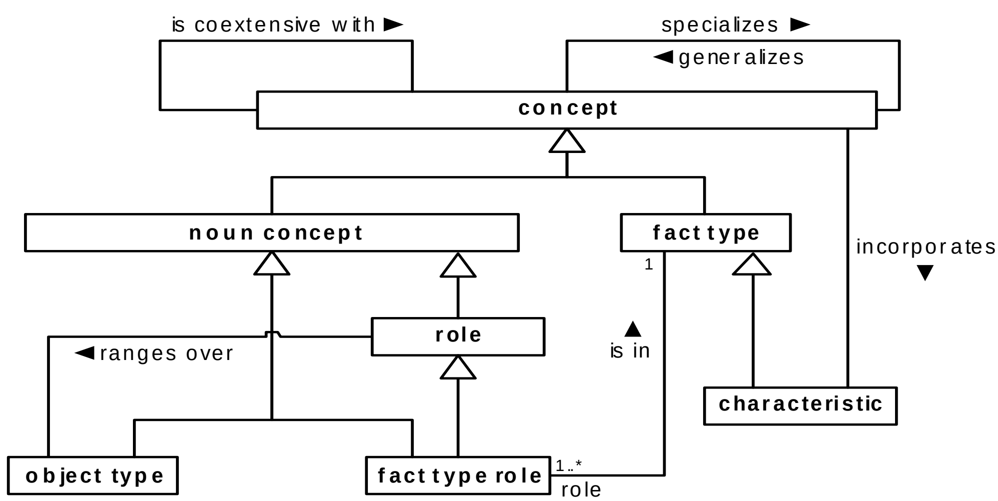


Figure 8.2

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 10571 Revise text
Issue 9948 Remove text

concept$_1$ *specializes* concept$_2$ FL

Definition: the concept$_1$ incorporates each characteristic that is incorporated by the concept$_2$ plus at least one differentiator

Synonymous Form: concept$_2$ *generalizes* concept$_1$

Note: The extension of a concept that specializes another is always a subset of the extension of the other, but not necessarily a proper subset. The differentiator that makes one concept more specific than the other is conceptual and does not necessarily restrict the extension of the concept.

Example: The noun concept 'whole number' specializes the noun concept 'integer', the differentiator being that whole numbers are nonnegative.

~~Example: The role 'sum' specializes the noun concept 'number', the differentiator being that each sum is the result of adding up a set of numbers. It turns out that every number is a sum of that number added to zero, so the extensions of the concepts 'sum' and 'number' are always the same. Nevertheless, the role 'sum' incorporates the differentiating characteristic of being the result of addition, so it specializes the noun concept 'number'.~~

Example: The individual concept 'Los Angeles' specializes the concept 'city', the differentiator being that Los Angeles is one particular city in California.

Issue 10779 Replace text
Issue 9948 Remove text, replace text

concept$_1$ *is coextensive with* concept$_2$

FL

Definition: the extension *of* the concept$_1$ ~~always *equals*~~ *is* always the extension *of* the concept$_2$

Note: Semantic integrations between communities often involve recognizing where different concepts (having different intensions) have the same extensions in all possible worlds. Also, it is possible that concepts employing different methods of conceptualization have the same extension in all cases. For example, a noun concept that specializes the concept 'actuality' can be coextensive with a fact type.

~~Example: The role 'sum' is coextensive with the noun concept 'number'.~~

Example: The individual concept defined as "the thirtieth president of the United States" is coextensive with an object type defined as "president of the United States in 1925." The two concepts have the same extension (which includes only Calvin Coolidge) but they are different concepts.

Issue 10571 Add text, remove text, add note and example
Issue 9948 Revise text

concept *incorporates* characteristic

FL

Definition: the characteristic is an abstraction of a property of each instance of the concept and is one of the characteristics that makes up the concept

~~Synonymous Form: characteristic *is incorporated into* concept~~

Note: Every characteristic incorporated by a concept is a necessary characteristic of the concept, but not every necessary characteristic of the concept is incorporated by the concept. Only those that are part of what makes up the concept are considered to be incorporated. Given an intensional definition of a concept, incorporated characteristics include all of these:

1. characteristics incorporated by the definition's more general concept (recursively)
2. the definition's delimiting characteristics

3. characteristics intrinsic to the delimiting characteristics (see example below)

4. any conjunctive combination of any of the characteristics above

Given an extensional definition, one that uses disjunction, characteristics that are found on each side of the disjunction are incorporated characteristics. Two definitions can define the same ~~noun concept~~ object type by producing the same set of incorporated characteristics. The two definitions can directly identify different sets of incorporated characteristics (1 and 2 above) that are sufficient to determine the others (3 and 4 above). The way incorporated characteristics fall into 1 through 4 above can differ from one definition to another while producing the same overall set.

Example: The concept "wrecked rental car," defined as "rental car that is nonoperational due to being in an accident," incorporates the following characteristics:

1. characteristics incorporated by the more general concept 'rental car' - e.g., being a car, being a vehicle, being rentable, and (combining them all) being a rental car

2. the delimiting characteristic: being nonoperational due to being in an accident

3. characteristics intrinsic to the delimiting characteristics - e.g., being nonoperational and having been in an accident

4. all conjunctive combinations of the characteristics given above - e.g., being a nonoperational vehicle, being a wrecked car

Example: The concept 'qualified driver' incorporates the characteristic 'driver is licensed' because it is necessary (by the definition of 'qualified driver') that each qualified driver is licensed.

Issue 9948 Add entry, replace text

---

role *ranges over* object type

Definition: each characteristic that *is incorporated by* the object type *is incorporated by* the role

Note: Saying that a role ranges over an object type is similar to saying the role specializes the object type in that the role incorporates every characteristic incorporated by the object type, and therefore, each instance of the role is necessarily an instance of the object type. But "ranges over" is different in that it allows that both the role and the object type incorporate the same characteristics - the object type can incorporate a characteristic that its instances fill that role.

Note: An object type ranged over by a role can be a situational role.

Example: The role 'company' of the fact type 'company employs person' ranges over the object type 'company'.

fact type *has* role FL

Definition: the role is an abstraction of a thing playing a part in an instance of the fact type

Synonymous Form: fact type role *is in* fact type

8.1.2 Propositions

Issue 9721 Add figure

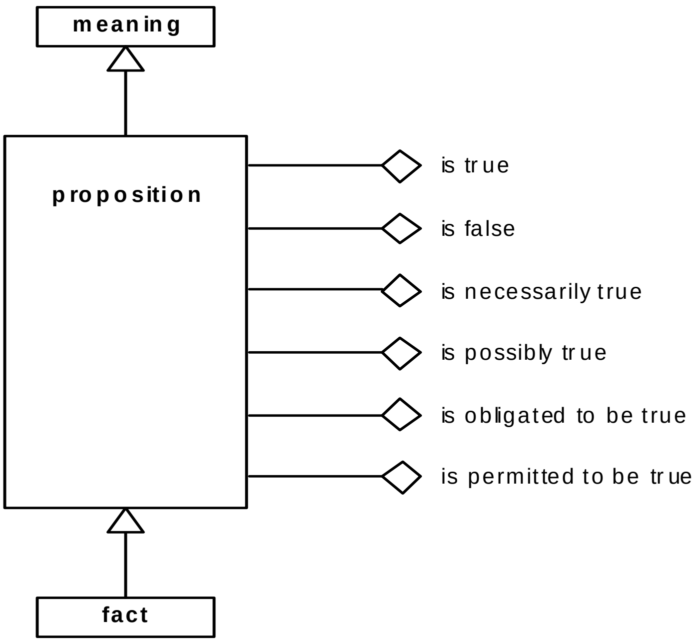


Figure 8.3

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9715 Revise text

proposition

FL

Definition: meaning that is true or false

Note: Every meaning that *is true* or *is false* is a proposition. That is, a proposition is a meaning that has a truth value.

Note: A proposition corresponds to a state of affairs in a possible world defined by a collection of things of interest and possibly a time frame. The same proposition can be true in one possible world and false in another.

Note: The word "proposition" has two common meanings: first, a statement that affirms or denies something, and second, the meaning of such a statement. The concept 'proposition' is here defined in the second sense and should not be confused with the statement of a proposition.

Reference Scheme: a closed logical formulation that *means* the proposition

Issue 9882 Add text
Issue 10569 Add FL to entries

proposition *is true*

FL

Definition: the proposition *corresponds to* an actuality

proposition *is false* FL

Definition: the proposition does not *correspond to* an actuality

fact FL

Definition: proposition that is taken as true

Note: How one ascertains what is true, whether by assertion, observation, or other means, is outside the scope of this specification. However, taking a proposition as true must be consistent with epistemic commitment. The concept 'fact' is here defined to be consistent with the operations of truth-functional logic, which produce results based on true and false.

Issue 9721 Remove text / Replace text

~~**modality**~~ ~~FL~~

~~Definition:~~ ~~noun concept that classifies a proposition based on any of a number of operations of modal logics such as deontic and alethic logics~~

~~Concept Type:~~ ~~concept type~~

~~Necessity:~~ ~~Each modality *specializes* the concept 'proposition'.~~

~~**necessity**~~ ~~FL~~

~~Definition:~~ ~~fact that is necessarily true, always true~~

~~Concept Type:~~ ~~modality~~

~~Synonym:~~ ~~logical necessity~~

~~**possibility**~~ ~~FL~~

~~Definition:~~ ~~proposition that is possibly true, there being no necessity that it be false~~

~~Concept Type:~~ ~~modality~~

~~Synonym:~~ ~~logical possibility~~

~~**obligation**~~ ~~FL~~

~~Definition:~~ ~~proposition that is required to be true, that is obligatory, that is not permitted to be false~~

~~Concept Type:~~ ~~modality~~

~~**permissibility**~~ ~~FL~~

~~Definition:~~ ~~proposition that is permitted to be true, there being no obligation that it be false~~

~~Concept Type:~~ ~~modality~~

proposition *is necessarily true* FL

Definition: the proposition always *corresponds to* an actuality

Note: A proposition is considered to be necessarily true if it is true by definition - the definitions of relevant concepts make it logically impossible for the proposition to be false.

proposition *is possibly true* FL

Definition: it is possible that the proposition *corresponds to* an actuality

Issue 10798 Replace text

proposition *is obligated to be true* FL

~~Definition:~~ ~~the proposition must *correspond to* an actuality~~

Definition: the proposition *corresponds to* an actuality in all acceptable worlds.

Note: The concept 'acceptable world' is described in Clause 10.

proposition *is permitted to be true* FL

~~Definition:~~ ~~the proposition may *correspond to* an actuality~~

Definition: the proposition *corresponds to* an actuality in at least one acceptable world.

Note: The concept 'acceptable world' is described in Clause 10.

8.1.3 Questions

question

Definition: meaning of an interrogatory

Note: The word "question" has two common meanings: first, a written or spoken expression of inquiry, and second, the meaning of such an inquiry. By the second definition, a single question could be asked in two languages. But by the first definition, using two language results in two expressions, and therefore, two questions. The concept 'question' is here defined in the second sense (meaning) and should not be confused with the expression or representation of a question.

Reference Scheme: a closed projection that *means* the question

8.2 Expressions

expression

Definition: something that expresses or communicates, but independent of its interpretation

Example: the sequence of characters "car"

Example: the sequence of speech sounds (t), (r), and ($\bar{e}$)

Example: a smile

Example: a diagram

Example: The entire text of a book

Issue 9952 Replace text

signifier

Definition: expression that is a linguistic unit or pattern, such as a succession of speech sounds, written symbols or gestures, used in a designation of a concept

Concept Type: role

Example: the sequence of characters "car" used in a designation of the concept 'automobile' or used in a designation of the concept 'railroad car'

Example: the sequence of speech sounds (t), (r), and (ē) used in a designation of the concept ‘tree’

Example: The ~~character~~ graphic “€” used in a designation of the concept ‘Euro’

text

Source: Unicode 4.0.0 Glossary ['Character Sequence']

General Concept: expression

Note: The concept ‘text’ has no explicit reference scheme, but rather, is used as a target for reference schemes.

Note: A detailed vocabulary concerning text is provided by the Unicode specification. Taking the concept ‘text’ from the Unicode specification does not mean that a text is a Unicode encoding, but rather, it implies that a text can be represented by a Unicode encoding in electronic communications. Unicode encodings provide the common means of text representation in word processors, mail systems, the Internet, and so on. The encodings tend to be invisible to people writing and reading the text.

starting character position

Definition: positive integer that is an ordinal position where a text starts within an encompassing text

Concept Type: role

Issue 10629 Remove text / add Definition

URI

Source: Uniform Resource Identifiers Vocabulary ['URI']

~~General Concept:~~ ~~text~~

~~Concept Type:~~ ~~role~~

Definition: text that identifies a resource as specified by [IETF RFC 2396]

Synonym: uniform resource identifier

Note: The concept ‘URI’ is introduced into this specification in order to provide a universal context for reference schemes.

8.3 Representations

Issue 9931 Replace figure
Issue 9932 Replace figure, add text (includes changes from Issue 9958)
Issue 9257 Replace figure
Issue 9930 Replace figure, revise figure caption
Issue 9952 Replace figure

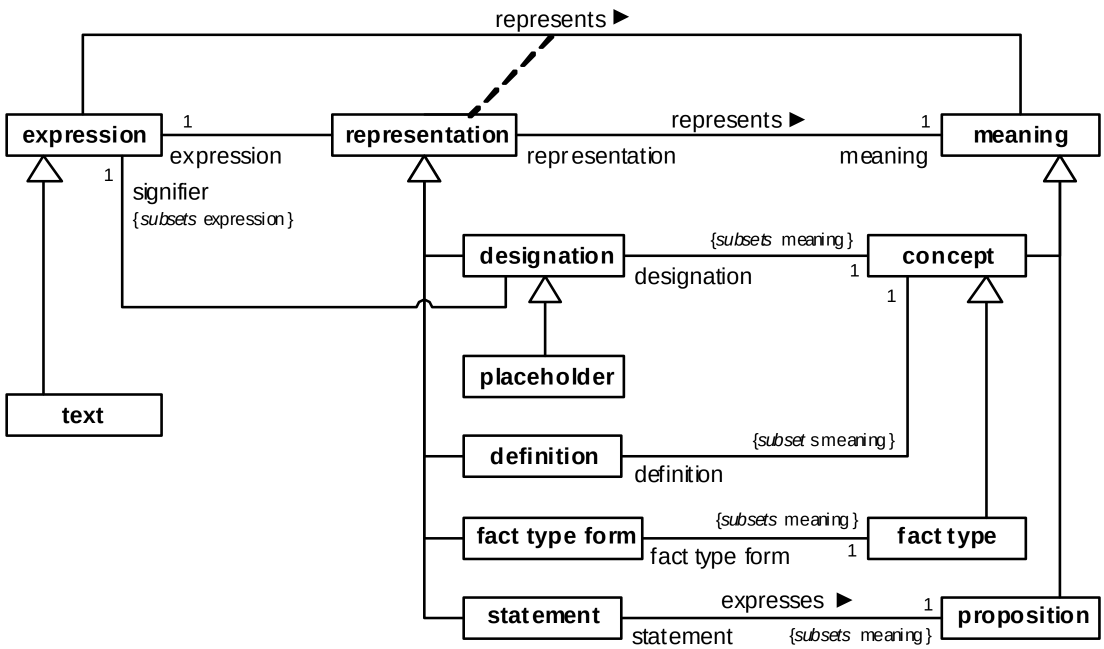


Figure 8.4

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

expression *represents* meaning

Definition: the expression portrays or signifies the meaning

representation

Definition: ~~portrayal of a meaning by an expression~~

Definition: actuality that a given expression *represents* a given meaning

Necessity: Each representation *has* exactly one expression.

Necessity: Each representation *represents* exactly one meaning.

representation *has* expression

Issue 9467 Remove text
Issue 9931 Add text

representation *represents* meaning

~~Synonymous Form:~~ ~~meaning *is represented by* representation~~

Synonymous Form: meaning *has* representation

Synonymous Form: representation *has* meaning

8.3.1 Designations

Issue 9931 Change styling "of a"
Issue 9952 Replace definition, add note

designation

Source: ISO 1087-1 (English) (3.4.1) ['designation']

~~Definition:~~ ~~representation of a concept by a sign which [means the concept and] denotes [the extension of] it~~

Definition: representation *of* a concept by a sign which denotes it

Note: In common usage, the signifier of a designation is used to refer to the instances of the designated concept. The designation, as defined here and in ISO 1087-1, does not refer to those instances directly, but relates the signifier to the concept. See 'concept has instance' in 8.6.1.

Necessity: Each designation *represents* a concept.

Reference Scheme: the signifier *of* the designation and a namespace that *includes* the designation

Reference Scheme: the signifier *of* the designation and the concept that *is represented by* the designation

designation *has* signifier

Definition: the signifier *is* the expression *of* the designation

concept *has* designation

Definition: the designation *represents* the concept

8.3.2 Definitions

Issue 9931 Change styling "of a"
Issue 10569 Move Necessity to 9.3
Issue 10571 Add Definition
Issue 10790 Add text / replace text

definition

Source: ISO 1087-1 (English) (3.3.1) ['definition']

Definition: representation *of* a concept by a descriptive statement [expression] which serves to differentiate it from related concepts

Definition: representation (as through a word or phrase) expressing the essential nature of a person or thing or class of persons or of things : an answer to the question "what is x?" or "what is an x?"

Necessity: Each definition *represents* a concept.

~~Necessity:~~ ~~Each closed projection that *formalizes* a definition *of* a concept *defines* the concept.~~

Reference Scheme: the expression *of* the definition and ~~the concept that *is represented by* the definition~~ a closed projection that *formalizes* the definition

Note: 'definition' is used in SBVR in the sense of the formal term "definiens."

concept *has* definition

Definition: the definition *represents* the concept

8.3.3 Statements

Issue 9931 Change styling "of a"
Issue 10569 Move Necessity to 9.2

statement

Definition: representation *of* a proposition by an expression of the proposition

Necessity: Each statement *expresses* exactly one proposition.

~~Necessity:~~ ~~Each closed logical formulation that *formalizes* a statement *means* the proposition that *is expressed by* the statement.~~

Reference Scheme: the expression *of* the statement and a closed logical formulation that *formalizes* the statement

Issue 9467 Remove text

statement *expresses* proposition

Definition: the statement *represents* the proposition

Synonymous Form: proposition *has* statement

~~Synonymous Form:~~ ~~proposition *is expressed by* statement~~

8.3.4 Fact Type Forms

8.3.4 ~~Forms of Expression~~

Issue 9453 Revise figure
Issue 9958 Replace figure
Issue 9958 Replace Forms of Expression with Fact type Forms (see above)

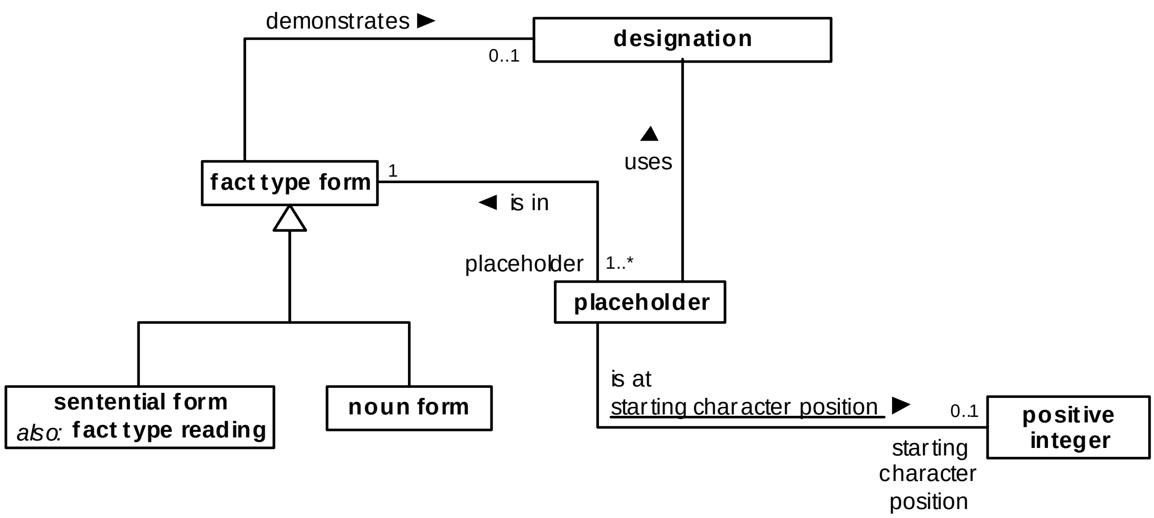


Figure 8.5

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9931 Change styling "of a"
Issue 9462 Remove text/Revise text (Reference Scheme)
Issue 9958 Change form of expression to fact type form

fact type form
~~form of expression~~

Definition: representation *of* a fact type by a pattern or template of expressions based on the fact type

Necessity: Each fact type form ~~form of expression~~ *represents* exactly one fact type.

Necessity: Each fact type form ~~form of expression~~ *has* at least one placeholder.

Necessity: At most one role *of* a fact type that *has* a fact type form ~~form of expression~~ *is* not *represented by* a placeholder *of* the fact type form ~~form of expression~~.

Necessity: No fact type form ~~form of expression~~ *is* a designation.

Necessity: Each fact type form ~~form of expression~~ *demonstrates* at most one designation.

Necessity: If a designation *is demonstrated by* a fact type form ~~form of expression~~ *of* a fact type then the fact type *has* the designation.

Example: The fact type form ~~form of expression~~ 'customer rents car' demonstrates the designation 'rents' and has two placeholders. One placeholder uses the designation 'customer' and is at the starting character position 1. The other placeholder uses the designation 'car' and is at the starting character position 16.

Example:	The fact type form ~~form of expression~~ 'driver of car' demonstrates a designation 'of' and has two placeholders, one using the designation 'driver' at the starting character position 1, and the other using the designation 'car' at the starting character position 11.
Example:	The fact type form ~~form of expression~~ 'country charges tax rate on date' demonstrates the designation 'charges on' that represents the same fact type as the fact type form ~~form of expression~~.
Note:	In some languages, fact type forms ~~forms of expression~~ occur that involve only a positioning of placeholders with no other designation — no verb or preposition.
Reference Scheme:	the expression *of* the fact type form ~~form of expression~~ and the set of ~~each~~ placeholders *of* the fact type form ~~form of expression~~ and a namespace that *includes* the fact type form ~~form of expression~~

Issue 9958 Change form of expression to fact type form
Issue 9930 Add Definition

fact type *has* fact type form

~~fact type *has* form of expression~~

Definition:	the fact type form ~~form of expression~~ provides a pattern or template for expressions denoting the fact type
Definition:	the fact type form *represents* the fact type

Issue 9467 Remove text
Issue 9958 Change form of expression to fact type form

fact type form *demonstrates* designation

~~form of expression *demonstrates* designation~~

Definition:	the fact type form ~~form of expression~~ shows a pattern of using the designation in an expression
~~Synonymous Form:~~	~~designation *is demonstrated by* form of expression~~

Issue 9958 Change form of expression to fact type form

fact type form *has* placeholder

~~form of expression *has* placeholder~~

Definition:	the placeholder indicates a place for expression of what fills a role in the fact type form ~~form of expression~~
Synonymous Form:	placeholder *is in* fact type form ~~form of expression~~

Issue 9958 Change form of expression to fact type form
Issue 9948 Replace text

sentential form

Definition: fact type form ~~form of expression~~ that is a pattern or template that can be used for stating a proposition based on a fact type

Synonym: fact type reading

Example: 'car is used in rental agreement' is a sentential form of a binary fact type.

Example: 'car is unavailable' is a sentential form of a unary fact type.

Example: Assuming there is a role 'renter' ~~specializing~~ ranging over the concept 'customer', the following can all be alternative sentential forms of the same fact type:

car has renter
customer rents car
car is rented by customer
renter rents car

Necessity: Each role *of* the fact type that *has* a sentential form *is represented by* a placeholder *of* the sentential form.

Issue 9453 Add Note, Replace Examples, Remove text
Issue 9958 Change form of expression to fact type form

---

noun form

Definition: fact type form ~~form of expression~~ that acts as a noun rather than forming a proposition

Note: A noun form can have a placeholder for each role of a fact type, in which case the noun form result comes from the of role the first placeholder is for. A noun form can also have one less placeholder than there are roles, in which case the noun form result comes from the role that no placeholder is for.

Example: 'transferred car of car transfer' for the fact type 'car transfer has transferred car'. This form yields a transferred car.

Example: '| number |'for the fact type 'number has absolute value'. The form yields the absolute value of the number.

Example: '$number_1$ + $number_2$' for the fact type '$number_1$ + $number_2$ = $number_3$'. This form yields the third number (the sum of adding the first two numbers).

Example: 'transferring rental car' for the fact type 'car transfer has transferred car'. This form yields the car transfer, which is an action. Gerunds are used in noun forms like this for actions, events, and states. They are used in sentences like this: "A rental car must be cleaned before transferring the rental car."

~~mathematical form~~

~~Definition:~~ ~~noun form that is a form for a mathematical expression whose result corresponds to a role not represented by a placeholder, but rather, by the whole form~~

~~Necessity:~~ ~~if a fact type *has* a mathematical form then exactly one role *of* the fact type *is represented by* no placeholder *of* the mathematical form~~

~~Note:~~ ~~In a mathematical form, there is no placeholder representing the result of an expression based on that form. This is a form common in mathematical expression.~~

~~Example:~~ ~~'number + number' is a mathematical form of a fact type having three roles. That same fact type could have sentential forms such as 'number + number = number' and nominal restrictive forms such as 'sum of number and number'.~~

~~Example: '|number|' is a mathematical form of a binary fact type. That same fact type could have a sentential form: 'number has absolute value' and a nominal restrictive form: 'absolute value of number'.~~

~~nominal restrictive form~~

~~Definition: noun form whose result corresponds to the role represented by its first placeholder~~

~~Necessity: Each role of the fact type that has a nominal restrictive form is represented by a placeholder of the nominal restrictive form.~~

~~Example: 'car used in rental agreement' is a nominal restrictive form that is a pattern of expression to refer to cars based on their being used in rental agreements. It is of the same fact type that could have a sentential form, 'car is used in rental agreement'.~~

~~Example: 'of' is used very commonly in nominal restrictive forms in English for binary fact types having a sentential form using the word 'has'. For example, the sentential form 'customer has name' implies a nominal restrictive form, 'name of customer'.~~

~~gerund form~~

~~Definition: noun form of a fact type used to denote an actuality that is an instance of the fact type~~

~~Necessity: Each role of the fact type that has a gerund form is represented by a placeholder of the gerund form.~~

~~Example: 'car being used in rental agreement' is a gerund form that is a pattern of expression to refer to actualities of cars being used in rental agreements. It is of the same fact type that could have a sentential form, 'car is used in rental agreement'.~~

~~Example: 'car being rented' is a gerund form of the characteristic 'car is rented'. It is used to denote actualities of cars being rented.~~

Issue 9931 Revise text
Issue 9958 Change form of expression to fact type form
Issue 9257 Delete text / revise text / add text
Issue 9948 Replace role with fact type role
Issue 9952 Replace text

---

placeholder

Definition: ~~representation~~ designation *of* a fact type role within a fact type form ~~form of expression~~ marking a place where, in uses of the fact type form ~~form of expression~~, an expression denotes what fills the fact type role

Necessity: Each placeholder *is in* exactly one fact type form ~~form of expression~~.

Necessity: Each placeholder *represents* exactly one fact type role.

Necessity: Each placeholder *of* each fact type form ~~form of expression~~ *of* a fact type *represents* a fact type role *of* the fact type.

Necessity: Each placeholder *has* at most one starting character position.

Necessity: Each placeholder *of* a fact type form ~~form of expression~~ that *has* a text *has* a starting character position.

~~Necessity: Each placeholder *uses* exactly one designation.~~

Reference Scheme: the fact type form ~~form of expression~~ that *has* the placeholder and the ~~designation that *is used for*~~ expression *of* the placeholder and the starting character position *of* the placeholder

Note: The expression of a placeholder often consists of the signifier of a designation used by the placeholder, but it can include other things such as delimiting characters (as in '[proposition] is true') or a subscript (as in 'proposition1 is true') by which the placeholder can be distinguished within the fact type form that has it. A placeholder need not use a designation (as in '… is true').

placeholder *is at* starting character position

Definition: the expression of the placeholder is textual and occurs within a textual expression of a fact type form starting at the starting character position

Synonymous Form: placeholder *has* starting character position

---

Issue 9958 Change form of expression to fact type form
Issue 9257 Delete Definition and Synonymous Form / add text

---

placeholder *uses* designation

~~Definition:~~ ~~the placeholder uses the designation of a concept for which use of the fact type form form of expression is understood~~

~~Synonymous Form:~~ ~~designation *is used for* placeholder~~

Definition: the expression of the placeholder incorporates the signifier of the designation thereby indicating that any use of the fact type form having the placeholder substitutes for the placeholder an expression understood to denote instances of the concept represented by the designation

Note: The means by which a placeholder incorporates a designation depends on convention. SBVR does not require a particular convention, but it uses one described in Annex C, SBVR Structured English.

Example: The 'proposition' placeholder in the fact type form 'proposition is true' uses the designation 'proposition'. The statement, "A fact is true," is understood to use that fact type form because a fact is a proposition, but "A line is true" is not recognized as using that fact type form because a line is not a proposition.

Example: Consider two fact type forms for the same fact type: 'rental is returned on date' and 'rental has return date'. The second placeholders of the two forms represent the same role, but they use different designations ('date' and 'return date'). If "Rental 876" denotes a rental, then the statement, "Rental 876 is returned on 30 June 2006," is understood to use the first fact type form because "30 June 2006" is understood to denote a date, but the statement, "Rental 879 has 30 June 2006," is not understood to use the second fact type form because "30 June 2006" is not understood to denote a return date (only a date). "Rental 879 has the return date 30 June 2006" uses the second fact type form.

Example: In the fact type form 'rental car_1 replaces rental car_2', both placeholders ('rental car_1' and 'rental car_2') use the same designation, 'rental car'.

8.3.5 Namespaces

Issue 9935, 9958, 10629, 9930, 9955Replace figure

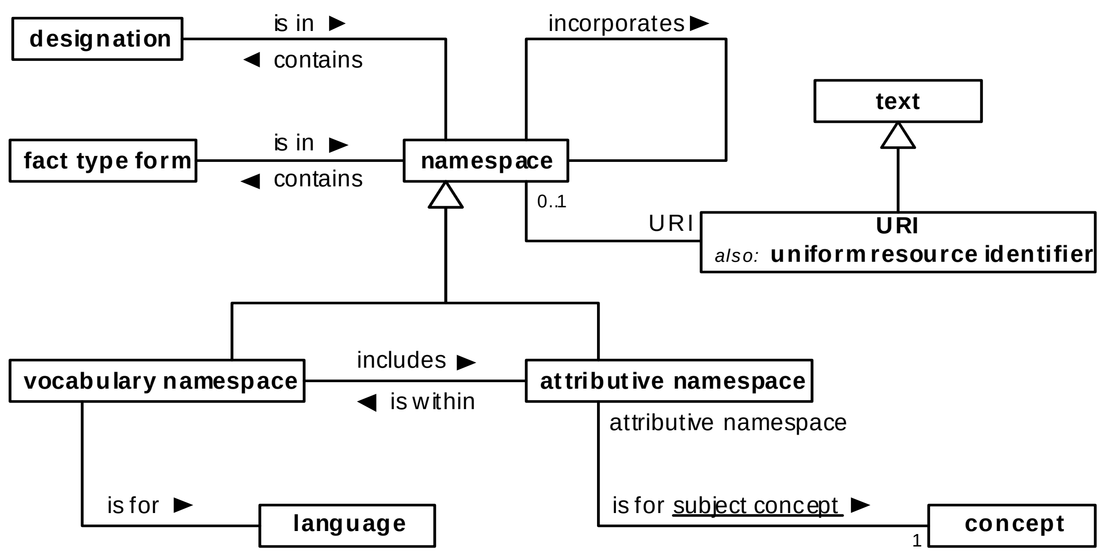


Figure 8.6

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9955 Replace text, add text

namespace

Definition: collection of desginations and/or fact type forms that are distinguishable from each other by uniqueness of designator or form

Reference Scheme: a URI *of* the namespace

namespace$_1$ *incorporates* namespace$_2$

Definition: each desgination and fact type form in the namespace$_2$ is in the namespace$_1$, and if the namespace$_1$ is a vocabulary namespace, each attributive namespace within the namespace$_2$ is incorporated into an attributive namespace in the namespace$_1$ for the same subject concept

designation *is in* namespace

Definition: the namespace contains the designation such that the signifier of the designation is the signifier of no other designation in the namespace

Synonymous Form: namespace *contains* designation

Issue 9958 Change form of expression to fact type form
Issue 9257 Add text
Issue 9930 Revise text

fact type form *is in* namespace

~~form of expression *is in* namespace~~

Definition: the namespace contains the fact type form ~~form of expression~~ such that it is distinguishable from every other fact type form ~~form of expression~~ in the namespace

Synonymous Form: namespace *contains* fact type form ~~form of expression~~

Note: The distinguishability of a fact type form from others within a namespace is based on how a use of the fact type form is recognized. Distinguishability considers positions of placeholders, meanings of designations used by placeholders and the expression of the fact type form excluding expressions of placeholders.

Example: The fact type form 'proposition is true' (with placeholder 'proposition') is indistinguishable from '[proposition] is true' (with placeholder '[proposition]') because both placeholders use a designation of the same concept ('proposition'), but those two forms are distinguishable from 'line is true' (with placeholder 'line') because 'proposition' and 'line' designate different concepts.

namespace *has* URI

Definition: the URI uniquely identifies the namespace

Necessity: Each URI *is* the URI *of* at most one namespace.

Issue 9955 Revise text

vocabulary namespace

Definition: namespace that is derived from a vocabulary ~~which includes all of the vocabulary's designations and fact type forms forms of expression that are distinguishable by the uniqueness of signifier or form, and which possibly includes attributive namespaces containing designations and fact type forms forms of expression of the vocabulary~~

Issue 9952 Revise text (note)

attributive namespace

~~role namespace~~

Definition: namespace that contains designations recognizable in the context of being attributed to instances of a particular concept

Necessity: Each attributive namespace *is for* exactly one subject concept.

Reference Scheme: a vocabulary namespace that *includes* the attributive namespace and the subject concept that *has* the attributive namespace

Note: A designation in an attributive namespace typically represents a role of a binary fact type. In English, such a designation can typically be used with any of several attributive forms, such as "... has …" or "… of …". A designation in an attributive namespace can also ~~be for~~

represent a characteristic. Different languages have different attributive forms - different grammatical structures relating a subject to something attributed to it.

Example: Given an attributive namespace for the subject concept 'rental', a designation 'drop-off date' can be used in any of several attributive forms: "rental has drop-off date," "drop-off date of rental," "rental's drop-off date," "drop-off date is of rental," etc.

Example: Given an attributive namespace for the subject concept 'rental', the designation 'assigned' for the characteristic 'rental is assigned' is recognized where it applies to a rental, as in "assigned rental."

attributive namespace *is for* subject concept

~~rolenamespace *is for* subject concept~~

Definition: the designations in the attributive namespace are for concepts attributable to instances of the subject concept

Synonymous Form: concept *has* attributive namespace

subject concept

Definition: concept that provides a context for recognizing designations used to attribute properties to instances of the concept

Concept Type: role

Example: In the phrase, "each rental's drop-off date," the concept 'rental' is a subject concept with respect to recognizing the designation 'drop-off date' representing a role in a fact type that relates a rental to its drop-off date.

Example: In the phrase, "an assigned rental," the concept 'rental' is a subject concept with respect to recognizing the designation 'assigned' representing a characteristic attributable to rentals ('rental is assigned').

attributive namespace *is within* vocabulary namespace

Definition: the attributive namespace is a section of the vocabulary namespace attributable to the concept that has the attributive namespace

Synonymous Form: vocabulary namespace *includes* attributive namespace

language

Definition: system of arbitrary signals (such as voice sounds or written symbols) and rules for combining them as used by a nation, people, or other distinct community

Source: based on AH

Note: A language can be a natural language or an unnatural one, such as a computer language or a system of mathematical symbols.

Note: A language is often identified by its name. ISO provides names of many languages in ISO 639-2 (English) and provides short (at most 3 letters) language-independent codes in ISO 639-2 (Alpha-3 Code).

Example: English, French, German, Arabic

Example: Moroccan Arabic (a dialect of Arabic)

Example: Unified Modeling Language (a graphical modeling language)

Issue 9467 Remove text

vocabulary namespace *is for* language

Definition: each representation in the vocabulary namespace is for expression in the language

~~Synonymous Form:~~ ~~language *is supported by* vocabulary namespace~~

8.4 Reference Schemes

Issue 9930 Replace figure
Issue 9948 Replace figure

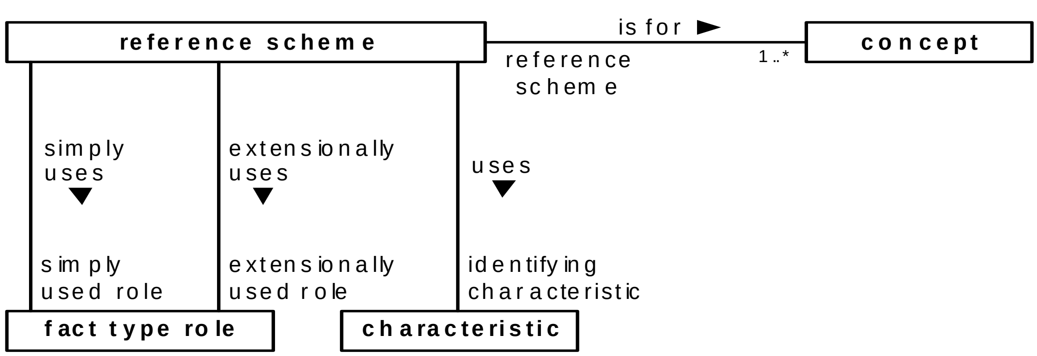


Figure 8.7

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9452 Remove text/Revise text
Issue 10377 Add text
Issue 9948 Change role to fact type role

reference scheme FL

Definition: chosen way of identifying instances of a given concept

Note: A reference scheme is a way of referring to instances of a concept by way of related things that are either lexical or are otherwise identifiable. A reference scheme usually uses one or more fact type roles of binary fact types in order to identify an instance of a concept from facts about the instance. A reference scheme can also use one or more characteristics.

Note: A reference scheme can be partial or complete. It is complete if it can always be used to refer to every instance of a concept. An overall complete reference scheme for a concept can result from there being multiple partial reference schemes for that concept, its more general concepts, and its categories.

Note: Choice of reference schemes must be based on uniqueness (providing an identifier that refers to exactly one thing), but it should consider more than uniqueness. It should also consider permanence – if the actualities considered by the scheme change often, then references can become invalid. A reference scheme should also not lead into an inescapable reference cycle where things only identify each other, but should lead either directly or indirectly to an expression. It should also consider convenience and relevance from a business perspective.

Note: A fact type role is used in a reference scheme in either of two ways. A simple use of a fact type role involves a single instance of the fact type role in each reference based on the scheme. An extensional use of a fact type role involves the entire set of related instances of the fact type role in each reference based on the scheme.

Note: A reference scheme implies that there is uniqueness – that whatever facts are used to reference an individual thing uniquely identify that one thing.

Reference Scheme: the set of ~~each~~ fact type roles that *are simply used by* the reference scheme and the set of ~~each~~ fact type roles that *are extensionally used by* the reference scheme and the set of characteristics that *are used by* the reference scheme

reference scheme *is for* concept

FL

Definition: instances of the concept can be identified using the reference scheme

Synonymous Form: concept *has* reference scheme

Necessity: Each reference scheme *is for* at least one concept.

Issue 9467 Remove text
Issue 10377 Add text
Issue 9930 Add text
Issue 9948

reference scheme *simply uses* fact type role

FL

Definition: any given instance of the fact type role, which is of a binary fact type, serves as identification or partial identification of an instance of the concept having the reference scheme where the given instance is related by way of the binary fact type that has the fact type role

Synonymous Form: reference scheme *has* simply used role

~~Synonymous Form:~~ ~~role *is simply used by* reference scheme~~

Necessity: Each fact type role that *is simply used by* a reference scheme *is in* a binary fact type.

Example: A reference scheme for 'car model' simply uses the 'name' role of the binary fact type 'car model has name'. An example of a reference based on this reference scheme identifies a particular car model as having the name "Chevrolet Cavalier." The meaning of the reference is an individual concept having this definition: the car model that has the name "Chevrolet Cavalier."

Issue 9467 Remove text
Issue 10377 Move and Add text
Issue 9930 Add text

reference scheme *extensionally uses* fact type role FL

Definition:	a set of instances of the fact type role, which is of a binary fact type, serves as identification or partial identification of an instance of the concept having the reference scheme where the set is the set of all instances of the fact type role related by way of the binary fact type that has the fact type role
Synonymous Form:	reference scheme *has* extensionally used role
~~Synonymous Form:~~	~~role *is extensionally used by* reference scheme~~
Necessity:	Each fact type role that *is extensionally used by* a reference scheme *is in* a binary fact type.
Example:	The reference scheme given above for the concept 'reference scheme' itself exemplifies extensional use of roles. Any particular reference scheme can be identified by the combination of what roles it simply uses, what roles it extensionally uses, and what characteristics it uses. For example, the reference scheme for 'car model' (in the example above) is identified by the facts that it simply uses only the 'name' role of the binary fact type 'car model has name', it extensionally uses no roles and it uses no characteristics.

Issue 9467 Remove text
Issue 10377 Replace text/Add text
Issue 9930 Add text

reference scheme *uses* characteristic FL

Definition:	having or not having the characteristic serves as identification or partial identification of an instance of the concept having the reference scheme
Synonymous Form:	reference scheme *has* identifying characteristic
~~Synonymous Form:~~	~~characteristic *is used by* reference scheme~~
Note:	Reference schemes generally use a characteristic only in combination with one or more roles of binary fact types such that facts of those types about any referenced thing reduce the number matching instances down to two, one instance having the characteristic and not the other. A reference scheme using no more than a characteristic works only for the unusual case of a concept that always has at most two instances.
Example:	A concept 'tire position', which has only four instances, has a reference scheme that uses two characteristics, 'tire position is in front' and 'tire position is on the right'. Any of the four positions can be identified by knowing whether or not it is in front and whether or not it is on the right. The meaning of a reference based on this scheme is an individual concept having the more general concept 'tire position' and having a delimiting characteristic that is either being in front or not being in front and another delimiting characteristic that is either being on the right or not being on the right.

8.5 Conceptual Schemas and Models

Issue 9447 Replace figure

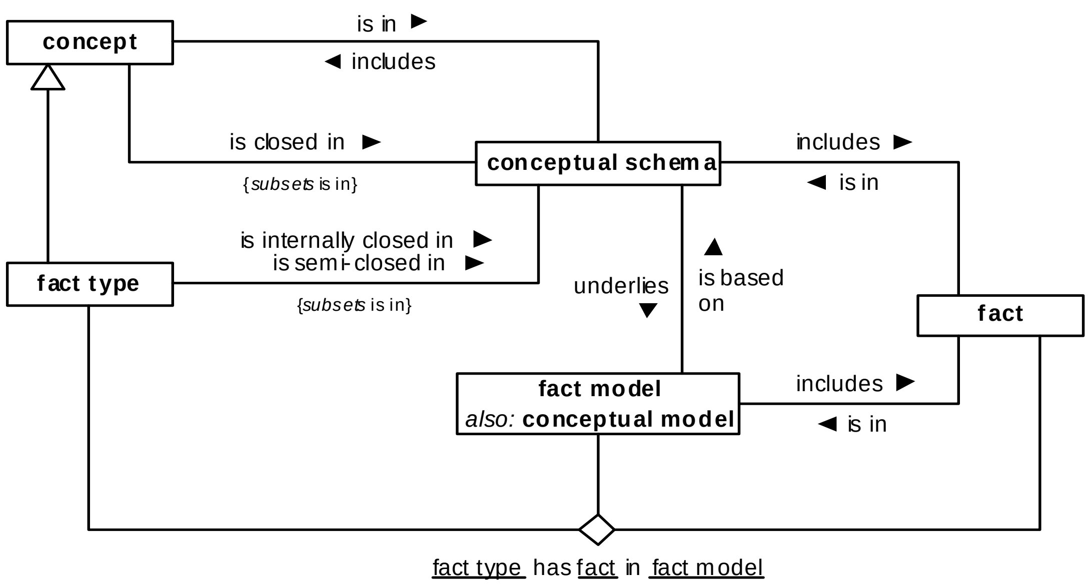


Figure 8.8

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

conceptual schema FL

Definition: combination of concepts and facts (with semantic formulations that define them) of what is possible, necessary, permissible, and obligatory in each possible world

conceptual schema *includes* concept FL

Definition: the concept is used in models based on the conceptual schema

Synonymous Form: concept *is in* conceptual schema

Necessity: Each role *of* each fact type that *is in* a conceptual schema *is in* the conceptual schema.

conceptual schema *includes* fact FL

Definition: the fact determines something possible, necessary, permissible, or obligatory in each possible world that can be modeled based on the conceptual schema

Synonymous Form: fact *is in* conceptual schema

Issue 9447 Remove text/Revise text

fact type *is internally closed in* conceptual schema FL

Definition: in each ~~conceptual~~ fact model based on the conceptual schema, for each instance of the fact type, the ~~conceptual~~ fact model includes a corresponding fact if, for each thing filling any of the fact type's roles in the instance, the ~~conceptual~~ fact model also includes a fact of the existence of that thing

Synonymous Form: fact type *is semi-closed in* conceptual schema

Necessity: Each fact type that *is semi-closed in* a conceptual schema *is in* the conceptual schema.

Note: Open world semantics are assumed by default, but closure may be explicitly asserted for any fact type, on an individual basis, to declare that each ~~conceptual~~ fact model population agrees with that of the fact type's extension in the actual business domain. Semi-closure is with respect to the domain model population of the noun concepts playing a role in the fact type. In other words, if the things participating in a fact are known within a model, then the fact is also known within that model.

Issue 9447 Remove text/Revise text

concept *is closed in* conceptual schema

FL

Definition: in each ~~conceptual~~ fact model based on the conceptual schema, the entire extension of the concept is given in the facts included in the ~~conceptual~~ fact model

Necessity: Each concept that *is closed in* a conceptual schema *is in* the conceptual schema.

Note: A concept can be closed in one conceptual schema and not in another. For example, consider a corporate customer of EU-Rent that adopts several of EU-Rent's concepts. The corporate customer's conceptual schema might have the concept 'rental' as not closed because the customer is not aware of all rentals, but EU-Rent's conceptual schema has the concept as closed.

Issue 9447 Remove text/Revise text

fact model

~~conceptual model~~

FL

Definition: combination of a conceptual schema and, for one possible world, a set of facts (defined by semantic formulations using only the concepts of the conceptual schema)

Synonym: conceptual model

Note: Each necessity of the conceptual schema is satisfied by a ~~conceptual~~ fact model, but obligations are not necessarily satisfied.

Issue 9447 Remove text/Revise text

fact model *is based on* conceptual schema

~~conceptual model *is based on* conceptual schema~~

FL

Definition: the conceptual schema provides the concepts and modal facts of the ~~conceptual~~ fact model

Synonymous Form: conceptual schema *underlies* ~~conceptual~~ fact model

Issue 9447 Remove text/Revise text

fact model ***includes*** **fact**

~~**conceptual model** ***includes*** **fact**~~ FL

Definition: the fact corresponds to an actuality in the possible world modeled by the ~~conceptual~~ fact model

Synonymous Form: fact *is in* ~~conceptual~~ fact model

Issue 9447 Remove text/Revise text

fact type ***has*** **fact** ***in*** **fact model**

~~**fact type** ***has*** **fact** ***in*** **conceptual model**~~ FL

Definition: the fact *is in* the ~~conceptual~~ fact model and the fact *corresponds to* an instance *of* the fact type

8.6 Extensions

Issue 9467 Replace figure
Issue 9948 Replace figure

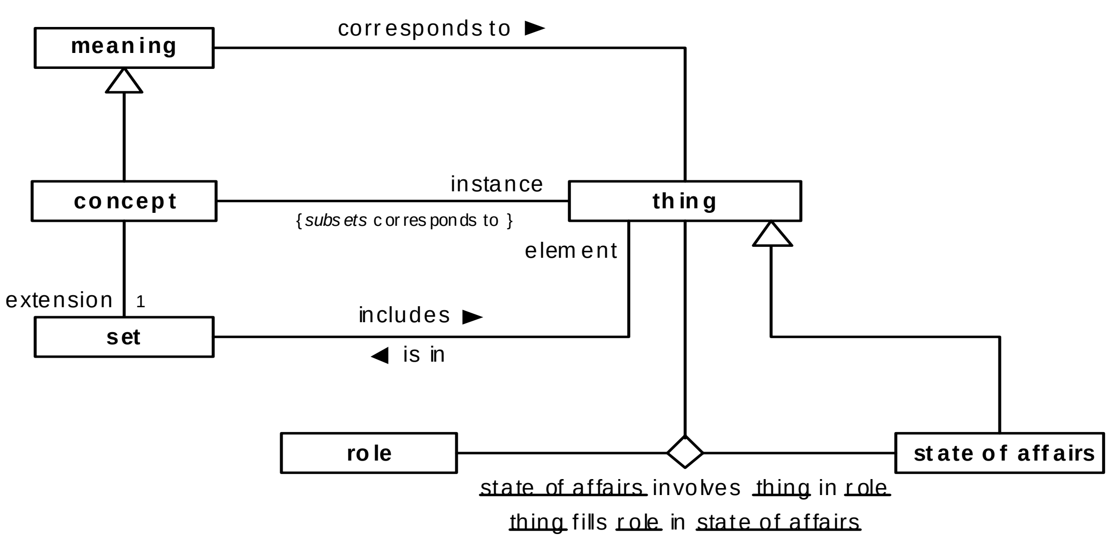


Figure 8.9

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9948 Add text

state of affairs

FL

Definition: event, activity, situation, or circumstance

Reference Scheme: a proposition that *corresponds to* the state of affairs

Note: A state of affairs can be possible or impossible. Some of the possible ones are actualities. A state of affairs is what is denoted by a proposition. A state of affairs either occurs or does not occur, whereas a proposition is either true or false. A state of affairs is not a meaning. It is a thing that exists and can be an instance of a concept, even if it does not happen.

Example: EU-Rent owning 10,000 rental cars is a state of affairs corresponding to the proposition "EU-Rent owns 10,000 rental cars."

Example: It being obligatory that each rental have at most three additional drivers is a state of affairs corresponding to the rule, "Each rental must have at most three additional drivers."

Issue 9948 Add text

actuality

FL

Definition: state of affairs that occurs in the actual world

Note: Actualities are states of affairs that actually happen, as distinct from states of affairs that don't happen but nevertheless exist as subjects of discourse and can be imagined or planned.

Issue 9467 Remove text
Issue 9948 Remove entry and replace with "state of affairs involves thing in role"

~~actuality *involves* thing *in* role~~

~~FL~~

~~Definition:~~ ~~the actuality is an instance of the fact type that has the role and the thing plays the role in that actuality~~

~~Synonymous Form:~~ ~~role *is filled by* thing *in* actuality~~

~~Synonymous Form:~~ ~~thing *fills* role *in* actuality~~

~~Note:~~ ~~This fact type supports talking generically about involvement of things in the instances of fact types.~~

state of affairs *involves* thing *in* role

FL

Definition: the thing plays the role in the state of affairs, and, if the role is a fact type role and the state of affairs is an actuality, the state of affairs is an instance of the fact type that has the role

Synonymous Form: thing *fills* role *in* state of affairs

Note: If the role is an object type, it is necessarily a situational role and the state of affairs is a "situation" for which the role is defined (See 11.1.5).

Note: This fact type is used to capture the fact of involvement of a thing in an actuality that is an instance of a fact type, or more generally, in a state of affairs whether or not it is an actuality.

extension FL

Source: ISO 1087-1 (English) (3.2.8) ['extension']

Definition: totality of objects [every thing] to which a concept corresponds

Concept Type: role

General Concept: set

instance FL

Definition: thing that *is in* an extension *of* a concept

Concept Type: role

Example: The actual City of Los Angeles is an instance of the concept 'city'. It is also the one instance of the individual concept 'Los Angeles'.

8.6.1 Relating Meaning to Extension

Issue 9882 Remove text
Issue 10569 Add FL to entries
Issue 10790 Replace text
Issue 9959 Add text

meaning *corresponds to* thing

Definition: ~~the thing is the actual thing~~ the thing is conceptualized by and is consistent with the meaning

~~Synonymous Form:~~ ~~thing *is conceptualized as* meaning~~

Note: A concept corresponds to each instance of the concept. A proposition corresponds to a state of affairs (which might or might not be actual). A ~~fact~~ proposition that is true corresponds to an actuality.

Note: For some kinds of meanings this is a many-to-many relationship. For others it is many-to-one.

concept *has* extension FL

Definition: the extension is the set of things to which the meaning corresponds

concept *has* instance FL

Definition: the concept *corresponds to* the instance

8.6.2 Necessities Concerning Extension

Issue 9447 Replace text
Issue 9882 Revise Necessity statement
Issue 10569 Revise Necessity statement
Issue 10779 Revise text
Issue 9948 Add Necessity text
Issue 10790 Remove text

The following statements of necessity apply to the relationships between a meaning and its extension. Other necessities stated in the context of the Meaning and Representation Vocabulary concern the contents of conceptual schemas and their representations. But the following necessities concern each ~~conceptual~~ fact model in relation to the conceptual schema that underlies it.

Necessity: Each concept *has* exactly one extension.

Necessity: A thing *is* an instance *of* a concept if and only if the thing *is in* the extension *of* the concept.

~~Necessity: Each individual concept *has* exactly one instance.~~

Necessity: Each instance *of* a fact type *is* an actuality.

Necessity: Each proposition *corresponds to* at most one state of affairs.

Necessity: Each proposition that *is true corresponds to* exactly one actuality.

Necessity: Each actuality that *is* an instance *of* a fact type *involves* some thing *in* each role *of* the fact type.

Necessity: Each thing that *fills* a role *in* an actuality *is* an instance *of* the role.

Necessity: An actuality *is* an instance *of* a fact type if the actuality *involves* a thing *in* a role *of* the fact type.

Necessity: If a concept *incorporates* a characteristic then each instance *of* the concept *is* an instance *of* the role *of* the characteristic.

Necessity: If a concept_1 *is coextensive with* a concept_2 then the extension *of* the concept_1 ~~*equals*~~ *is* the extension *of* the concept_2.

Necessity: Each instance *of* a role that *ranges over* an object type *is* an instance *of* the object type.

Necessity: A thing *is* an instance *of* a fact type role if and only if the thing *fills* the fact type role *in* an actuality.

Necessity: A thing *fills* a fact type role *in* an actuality if and only if the actuality *is* an instance *of* the fact type that *has* the fact type role.

Necessity: Each individual concept that *corresponds to* a thing always *corresponds to* that thing.

8.7 Elementary Concepts

Issue 9344, 9724, 10779Replace figure

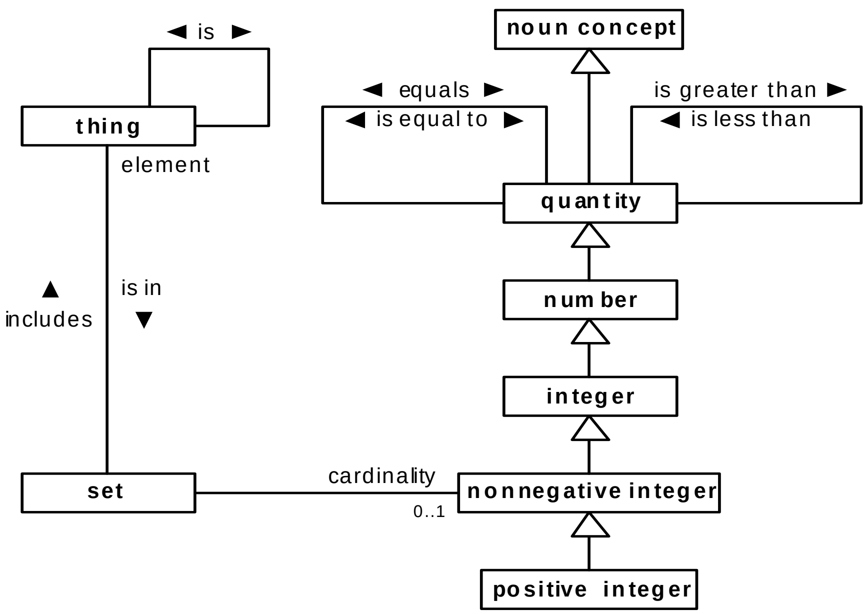


Figure 8.10

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

thing

FL

Source: ISO 1087-1 (English) (3.1.1) ['object']

Definition: anything perceivable or conceivable

Note: Every other concept implicitly specializes the concept 'thing'.

Reference Scheme: an individual concept that *corresponds to* the thing

Issue 10779 Remove text

thing$_1$ *is* thing$_2$

FL

Definition: The thing$_1$ and the thing$_2$ are the same thing

~~Synonymous Form:~~ ~~thing$_1$ *equals* thing$_2$~~

~~Synonymous Form:~~ ~~thing$_1$ *is equal to* thing$_2$~~

~~Synonymous Form:~~ ~~hing$_1$ = thing$_2$~~

set

FL

Definition: collection of zero or more things considered together without regard to order

Issue 9930 Add text

thing *is in* set FL

Definition: the thing is an element of the set

Synonymous Form: set *includes* thing

Synonymous Form: set *has* element

Issue 9724 Add new entries
Issue 11290 Remove text

---

set *has* cardinality FL

Definition: the cardinality is the number of distinct elements in the set

Necessity: Each set *has* at most one cardinality.

cardinality FL

Definition: nonnegative integer that is the number of ~~distinct~~ elements in a given set or collection

Concept Type: role

Issue 9344 Add text
Issue 10568 Add Note to number entry

---

quantity FL

Definition: the aspect in which a thing is measurable in terms of greater, less, or equal [MWU]

General Concept: noun concept

Note: The concept quantity can be elaborated into mathematical systems, such as integers and real numbers, and into systems of measures. This specification elaborates only the concepts for integer, because they are commonly used in structural rules. For measurement systems and units of measure there are accepted vocabularies and perhaps standard ontologies, but the specification of such a vocabulary is beyond the scope of this specification.

Issue 10779 Add new entry

$quantity_1$ *equals* $quantity_2$ FL

Definition: the $quantity_1$ is mathematically equivalent to the $quantity_2$

Synonymous Form: $quantity_1$ *is equal to* $quantity_2$

$quantity_1$ *is less than* $quantity_2$ FL

Definition: the $quantity_1$ is mathematically less than the $quantity_2$

~~Synonymous Form:~~ ~~$quantity_1$ < $quantity_2$~~

Synonymous Form: $quantity_2$ *is greater than* $quantity_1$

~~Synonymous Form:~~ ~~$quantity_2$ > $quantity_1$~~

number

FL

Definition: quantity belonging to an abstract mathematical system and subject to laws of succession, addition, and multiplication

Dictionary Basis: An arithmetical value, expressed by a word, symbol, or figure, representing a particular quantity and used in counting and making calculations [ODE: “number,” 1]

Note: The ISO 6093 Number Namespace has designations for decimal numbers.

Issue 9344 Change text
Issue 10568 Remove Note

---

integer

FL

Definition: number that has ~~with~~ no fractional part

~~Note:~~ ~~The Integer Namespace, in the Namespace Registration Vocabulary, has designations for all of the integers~~

Issue 9344 and 9930 Delete text

~~integer1 *is less than* integer2~~

~~FL~~

~~Definition:~~ ~~The integer1 is numerically less than integer2~~

~~Synonymous Form:~~ ~~integer1 < integer2~~

~~Synonymous Form:~~ ~~integer2 *is greater than* integer1~~

~~Synonymous Form:~~ ~~integer2 > integer1~~

nonnegative integer

FL

Definition: integer that is greater than or equal to zero

positive integer

FL

Definition: nonnegative integer that is not zero

9 Logical Formulation of Semantics Vocabulary

Issue 10423 Replace text
Issue 9721 Replace text
Issue 9712 Revise text
Issue 9930 Replace figure caption to all figures in this Clause
Issue 9952 Revise text

The vocabulary in this clause is not intended for use by business people in general, but rather, it is a vocabulary used to describe the formal semantic structures of busin ess discourse. It is not for discussing business, but for discussing the semantic structures underlying business communications of concepts, facts, and rules. For example, a typical business person does not tend to talk about quantifications, but he expresses quantifications in almost every statement he makes. He doesn't tend to talk about conjunctions, disjunctions, logical negations, ~~conjuncts, disjuncts, negands~~ antecedents and consequents, but these are all part of the formulation of his thinking. The vocabulary in this clause is for talking about these conceptual devices that people use all the time.

Semantic formulations are not representations or expressions of meaning. Rather, they are structures of meaning – the logical composition of meaning.

Business rules are generally expressed in natural language, although some rules are at times illustrated graphically. SBVR does not provide a logic language for restating business rules in some other language that business people don't use. Rather, SBVR provides a means for describing the structure of the meaning of rules expressed in the natural language that business people use. Semantic formulations are not expressions or statements. They are structures that make up meaning. Using SBVR, the meaning of a definition or statement is communicated as facts about the semantic formulation of the meaning, not as a restatement of the meaning in a formal language.

There are two kinds of semantic formulations. The first kind, logical formulation, structures propositions, both simple and complex. Specializations of that kind are given for various logical operations, quantifications, atomic formulations based on fact types and other formulations for special purposes such as objectification and nominalization.

The second kind of semantic formulation is projection. It structures intensions as sets of things that satisfy constraints. ~~with regard to what comprises sets or multisets.~~ Projections formulate definitions, aggregations, and questions. ~~A projection is over a logical formulation of a condition that determines what is included in a resulting set or multiset~~.

Semantic formulations are recursive. Several kinds of semantic formulations embed other semantic formulations. Logic variables are introduced by quantifications (a kind of logical formulation) and projections so that embedded formulations can refer to instances of concepts. A logic variable used in a formulation is free within that formulation if it is not introduced within that formulation. A formulation is closed if no variable is free within it. Only a closed semantic formulation can formulate a meaning. If a formulation has a variable that is free within it, then it can be part of a larger formulation of a meaning (one that introduces the variable) but it does not by itself formulate a meaning.

The hierarchical composition of semantic formulations is seen in the following example of a very simple business rule. The rule is stated in different ways but is one rule having one meaning. Many other statements are possible.

- A rental must have at most three additional drivers.
- It is obligatory that each rental has at most three additional drivers.

Below is a representation of a semantic formulation of the rule above as sentences that convey the full structure of the rule as a collection of facts about it. Note that different semantic formulations are possible for the same meaning. Two semantic formulations can be determined to have the same meaning either by logical analysis or by assertion (as a matter of definition). A single formulation is shown below.

The rule is a proposition meant by an obligation ~~claim~~ formulation.
. That obligation ~~claim~~ formulation embeds a universal quantification.
. . The universal quantification introduces a first variable.
. . . The first variable ranges over the concept 'rental'.
. . The universal quantification scopes over an at-most-n quantification.
. . . The at-most-n quantification has the maximum cardinality 3.
. . . The at-most-n quantification introduces a second variable.
. . . . The second variable ranges over the concept 'additional driver'.
. . . The at-most-n quantification scopes over an atomic formulation.
. . . . The atomic formulation is based on the fact type 'rental has additional river'.
. . . . . The atomic formulation has a role binding.
. . . . . . The role binding is of the role 'rental' of the fact type.
. . . . . . The role binding binds to the first variable.
. . . . . The atomic formulation has a second role binding.
. . . . . . The second role binding is of the role 'additional driver' of the fact type.
. . . . . . The second role binding binds to the second variable.

Note that designations like 'rental' and 'additional driver' ~~are used above to refer to~~ represent concepts. The semantic formulations involve the concepts themselves, so identifying the concept 'rental' by another designation (such as from another language) does not change the formulation.

The indentation in the example shows a hierarchical structure in which a semantic formulation at one level operates on, applies a modality to, or quantifies over one or more semantic formulations at the next lower level. Each kind of logical formulation, including ~~modality claims~~ modal formulations, quantifications, and logical operations, can be embedded in other semantic formulations to any depth and in almost any combination.

Within the one atomic formulation in the example are bindings to two variables. The variables are free within the atomic formulation because they are introduced outside of it (higher in the hierarchical structure). For this reason, the atomic formulation has no meaning. But the obligation ~~claim~~ formulation has a meaning (the rule) and so does the universal quantification within the obligation ~~claim~~ formulation because both are closed. ~~even the logical universal quantification within the obligation claim each has a meaning because each of those two formulations is closed~~.

Semantic formulations are further exemplified for a simple definition of a characteristic, "driver is of age."

Definition: the age of the driver is at least the EU-Rent Minimum Driving Age

Below is a representation of a semantic formulation of the definition. Note that different semantic formulations are possible. A single formulation is shown below.

Issue 10596 Revise text (and add "." before some lines)
Issue 9712 delete the word "set"

The characteristic is defined by a ~~set~~ projection.
. The ~~set~~ projection is on a first variable.
. . The first variable ranges over the concept 'driver'.

. . The first variable maps to the one role of the characteristic.
. The ~~set~~ projection is constrained by a first universal quantification.
. . The first universal quantification introduces a second variable.
. . . The second variable ranges over the concept ‘age’.
. . . The second variable is unitary.
~~. . The first universal quantification is restricted by an atomic formulation.~~
. . . The second variable is restricted by an atomic formulation.
. . . . The atomic formulation is based on the fact type ‘driver has age’.
. . . . The atomic formulation has a role binding.
. . . . . The role binding is of the role ‘driver’ of the fact type.
. . . . . The role binding binds to the first variable.
. . . . The atomic formulation has a second role binding.
. . . . . The second role binding is of the role ‘age’ of the fact type.
. . . . . The second role binding binds to the second variable.
. . The first universal quantification scopes over a second universal quantification.
. . . The second universal quantification introduces a third variable.
. . . . The third variable ranges over the concept ‘EU-Rent Minimum Driving Age’.
. . . . The third variable is unitary.
. . . The second universal quantification scopes over an atomic formulation.
. . . . The atomic formulation is based on the fact type ‘$\text{quantity}_1 \geq \text{quantity}_2$’.
. . . . . The atomic formulation has a role binding.
. . . . . . The role binding is of the role ‘quantity_1’ of the fact type.
. . . . . . The role binding binds to the second variable.
. . . . . The atomic formulation has a second role binding.
. . . . . . The second role binding is of the role ‘quantity_2’ of the fact type.
. . . . . . The second role binding binds to the third variable.

The ~~set~~ projection that defines the characteristic is on a single variable. A ~~set~~ projection defining a binary fact type is on two variables, one mapped to each role. Note that the definition of the characteristic above uses two binary fact types, but all of the roles of those fact types are bound to variables introduced by the projection or by formulations within in, so the projection is closed and conveys a meaning.

SBVR does not attempt to provide special semantic formulations for tenses or the variety of ways states and events can relate to each other with respect to time or can be related to times, periods, and durations. However, an objectification is a logical formulation that enables a state or event indicated propositionally to be the subject or object of other propositions. An encompassing formulation can relate a state or event indicated using objectification to points in time, periods, and durations, or to another state or event (possibly also identified using objectification) with respect to time (e.g., occurring after or occurring before). The specific relations of interest can be defined as fact types. SBVR’s treatment of time in relation to states and events allows temporal relations to be defined generically and orthogonally to the many fact types whose extensions change over time.

A propositional nominalization is similar to objectification. It is a kind of logical formulation that structures the meaning represented by a mention of a statement or proposition as opposed to a use of it. Other similar types of formulations structure meanings represented by mention of concepts, questions, and answers. Furthermore, rules about change often involve noun concept nominalizations, which are special formulations that allow a concept to be a subject or object of a proposition in much the same way that proposition nominalization allows a proposition to be a subject or object. F~~urthermore, rules about change~~

~~often involve concept formulations, which are special formulations that allow concepts to be a subject or object of a proposition in much the same way that proposition nominalization allows propositions to a subject or object.~~

Issue 9713 Add text

Semantic formulations are structures, and as such, are identified structurally as finite directed graphs. The reference schemes for semantic formulations and their parts take into account their entire structure. In some cases, a transitive closure of a reference scheme shows partial loops (partial in the sense that only a part of a reference scheme loops back, never all of it). This approach allows parts of a closed formulation to be identified by what it is in its particular context while, at the same time, contributing to the unique identity of the formulation that contains it.

Logical Formulation of Semantics Vocabulary

Language:	English
Included Vocabulary:	Meaning and Representation Vocabulary

9.1 Semantic Formulations

Issue 9960 Add "See Annex H." to all figures in this clause.

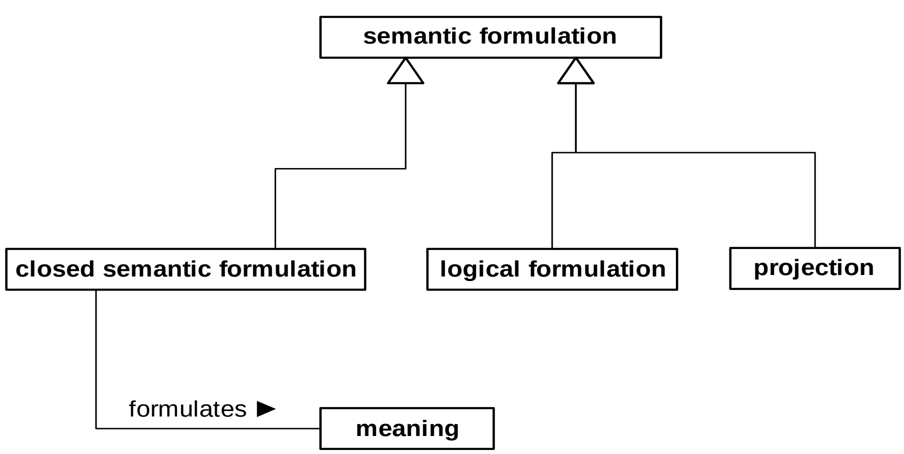


Figure 9.1

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9729 Add Note

semantic formulation FL

Definition: conceptual structure of meaning

Note: The definitions of several specializations of 'semantic formulation' explain what meaning is formulated. A meaning is directly formulated only for a closed semantic formulation. In the case of variables being free within a semantic formulation, a meaning is formulated with respect there being exactly one referent thing given for each free variable.

closed semantic formulation FL

Definition: semantic formulation that *includes* no variable *without binding*

Issue 9467 Remove text

closed semantic formulation ***formulates*** **meaning**

Definition: the meaning is structured by the closed semantic formulation

~~Synonymous Form:~~ ~~meaning *is formulated by* closed semantic formulation~~

Issue 9753 Change heading numbers: 9.1.1 becomes 9.2, 9.1.2 becomes 9.3

9.2 Logical Formulations

Issue 9948 Replace figure

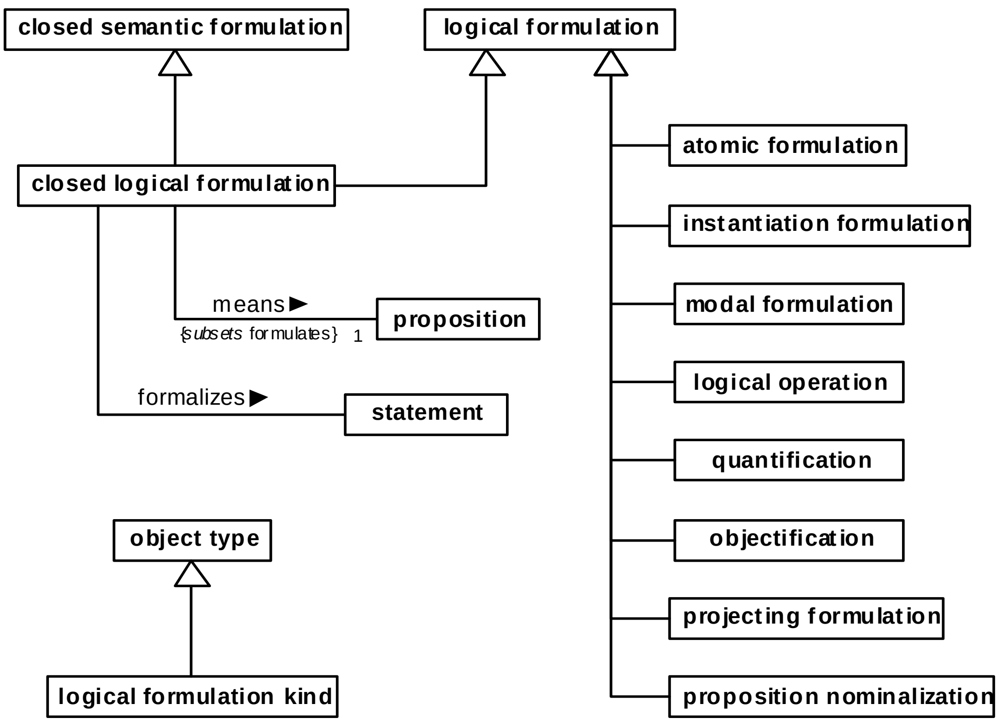


Figure 9.2

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9729 Remove text, Replace Definition
Issue 9948 Replace noun concept text

logical formulation FL

Definition: semantic formulation that formulates a proposition

~~Note:~~ ~~Predicate and modal logics are covered. Mathematical formulations are based on mathematical fact types.~~

Necessity: Each logical formulation *is* an instance *of* exactly one logical formulation kind.

logical formulation kind FL

Definition: object type ~~noun concept~~ that *specializes* the concept 'logical formulation' and that classifies a logical formulation based on the presence or absence of a main logical operation or quantification

Note: The absence of a main logical operator occurs for an atomic formulation or instantiation formulation.

Example: logical negation, conjunction, universal quantification

Issue 10569 Add Necessity (from 8.3.3)

closed logical formulation FL

Definition: logical formulation that *is* a closed semantic formulation

Necessity: Each meaning *formulated by* a closed logical formulation *is* a proposition.

Necessity: Each closed logical formulation *means* exactly one proposition.

Necessity: Each closed logical formulation that *formalizes* a statement *means* the proposition that *is expressed by* the statement.

Issue 9467 Remove text
Issue 10622 Add text

closed logical formulation *means* proposition FL

Definition: the closed logical formulation *formulates* the proposition

~~Synonymous Form:~~ ~~proposition *is meant by* closed logical formulation~~

closed logical formulation *formalizes* statement FL

Definition: the closed logical formulation *means* the proposition that *is expressed by* the statement and the closed logical formulation refers to the concepts represented in the statement

~~Synonymous Form:~~ ~~statement *is formalized by* closed logical formulation~~

Example: If 'barred driver' is defined as "person that must not drive a car," then the statements "Ralph is a barred Driver" and "Ralph is a person that must not drive a car" express the same proposition. But those two statements are formalized differently: one in reference to 'barred driver' and the other in reference to 'person', 'car', and 'person drives car'. The two formulations are different but mean the same proposition.

9.2.1 Variables and Bindings

Issue 9586 Replace figure
Issue 10570 Replace figure
Issue 10596 Replace figure

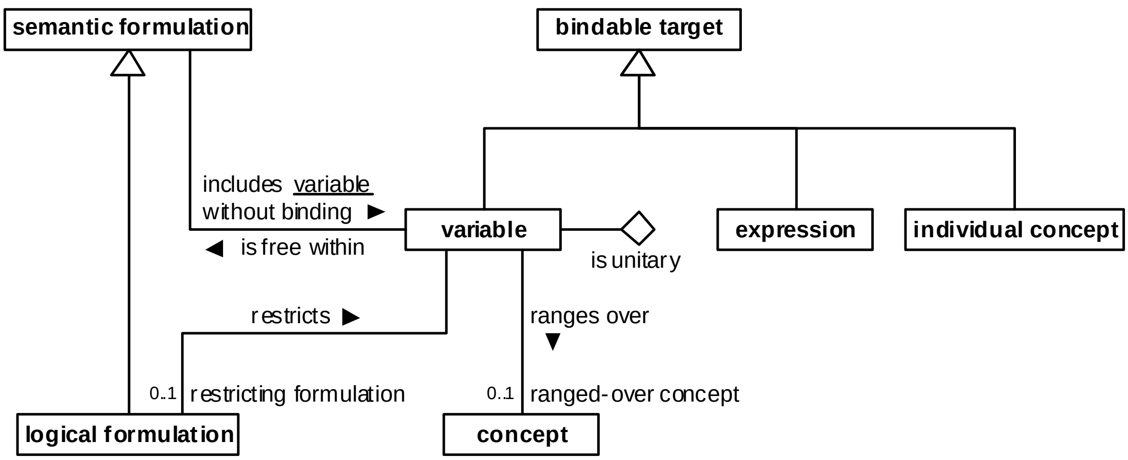


Figure 9.3

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 10596 Add Note, Necessity, and Reference Scheme text and revise text

variable

FL

Definition: reference to an element of a set, whose referent may vary or is unknown

Note: The set of referents of a variable is defined by the two fact types 'variable *ranges over* concept' and 'logical formulation *restricts* variable'. The set is limited to instances of the concept, if given. If the variable is restricted by a logical formulation, the set is further limited to those things for which the meaning formulated by that logical formulation is true when the thing is substituted for each occurrence of the variable in the formulation. If there is no concept and no restricting logical formulation the set includes every thing.

Necessity: Each variable *ranges over* at most one concept.

Necessity: Each variable *is restricted by* at most one logical formulation.

Issue 9462 Remove text
Issue 9713 Change text

Reference Scheme: a quantification that *introduces* the variable and the set of ~~each~~ concepts that *are ranged over by* the variable and the set of logical formulations that *restrict* the variable and whether the variable *is unitary*

~~Reference Scheme: a projection that *is over* the variable and the set of each concepts that *are ranged over by* the variable and whether the variable *is unitary*~~

Reference Scheme: a projection that *is on* the variable and a projection position *of* the variable and the set of concepts that *are ranged over by* the variable and the set of logical formulations that *restrict* the variable and whether the variable *is unitary*.

Issue 10423 Remove text
Issue 10596 Remove text
Issue 9930 Add text

variable *ranges over* concept

FL

Definition: each referent of the variable is ~~necessarily~~ an instance of the concept

~~Synonymous Form:~~ ~~concept *is ranged over by* variable~~

Synonymous Form: variable *has* ranged-over concept

Issue 10596 Add new entry
Issue 9930 Add text

logical formulation *restricts* variable

Definition: for each referent of the variable, the meaning formulated by the logical formulation is true when the referent is substituted for each occurrence of the variable in the logical formulation

Synonymous Form: variable *has* restricting formulation

Note: The meaning of the logical formulation is true for every actual referent of the variable. The things for which the meaning of the logical formulation is false are not considered to be referents of the variable.

Note: A logical formulation restricts a variable in the same way that a concept ranged over by the variable restricts the variable. It limits what the variable refers to. A restrictive clause in a statement is generally formulated as a logical formulation that restricts a variable. A variable restricted by a logical formulation is, except in rare cases, a free variable of the logical formulation.

Example: "Each rental car that is inoperable is unavailable." In the formulation below, a variable ranges over the concept 'rental car' and is restricted by an atomic formulation based on the fact type 'vehicle is inoperable'. Referents of the variable are thereby restricted to being rental cars and to being vehicles that are inoperable.

Example: The proposition is meant by a universal quantification.
. The universal quantification introduces a variable.
. . The variable ranges over the concept 'rental car'.
. . The variable is restricted by an atomic formulation.
. . . The atomic formulation is based on the fact type 'vehicle is inoperable'.
. . . . The 'vehicle' role is bound to the variable.
. The universal quantification scopes over an atomic formulation.
. . The atomic formulation is based on the fact type 'rental car is unavailable'.
. . . The 'rental car' role is bound to the variable.

variable *is unitary*

FL

Definition: the variable is meant to have exactly one referent in the context where the variable is introduced

Note: This characteristic is used particularly in the formulation of definite descriptions.

If a set projection is on one variable and that variable is unitary, then the projection is meant to have exactly one result. For any other projection on a unitary variable, the projection is meant to have one referent for that variable for each combination of referents of other variables (including auxiliary variables) in the same projection.

If a unitary variable is introduced by a universal quantification, the variable ranges over a concept and is restricted by a logical formulation, then the quantification is satisfied if:

1. the unitary variable has exactly one referent, an instance of the concept, for which the restricting logical formulation is satisfied.

2. the logical formulation that the universal quantification scopes over is also satisfied for that one referent.

An exactly-one quantification introducing a non-unitary variable is satisfied differently:

1. the variable has at least one referent, an instance of the concept, for which the restricting logical formulation is satisfied.

2. the logical formulation that the exactly-one quantification scopes over is satisfied for exactly one referent from 1 above.

Example: Given the individual concept 'London-Heathrow Branch' defined as "the EU-Rent branch located at London-Heathrow Airport," the definition can be formulated as a projection on a variable that ranges over the concept 'EU-Rent branch'. The variable is unitary indicating the sense of the definite article "the." Based on this formulation, the concept 'London-Heathrow Branch' is understood to be an individual concept. If the variable is not made unitary, then the formulation captures only the characteristic of being located at London-Heathrow Airport without any indication of the intended meaning that there is exactly one such branch.

Example: A sensible projection formulating "the renter of a given rental" is on a unitary variable (renter) and has an auxiliary variable (rental). The rental variable being unitary indicates there is exactly one renter for each rental. But a set projection formulating "renter of at least one rental" is not on a unitary variable because the variable for rental is introduced within the logical formulation that constrains the projection and not by the projection itself. The projection result can include multiple renters and does not relate these to particular rentals.

Example: A possible formulation of the rule, "The pick-up location of each rental must be an EU-Rent branch," has a variable for 'pick-up location' that is unitary with respect to each rental as indicated by the use of the definite article "the." The possible formulation is an obligation ~~claim~~ formulation that embeds a universal quantification introducing a variable ranging over the concept "rental" and that embeds a second universal quantification introducing a second variable which is restricted by an atomic formulation based on the fact type '<u>rental</u> has pick-up location'. That second variable is unitary indicating that exactly one pick-up location is meant for each rental. The second universal quantification scopes over a formulation of the pick-up location being an EU-Rent branch. The overall formulation applies the obligation ~~claim~~ formulation to the pick-up location being an EU-Rent branch. It does not apply the obligation ~~claim~~ formulation to there being one pick-up branch per rental, which is understood structurally as what is meant in the expression of the rule and not part of the obligation.

Note that if the universal quantifications of the formulation above are reversed such that a quantification introducing the variable for 'pick-up location' embeds the quantification introducing the variable for 'rental', then the variable for 'pick-up rental' is not unitary because it would have multiple referents (one for each distinct pick-up location). Such a formulation would not properly capture the sense of the rule statement.

variable *is free within* semantic formulation
FL

Definition: the semantic formulation employs the variable, but does not introduce it

Synonymous Form: semantic formulation *includes* variable *without binding*

Issue 9586 Replace text
Issue 10570 Replace text / Add text
Issue 9730 Add text to Note

---

bindable target
FL

Definition: variable, expression or individual concept

Note: The meaning of binding to a variable from a logical formulation, such as an atomic formulation, is that a referent of the variable is the thing involved in or considered by the formulation.

Note: The meaning of binding to an individual concept from a logical formulation is that the formulation refers to the one instance of the individual concept. A difference between binding to an individual concept and binding to a variable that ranges over the individual concept is that a variable can be further restricted by a logical formulation giving it the possibility of refering to nothing.

Note: The meaning of binding to an expression (such as a text or graphic) from a logical formulation is that the formulation refers to the expression itself without regard to any meaning the expression might have.

Example: “The text ‘EU-Rent’ is inscribed on each EU-Rent vehicle.” A logical formulation of this proposition involves a binding to the text “EU-Rent,” which simply refers to that expression, not to the individual concept ‘EU-Rent’ nor to any representation of it. The logical formulation also involves a binding to a variable that ranges over the concept ‘EU-Rent vehicle’.

The proposition is meant by a universal quantification.
. The universal quantification introduces a variable.
. . The variable ranges over the concept ‘EU-Rent vehicle’.
. The universal quantification scopes over an atomic formulation.
. . The atomic formulation is based on the fact type
‘expression is inscribed on object’.
. . . The ‘expression’ role is bound to the text “EU-Rent.”
. . . The ‘object’ role is bound to the variable

Example: “The logo EU Rent is inscribed on each EU-Rent vehicle.” This example is the same as the one above except that the ‘expression’ role is bound to the logo EU Rent .

~~Note: The meaning of binding to an individual concept from a logical formulation is that the formulation refers to the one instance of the individual concept.~~

9.2.2 Atomic Formulations

Issue 10423 Replace figure
Issue 9948 Replace figure

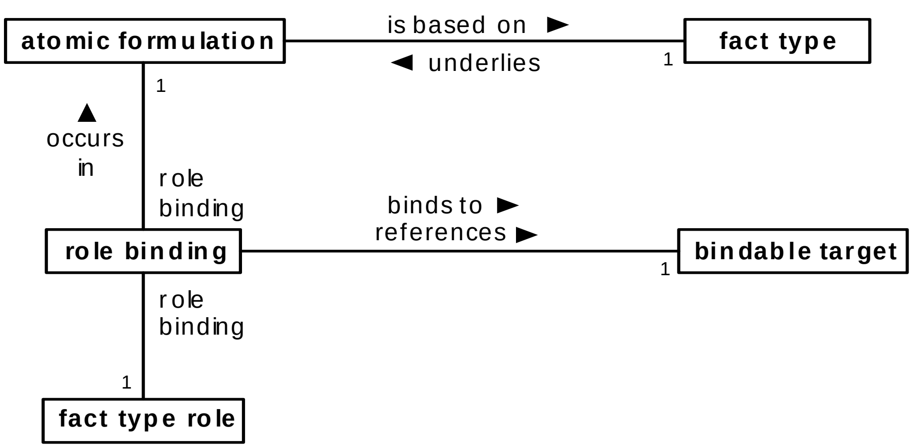


Figure 9.4

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9729 Replace Definition, Example
Issue 9462 Remove text/Revise text
Issue 9948 Replace text

atomic formulation FL

Definition: logical formulation that is based on a fact type and that has a role binding of each role of the fact type and that formulates the meaning: there is an actuality that involves in each role of the fact type the thing to which the bindable target of the corresponding role binding refers

Concept Type: logical formulation kind

Necessity: Each atomic formulation *is based on* exactly one fact type.

Reference Scheme: the set of ~~each~~ role bindings *of* the atomic formulation

Note: The meaning invoked by an atomic formulation puts each referent of each role binding in its respective fact type role. Where a fact type role ranges over ~~specializes~~ some other concept, that meaning implies (as a separate secondary meaning) that the referent of the role binding for that role is an instance of the other concept.

Example: "EU-Rent purchases from General Motors Company."
The statement is formulated by an atomic formulation.
. The atomic formulation is based on the fact type 'company purchases from vendor'.
. The atomic formulation has a first role binding.
. . The first role binding is of the role 'company' of the fact type.

. . The first role binding binds to the individual concept 'EU-Rent'.
. The atomic formulation has a second role binding.
. . The second role binding is of the role 'vendor' of the fact type.
. . The second role binding binds to the individual concept 'General Motors Company'.

atomic formulation *has* role binding

FL

Definition: the atomic formulation includes the role binding for a particular role of the fact type that is the basis of the atomic formulation

Synonymous Form: role binding *occurs in* atomic formulation

atomic formulation *is based on* fact type

FL

Definition: the meaning invoked by the atomic formulation is that of the fact type

Synonymous Form: fact type *underlies* atomic formulation

Issue 9948 Replace text

---

role binding

FL

Definition: connection of an atomic formulation to a bindable target

Necessity: Each role binding *occurs in* exactly one atomic formulation.

Necessity: Each role binding *is of* a role *of* the fact type that *underlies* the atomic formulation that *has* the role binding.

Necessity: Each role binding *binds to* exactly one bindable target.

Necessity: Each role binding *is of* exactly one fact type role.

Necessity: Each variable that *is referenced by* a role binding *of* an atomic formulation *is free within* the atomic formulation.

Reference Scheme: the bindable target that *is referenced by* the role binding and the fact type role that *has* the role binding

---

Issue 10423 Replace text
Issue 10570 Remove text
Issue 9948 Revise definition

---

role binding *binds to* bindable target

FL

Definition: the bindable target provides what thing fills the fact type role that has the role binding in the meaning formulated by the atomic formulation that has the role binding

Synonymous Form: role binding *references* bindable target

~~Note:~~ ~~The meaning of a role binding to a variable is that a referent of the variable is the thing involved in the role. The meaning of a role binding to a text is that the thing involved in the role is the text itself.~~

Issue 9938 Replace text
Issue 9948 Revise text

---

fact type role *has* role binding

FL

Definition: the role binding is a binding of the fact type role, which is of the fact type that underlies an atomic formulation

9.2.3 Instantiation Formulations

Issue 10569 Replace figure

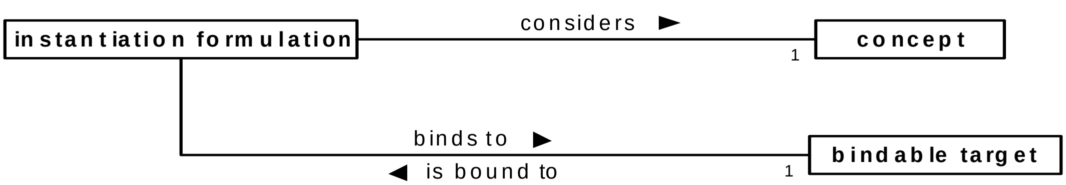


Figure 9.5

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9729 Replace Definition, Example

instantiation formulation

FL

Definition: logical formulation that considers a concept and binds to a bindable target and that formulates the meaning: the thing to which the bindable target refers is an instance of the concept

Concept Type: logical formulation kind

Necessity: Each instantiation formulation *considers* exactly one concept.

Necessity: Each instantiation formulation *binds to* exactly one bindable target.

Necessity: Each variable that *is bound to* an instantiation formulation *is free within* the instantiation formulation.

Reference Scheme: the bindable target that *is bound to* the instantiation formulation and the concept that *is considered by* the instantiation formulation

Note: An instantiation formulation is equivalent to an existential quantification that introduces a variable ranging over the concept considered by the instantiation formulation and that scopes over an atomic formulation based on the fact type 'thing *is* thing' where one role binding is to the variable and the other is to the bindable target bound to the instantiation formulation.

Example: "EU-Rent is a car rental company."
The statement is formulated by an instantiation formulation.
. The instantiation formulation considers the concept "car rental company."
. The instantiation formulation binds to the individual concept 'EU-Rent'.

Issue 9467 Remove text

instantiation formulation *considers* concept

Definition: the instantiation formulation classifies things to be an instance of the concept

~~Synonymous Form:~~ ~~concept *is considered by* instantiation formulation~~

instantiation formulation *binds to* bindable target

Definition: the bindable target indicates what thing is being classified by the instantiation formulation

Synonymous Form: bindable target *is bound to* instantiation formulation

9.2.4 Modal Formulations

Issue 9721 Replace figure, Replace text

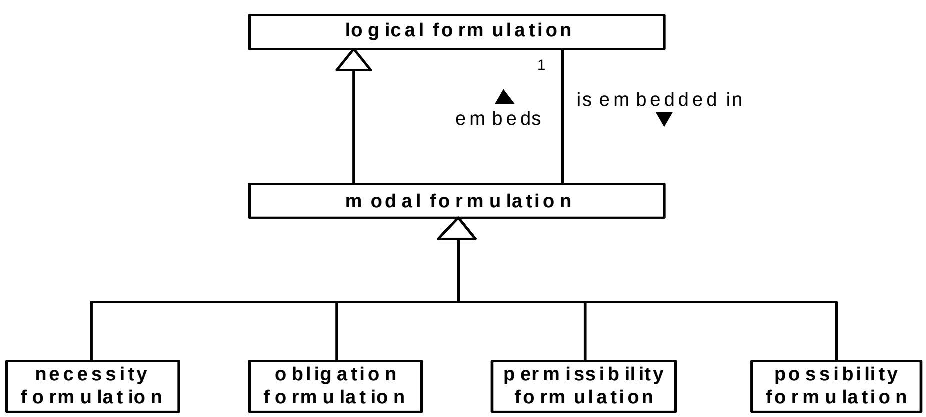


Figure 9.6

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

modal formulation

Definition: logical formulation that formulates that the meaning of another logical formulation has a particular relationship to possible worlds or to acceptable worlds
~~applies a modality to an another logical formulation~~

~~Necessity:~~ ~~Each modal formulation *claims* exactly one modality.~~

Necessity: Each modal formulation *embeds* exactly one logical formulation.

Necessity: Each variable that *is free within* a logical formulation that *is embedded in* a modal formulation *is free within* the modal formulation.

~~Note:~~ ~~The meaning of a modal formulation is the proposition that the proposition meant by the embedded logical formulation is an instance of the modality claimed by the modal formulation.~~

Example: "EU-Rent may purchase from General Motors Company." The statement is formulated by a permissibility formulation (a kind of modal formulation) that embeds the entire formulation shown in the previous subclause in the example under 'atomic formulation' - the formulation of "EU-Rent purchases from General Motors Company." The meaning of the permissibility formulation is that EU-Rent purchases from General Motors Company in some possible world.

Issue 9467 Remove text
Issue 9721 Remove entry

~~**modal formulation *claims* modality**~~ ~~FL~~

~~Definition:~~ ~~the modal formulation *relates* to the modality that is claimed to be applicable to the embedded logical formulation~~

~~Synonymous Form:~~ ~~modality *is claimed by* modal formulation~~

Issue 9721 Revise text

---

modal formulation *embeds* logical formulation FL

Definition: ~~the modality claimed by the modal formulation is applied to the logical formulation~~ the modal formulation formulates that the meaning of the logical formulation has a particular relationship to possible worlds or to acceptable worlds

Synonymous Form: logical formulation *is embedded in* modal formulation

necessity formulation
~~**necessity formulation claim**~~ FL

Definition: ~~modal formulation that *claims* the modality 'necessity'~~ modal formulation that formulates that the meaning of its embedded logical formulation is true in all possible worlds

Concept Type: logical formulation kind

Reference Scheme: the logical formulation that *is embedded in* the necessity formulation ~~claim~~

Issue 10523 Add text

---

obligation formulation
~~**obligation formulation claim**~~ FL

Definition: ~~modal formulation that *claims* the modality 'obligation'~~ modal formulation that formulates that the meaning of its embedded logical formulation is true in all acceptable worlds

Concept Type: logical formulation kind

Reference Scheme: the logical formulation that *is embedded in* the obligation formulation ~~claim~~

Example: A rental may be open only if an estimated rental charge is provisionally charged for the rental." The same rule can be stated this way: "It is prohibited that a rental is open if an estimated rental charge is not provisionally charged for the rental." Both statements can be formulated in the same way:

The rule is a proposition meant by an obligation formulation.
. The obligation formulation embeds a logical negation
. . The logical operand of the logical negation is a universal quantification.
. . . The universal quantification introduces a first variable.
. . . . The first variable ranges over the concept ‘rental’.
. . . The universal quantification scopes over an implication.
. . . . The consequent of the implication is an atomic formulation.
. . . . . The atomic formulation is based on the fact type ‘rental is open’.
. . . . . . The ‘rental’ role is bound to the first variable.
. . . . The antecedent of the implication is an existential quantification.
. . . . . The existential quantification introduces a second variable.
. . . . . . The second variable ranges over the concept ‘estimated rental charge’.
. . . . . The existential quantification scopes over a logical negation.
. . . . . . The logical operand of the logical negation is an atomic formulation.
. . . . The atomic formulation is based on the fact type
‘estimated rental charge is provisionally charged for rental’.
. . . . The ‘estimated rental charge’ role is bound to the second variable.
. . . . The ‘rental’ role is bound to the first variable.

Issue 9721 Revise text

---

permissibility formulation

~~permissibility formulation claim~~ FL

Definition: ~~modal formulation that *claims* the modality ‘permissibility’~~
modal formulation that formulates that the meaning of its embedded logical formulation is true in some acceptable world

Concept Type: logical formulation kind

Reference Scheme: the logical formulation that *is embedded in* the permissibility formulation ~~claim~~

possibility formulation

~~possibility formulation claim~~ FL

Definition: ~~modal formulation that *claims* the modality ‘possibility’~~
modal formulation that formulates that the meaning of its embedded logical formulation is true in some possible world

Concept Type: logical formulation kind

Reference Scheme: the logical formulation that *is embedded in* the possibility formulation ~~claim~~

~~9.2.5 Logical Operands~~

Issue 9258 Relocate text/Remove text after inconsequent entry

---

~~logical operand 1~~ ~~FL~~

~~Definition:~~ ~~logical operand that is the first of at least two operands to a logical operation~~

~~Concept Type:~~ ~~role~~

~~Necessity: Each logical operation *has* at most one logical operand 1.~~

~~**logical operand 2**~~ ~~FL~~

~~Definition: logical operand that is the second of at least two operands to a logical operation~~

~~Concept Type: role~~

~~Necessity: Each logical operation *has* at most one logical operand 2.~~

~~**antecedent**~~ ~~FL~~

~~Definition: logical operand that is the condition considered by a logical operation such as an implication (e.g., what is meant by the *p* in "if *p* then *q*")~~

~~Concept Type: role~~

~~**condition 1**~~ ~~FL~~

~~Definition: logical operand 1 that is the first of two operands to an equivalence~~

~~Concept Type: role~~

~~**condition 2**~~ ~~FL~~

~~Definition: logical operand 2 that is the second of two operands to an equivalence~~

~~Concept Type: role~~

~~**conjunct 1**~~ ~~FL~~

~~Definition: logical operand 1 that is the first of two operands to a conjunction~~

~~Concept Type: role~~

~~**conjunct 2**~~ ~~FL~~

~~Definition: logical operand 2 that is the second of two operands to a conjunction~~

~~Concept Type: role~~

~~**consequent**~~ ~~FL~~

~~Definition: logical operand that is the implied or result operand to a logical operation such as an implication (e.g., what is meant by the *q* in "if *p* then *q*")~~

~~Concept Type: role~~

~~**disjunct 1**~~ ~~FL~~

~~Definition: logical operand 1 that is the first of two operands to a disjunction~~

~~Concept Type: role~~

~~**disjunct 2**~~ ~~FL~~

~~Definition: logical operand 2 that is the second of two operands to a disjunction~~

~~Concept Type: role~~

~~**exclusive disjunction 1**~~ ~~FL~~

~~Definition: logical operand 1 that is the first of two operands to an exclusive disjunction~~

~~Concept Type: role~~

~~**exclusive disjunction 2**~~ ~~FL~~

~~Definition: logical operand 2 that is the second of two operands to an exclusive disjunction~~

~~Concept Type:~~ ~~role~~

~~inconsequent~~ ~~FL~~

~~Definition:~~ ~~logical operand that is an operand irrelevant to the logical result of a logical operation such as of a whether-or-not formulation~~

~~Concept Type:~~ ~~role~~

~~negand~~ ~~FL~~

~~Definition:~~ ~~logical operand that is the operand of a logical negation~~

~~Concept Type:~~ ~~role~~

9.2.5 Logical Operations

Issue 9258 Replace figure

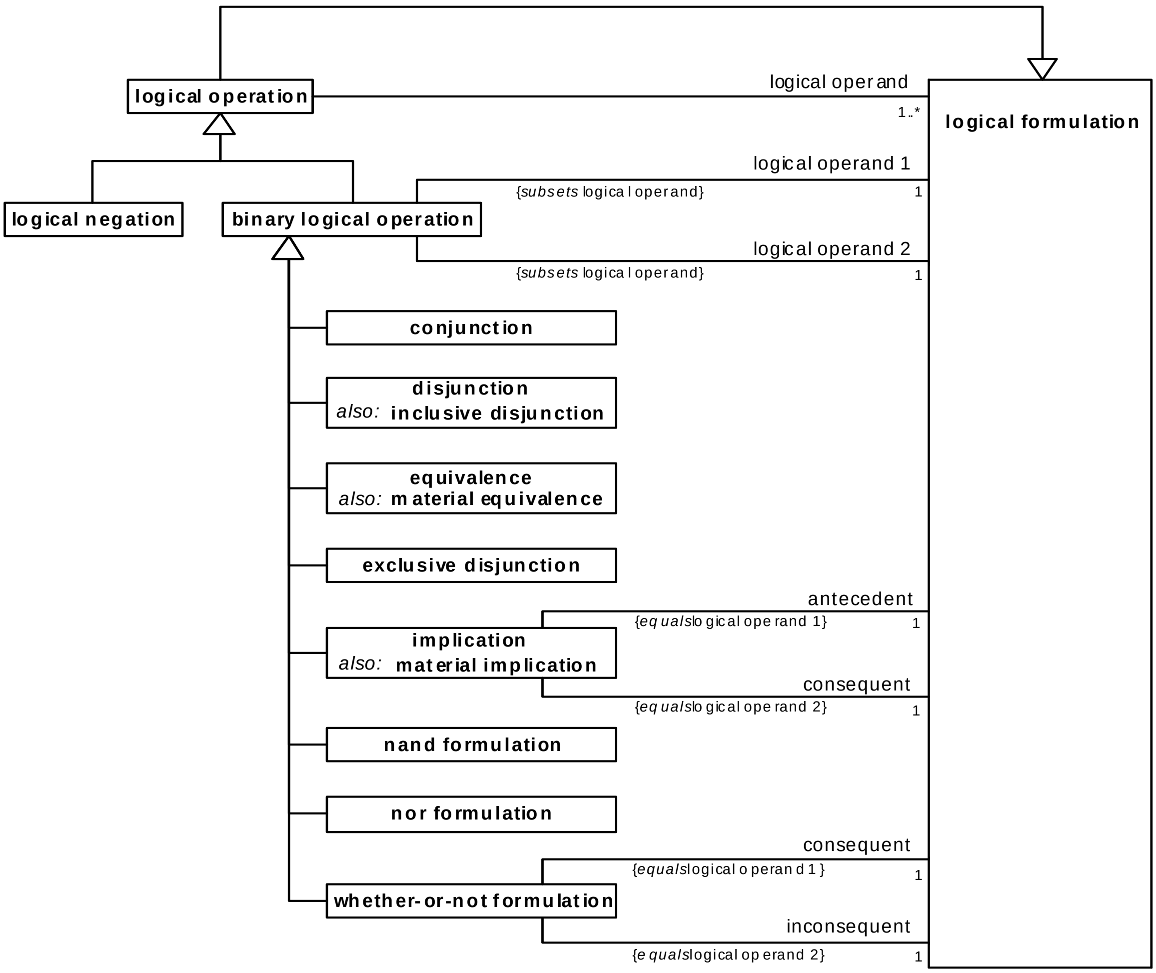


Figure 9.7

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9709 Replace Definition text

logical operation

FL

Definition: logical formulation that formulates a meaning based on only the truth or falseness of the meanings of one or more other logical formulations (its logical operands).

Necessity: Each logical operation *has* at least one logical operand.

Necessity: Each variable that *is free within* a logical operand *of* a logical operation *is free within* the logical operation.

Issue 9258 Move from 9.2.5 Logical Operands, Revise Definition text

logical operand FL

Definition: logical formulation upon which a given logical operation operates

Concept Type: role

logical operation *has* logical operand FL

Definition: the logical operation operates on the logical operand

Issue 9258 Add text
Issue 11288 Replace text

binary logical operation FL

Definition: ~~logical formulation~~ logical operation that operates on two logical operands

Necessity: Each binary logical operation *has* exactly one logical operand 1.

Necessity: Each binary logical operation *has* exactly one logical operand 2.

Note: Distinct roles are defined for the two operands of a binary logical operation even though there is no significant difference between the roles for some operations, such as for conjunction. The one distinction that remains, however, is that the roles are distinct from each other, and this distinction is important where an operation has the same logical formulation filling both roles, such as in ‘*p* and *p*’ or ‘*p* if and only if *p*’.

Issue 9258 Move from 9.2.5 Logical Operands

logical operand 1 FL

Definition: logical operand that is the first of at least two operands to a logical operation

Concept Type: role

Necessity: Each logical operation *has* at most one logical operand 1.

logical operand 2 FL

Definition: logical operand that is the second of at least two operands to a logical operation

Concept Type: role

Necessity: Each logical operation *has* at most one logical operand 2.

Issue 9258 Add text

binary logical operation *has* logical operand 1 FL

Definition: the binary logical operation operates on the logical operand 1

binary logical operation *has* logical operand 2 FL

Definition: the binary logical operation operates on the logical operand 2

Issue 9258 Revise text/Remove text

---

conjunction FL

Definition: binary logical operation that formulates that the meaning of each of its logical operands is true

Concept Type: logical formulation kind

~~Necessity: Each conjunction *has* exactly one conjunct 1.~~

~~Necessity: Each conjunction *has* exactly one conjunct 2.~~

Reference Scheme: the logical operand 1 *of* the conjunction and the logical operand 2 *of* the conjunction

---

Issue 9258 Remove text

~~conjunction *has* conjunct 1~~ ~~FL~~

~~Definition: the conjunction operates on the conjunct 1~~

~~Synonymous Form: conjunction *has* logical operand 1~~

~~conjunction *has* conjunct 2~~ ~~FL~~

~~Definition: the conjunction operates on the conjunct 2~~

~~Synonymous Form: conjunction *has* logical operand 2~~

Issue 9258 Revise text/Remove text

---

disjunction FL

Definition: binary logical operation that formulates that the meaning of at least one of its logical operands is true

Concept Type: logical formulation kind

Synonym: inclusive disjunction

~~Necessity: Each disjunction *has* exactly one disjunct 1.~~

~~Necessity: Each disjunction *has* exactly one disjunct 2.~~

Reference Scheme: the logical operand 1 *of* the disjunction and the logical operand 2 *of* the disjunction

~~disjunction *has* disjunct 1~~ ~~FL~~

~~Definition: the disjunction operates on the disjunct 1~~

~~Synonymous Form: disjunction *has* logical operand 1~~

~~disjunction *has* disjunct 2~~ ~~FL~~

~~Definition: the disjunction operates on the disjunct 2~~

~~Synonymous Form: disjunction *has* logical operand 2~~

Issue 9258 Revise equivalence text/Remove text

equivalence

FL

Definition: binary logical operation that formulates that the meaning of its logical operands are either all true or all false

Concept Type: logical formulation kind

Synonym: material equivalence

~~Necessity: Each equivalence has exactly one condition 1.~~

~~Necessity: Each equivalence has exactly one condition 2.~~

Reference Scheme: the logical operand 1 *of* the equivalence and the logical operand 2 *of* the equivalence

~~equivalence *has* condition 1~~

~~FL~~

~~Definition: the equivalence operates on the condition 1~~

~~Synonymous Form: equivalence *has* logical operand 1~~

~~equivalence *has* condition 2~~

~~FL~~

~~Definition: the equivalence operates on the condition 2~~

~~Synonymous Form: equivalence *has* logical operand 2~~

Issue 9258 Revise exclusive disjunction text/Remove text

exclusive disjunction

FL

Definition: binary logical operation that formulates that the meaning of one logical operand is true and the meaning of the other logical operand is false

Concept Type: logical formulation kind

~~Necessity: Each exclusive disjunction has exactly one exclusive disjunct 1.~~

~~Necessity: Each exclusive disjunction has exactly one exclusive disjunct 2.~~

Reference Scheme: the logical operand 1 *of* the exclusive disjunction and the logical operand 2 *of* the exclusive disjunction

~~exclusive disjunction *has* exclusive disjunct 1~~

~~FL~~

~~Definition: the exclusive disjunction operates on the exclusive disjunct 1~~

~~Synonymous Form: exclusive disjunct *has* logical operand 1~~

~~exclusive disjunction *has* exclusive disjunct 2~~

~~FL~~

~~Definition: the exclusive disjunction operates on the exclusive disjunct 2~~

~~Synonymous Form: exclusive disjunct *has* logical operand 2~~

Issue 9258 Revise implication text

implication FL

Definition: binary logical operation that operates on an antecedent and a consequent and that formulates that the meaning of the consequent is true if the meaning of the antecedent is true

Concept Type: logical formulation kind

Synonym: material implication

Necessity: Each implication *has* exactly one antecedent.

Necessity: Each implication *has* exactly one consequent.

Reference Scheme: the antecedent *of* the implication and the consequent *of* the implication

Issue 9258 Move from 9.2.5 Logical Operands

antecedent FL

Definition: logical operand that is the condition considered by a logical operation such as an implication (e.g., what is meant by the *p* in "if *p* then *q*")

Concept Type: role

consequent FL

Definition: logical operand that is the implied or result operand to a logical operation such as an implication (e.g., what is meant by the *q* in "if *p* then *q*")

Concept Type: role

Issue 9258 Remove text/Revise text

implication *has* antecedent FL

Definition: the antecedent *is* the logical operand 1 *of* the implication

~~Synonymous Form: implication *has* logical operand 1~~

implication *has* consequent FL

Definition: the consequent *is* the logical operand 2 *of* the implication

~~Synonymous Form: implication *has* logical operand 2~~

logical negation FL

Definition: logical operation that has exactly one logical operand and that formulates that the meaning of the logical operand is false

Concept Type: logical formulation kind

Necessity: Each logical negation *has* exactly one logical operand.

Reference Scheme: the logical operand *of* the logical negation

~~**logical negation *has* negand**~~ ~~FL~~

~~Definition: the logical negation operates on the negand~~

~~Synonymous Form: logical negation *has* logical operand~~

nand formulation

FL

Definition: binary logical operation that formulates that the meaning of at least one of its logical operands is false

Concept Type: logical formulation kind

~~Necessity: Each nand formulation has exactly one logical operand 1.~~

~~Necessity: Each nand formulation has exactly one logical operand 2.~~

Reference Scheme: the logical operand 1 *of* the nand formulation and the logical operand 2 *of* the nand formulation

~~nand formulation *has* logical operand 1~~

~~FL~~

~~Definition: the nand formulation operates on the logical operand 1~~

~~nand formulation *has* logical operand 2~~

~~FL~~

~~Definition: the nand formulation operates on the logical operand 2~~

nor formulation

FL

Definition: binary logical operation that formulates that the meaning of each of its logical operands is false

Concept Type: logical formulation kind

~~Necessity: Each nor formulation has exactly one logical operand 1.~~

~~Necessity: Each nor formulation has exactly one logical operand 2.~~

Reference Scheme: the logical operand 1 *of* the nor formulation and the logical operand 2 *of* the nor formulation

~~nor formulation *has* logical operand 1~~

~~FL~~

~~Definition: the nor formulation operates on the logical operand 1~~

~~nor formulation *has* logical operand 2~~

~~FL~~

~~Definition: the nor formulation operates on the logical operand 2~~

whether-or-not formulation

FL

Definition: binary logical operation that has a consequent and an inconsequent and that formulates that the meaning the consequent is true regardless of the meaning the inconsequent

Concept Type: logical formulation kind

Necessity: Each whether-or-not formulation *has* exactly one consequent.

Necessity: Each whether-or-not formulation *has* exactly one inconsequent.

Reference Scheme: the consequent *of* the whether-or-not formulation and the inconsequent *of* the whether-or-not formulation

---

Issue 9258 Move from 9.2.5 Logical Operands

---

inconsequent

FL

Definition: logical operand that is an operand irrelevant to the logical result of a logical operation such as of a whether-or-not formulation

Concept Type: role

whether-or-not formulation *has* consequent FL

Definition: the consequent *is* the logical operand 1 *of* the whether-or-not formulation

~~Synonymous Form: whether-or-not formulation *has* logical operand 1~~

whether-or-not formulation *has* inconsequent FL

Definition: the inconsequent *is* the logical operand 2 *of* the whether-or-not formulation

~~Synonymous Form: whether-or-not formulation *has* logical operand 2~~

9.2.6 Quantifications

Issue 10596 Replace figure
Issue 9723 Replace figure

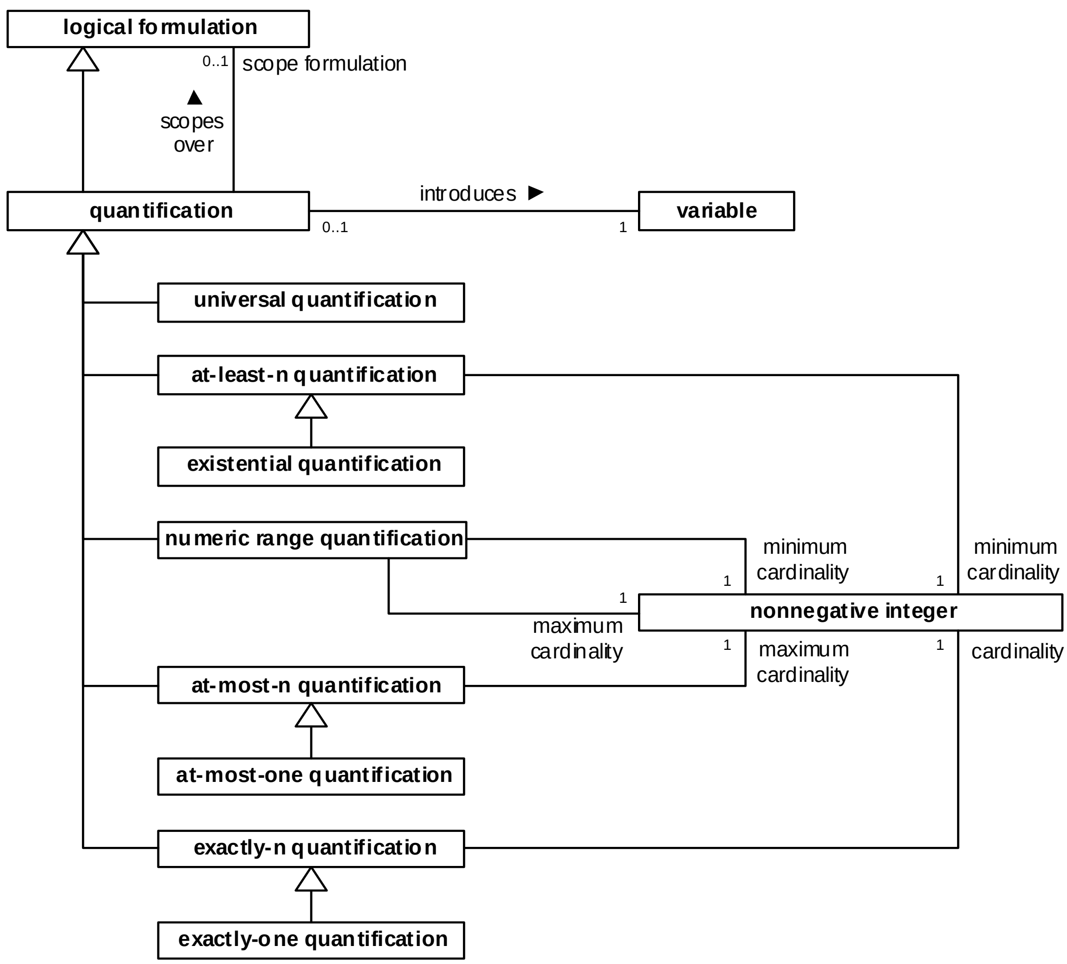


Figure 9.8

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9723 Replace text / add text
Issue 9724 Add Note
Issue 10596 Remove Necessity / Replace text

quantification

FL

Definition: logical formulation that ~~applies a logical quantification operation to~~ introduces a variable and that has either the meaning: all referents of the variable satisfy a scope formulation; or the meaning: a bounded number of referents of the variable exist and satisfy a scope formulation, if there is one

Note: A referent of the introduced variable satisfies a scope formulation if the meaning formulated by the scope formulation is true with every occurrence of the variable interpreted as referring to the referent.

Note: If a quantification scopes over no logical formulation, the meaning is that the bounded number of referents exist.

Note: Quantifications other than universal quantification and existential quantification involve cardinalities in a way that requires distinguishability of the things a variable refers to - a means to determine when one thing is not the same thing as another thing. For example, the quantification meant by “at least 2” in “EU-Rent owns at least 2 cars” means that there exists a first car and a second car and the first car is not the second car - the two cars are distinct. Physical things tend to be distinguished intuitively by having different physical locations at any point in time, but abstract things are indistinguishable without distinguishing properties. Reference schemes provide distinguishability and are often particularly important for abstract things.

Necessity: Each quantification *introduces* exactly one variable.

Necessity: Each variable *is introduced by* at most one quantification.

Necessity: Each quantification *scopes over* at most one logical formulation.

~~Necessity:~~ ~~Each quantification *is restricted by* at most one logical formulation.~~

Necessity: A variable that *is free within* a logical formulation that *is scoped over by* a quantification *is free within* the quantification if and only if the quantification does not *introduce* the variable.

~~Necessity:~~ ~~A variable that *is free within* a logical formulation that *restricts* a quantification *is free within* the quantification if and only if the quantification does not *introduce* the variable.~~

Necessity: A variable that *is free within* a logical formulation that *restricts* a variable that *is introduced by* a quantification *is free within* the quantification if and only if the quantification does not *introduce* the variable.

Example: “Each car model is supplied by a car manufacturer.”
The proposition is meant by a universal quantification.
. The universal quantification introduces a first variable.
. . The first variable ranges over the concept ‘car model’.
. The universal quantification scopes over an existential quantification.
. . The existential quantification introduces a second variable.
. . . The second variable ranges over the concept ‘car manufacturer’.
. . The existential quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type
‘car manufacturer supplies car model’.
. . . . The ‘car manufacturer’ role is bound to the second variable.
. . . . The ‘car model’ role is bound to the first variable.

Issue 10423 Remove text
Issue 9723 Replace text / add text

---

quantification *introduces* variable

FL

Definition: the quantification ~~is over referents of~~ binds the variable such that it is not free within the quantification

Note: For each referent of the variable the scope formulation, if there is one, is considered with every occurrence of the variable interpreted as referring to the referent.

~~Synonymous Form:~~ ~~variable *is introduced by* quantification~~

Issue 10423 Remove text
Issue 10596 Remove entire entry

~~logical formulation *restricts* quantification~~

~~FL~~

~~Definition:~~ ~~results of the quantification consider only the referents of the variable introduced by the quantification for which the logical formulation is satisfied~~

~~Synonymous Form:~~ ~~quantification *is restricted by* logical formulation~~

~~Note:~~ ~~A logical formulation restricts a quantification in the same way that a concept that is ranged over by a variable restricts the quantification that introduces the variable. In the formulation of a statement, a restricting logical formulation is used to capture the sense of a restrictive clause in a way that does not confuse it with the primary sense of the statement.~~

~~Example:~~ ~~A possible formulation of the rule, "Each scheduled drop-off location of each rental must be an EU-Rent branch," includes a universal quantification introducing a variable for 'rental' that scopes over another universal quantification introducing a variable for 'scheduled drop-off location.' The second quantification is restricted by a logical formulation so that for each rental the result considers only scheduled drop-off locations of that rental.~~

Issue 9723 Replace text / add text / add new entry

quantification *scopes over* logical formulation

FL

Definition: ~~the overall scope of the quantification is the logical formulation~~ each referent of the variable introduced by the quantification satisfies the logical formulation if the meaning formulated by the scope formulation is true with every occurrence of the variable interpreted as referring to the referent

Synonymous Form: ~~logical formulation *is scoped over by* quantification~~
quantification *has* scope formulation

Note: A quantification other than a universal quantification does not necessarily scope over a logical formulation (e.g., formulation of "some customer exists" can simply be an existential quantification introducing a variable that ranges over the concept 'customer').

Note: If a quantification scopes over a logical formulation, the variable introduced by the quantification is a free variable of that logical formulation, except in the rare case of a vacuous quantification.

scope formulation

FL

Definition: logical formulation that a given quantification *scopes over*

Concept Type: role

Issue 9462 Remove text/Revise text
Issue 9723 Replace text
Issue 10596 Remove text

universal quantification

FL

Definition: ~~quantification that applies the universal quantification operation (∀) scoping over a logical formulation~~
quantification that scopes over a logical formulation and that has the meaning: for each referent of the variable introduced by the quantification the meaning formulated by the logical formulation for the referent is true

Concept Type: logical formulation kind

Necessity: Each universal quantification *scopes over* a logical formulation.

Reference Scheme: the logical formulation that *is scoped over by* the universal quantification and the variable that *is introduced by* the universal quantification ~~and the set of each logical formulations that *restrict* the universal quantification~~

Issue 9462 Remove text/Revise text
Issue 9940 Remove "Concept Type"
Issue 9723 Replace Definition
Issue 9724 Add Note / Remove text
Issue 10596 Remove text

existential quantification

FL

Definition: at-least-n quantification that *has* the minimum cardinality 1~~applies the existential quantification operation (∃), 'n' being 1~~

Note: An existential quantification, unlike other at-least-n quantifications, does not require distinguishability of referents.

~~Concept Type:~~ ~~logical formulation kind~~

Reference Scheme: the set of ~~each~~ logical formulations that *are scoped over by* the existential quantification and the variable that *is introduced by* the existential quantification ~~and the set of each logical formulations that *restrict* the existential quantification~~

Issue 9948 Revise text

maximum cardinality

FL

Definition: nonnegative integer that is an upper bound in a quantification (such as an at-most-n quantification)

Concept Type: role

minimum cardinality

FL

Definition: nonnegative integer that is a lower bound in a quantification (such as an at-least-n quantification)

Concept Type: role

Issue 9462 Remove text/Revise text
Issue 9723 Replace Definition
Issue 9724 Add Note
Issue 10596 Remove text

at-least-n quantification

FL

Definition: quantification that has a minimum cardinality and that has the meaning: the number of referents of the variable introduced by the quantification that exist and that satisfy a scope formulation, if there is one, is not less than the minimum cardinality, and if the minimum cardinality is greater than one, the referents are distinct

~~applies the 'at least *n*' quantification operation ($\exists^{\geq n}$), '*n*' representing a minimum cardinality~~

Concept Type: logical formulation kind

Note: For a minimum cardinality of 1, distinctness of referents is irrelevant.

Necessity: Each at-least-n quantification *has* exactly one minimum cardinality.

Necessity: The minimum cardinality *of* each at-least-n quantification *is a* positive integer.

Reference Scheme: the minimum cardinality *of* the at-least-n quantification and the set of ~~each~~ logical formulations that *are scoped over by* the at-least-n quantification and the variable that *is introduced by* the at-least-n quantification ~~and the set of each logical formulations that *restrict* the at-least-n quantification~~

at-least-n quantification *has* minimum cardinality

FL

Definition: the at-least-n quantification is satisfied by the minimum cardinality or greater

Issue 9462 Remove text/Revise text
Issue 9723 Revise text / add text
Issue 10596 Remove text

at-most-n quantification

FL

Definition: quantification that has a maximum cardinality and that has the meaning: the number of distinct referents of the variable introduced by the quantification that exist and that satisfy a scope formulation, if there is one, is not greater than the maximum cardinality

~~applies the 'AT MOST *n*' quantification operation ($\exists^{\leq n}$), '*n*' representing a maximum cardinality~~

Concept Type: logical formulation kind

Necessity: Each at-most-n quantification *has* exactly one maximum cardinality.

Necessity: The maximum cardinality *of* each at-most-n quantification *is a* positive integer.

Reference Scheme: the maximum cardinality *of* the at-most-n quantification and the set of ~~each~~ logical formulations that *are scoped over by* the at-most-n quantification and the variable that *is introduced by* the at-most-n quantification ~~and the set of each logical formulations that *restrict* the at-most-n quantification~~

Example: "Each rental must have at most three additional drivers." See the introduction to Clause 9 for a semantic formulation of this rule.

at-most-n quantification *has* maximum cardinality

FL

Definition: the at-most-n quantification is satisfied by the maximum cardinality or less

Issue 9462 Remove text/Revise text
Issue 9940 Remove "Concept Type"
Issue 9723 Replace text / remove text
Issue 9724 Add Note
Issue 10596 Remove text

at-most-one quantification

FL

Definition: at-most-n quantification that *has* the maximum cardinality 1

~~applies the 'AT MOST ONE' quantification operation ($\exists^{0..1}$), 'n' being 1~~

Note: A number of referents is at most one if and only if every referent is the same referent.

~~Concept Type:~~ ~~logical formulation kind~~

Reference Scheme: the set of ~~each~~ logical formulations that *are scoped over by* the at-most-one quantification and the variable that *is introduced by* the at-most-one quantification ~~and the set of each logical formulations that *restrict* the at-most-one quantification~~

~~Note:~~ ~~An at-most-one quantification is logically equivalent to an at-most-n quantification that has a maximum cardinality of 1.~~

Issue 9462 Remove text/Revise text
Issue 9940 Remove "Concept Type"
Issue 9723 Revise Definition
Issue 10596 Remove text

exactly-n quantification

FL

Definition: quantification that has a cardinality and that has the meaning: the number of referents of the variable introduced by the quantification that exist and that satisfy a scope formulation, if there is one, equals the cardinality

~~applies the 'EXACTLY n' quantification operation ($\exists^{n}$), 'n' representing a cardinality~~

~~Concept Type:~~ ~~logical formulation kind~~

Necessity: Each exactly-n quantification *has* exactly one cardinality.

Necessity: The cardinality *of* each exactly-n quantification *is a* positive integer.

Reference Scheme: the cardinality *of* the exactly-n quantification and the set of ~~each~~ logical formulations that *are scoped over by* the exactly-n quantification and the variable that *is introduced by* the exactly-n quantification ~~and the set of each logical formulations that *restrict* the exactly-n quantification~~

Note: An exactly-n quantification is logically equivalent to a conjunction of an at-least-n quantification and an at-most-n quantification using the cardinality as minimum cardinality and maximum cardinality respectively.

exactly-n quantification *has* cardinality

FL

Definition: the exactly-n quantification is satisfied only by the cardinality

Issue 9462 Remove text/Revise text
Issue 9723 Replace text / Remove text
Issue 9724 Add Note
Issue 10596 Remove text

exactly-one quantification

FL

Definition: exactly-n quantification that *has* the cardinality 1

Note: A number of referents is exactly one if and only if there is a referent and every referent is that same referent.

Concept Type: logical formulation kind

Reference Scheme: the set of ~~each~~ logical formulations that *are scoped over by* the exactly-one quantification and the variable that *is introduced by* the exactly-one quantification ~~and the set of each logical formulations that *restrict* the exactly-one quantification~~

~~Note:~~ ~~An exactly-one quantification is logically equivalent to an exactly-n quantification that has a cardinality of 1.~~

Issue 9462 Remove text/Revise text
Issue 9723 Revise Definition
Issue 10596 Remove text

numeric range quantification

FL

Definition: quantification that has a minimum cardinality and a maximum cardinality greater than the minimum cardinality and that has the meaning: the number of referents of the variable introduced by the quantification that exist and that satisfy a scope formulation, if there is one, is not less than the minimum cardinality and is not greater than the maximum cardinality

~~applies the 'NUMERIC RANGE' quantification operation ($\exists^{n..m}$), '*n*' representing a minimum cardinality and '*m*' representing a maximum cardinality~~

Concept Type: logical formulation kind

Necessity: Each numeric range quantification *has* exactly one maximum cardinality.

Necessity: Each numeric range quantification *has* exactly one minimum cardinality.

Necessity: The minimum cardinality *of* each numeric range quantification *is less than* the maximum cardinality *of* the numeric range quantification.

Reference Scheme: the minimum cardinality *of* the numeric range quantification and the maximum cardinality *of* the numeric range quantification and the set of ~~each~~ logical formulations that *are scoped over by* the numeric range quantification and the variable that *is introduced by* the numeric range quantification ~~and the set of each logical formulations that *restrict* the numeric range quantification~~

Note: A numeric range quantification is logically equivalent to a conjunction of an at-least-n quantification and an at-most-n quantification using the minimum cardinality and maximum cardinality respectively.

numeric range quantification ***has*** **maximum cardinality** FL

Definition: the numeric range quantification cannot be satisfied by a number greater than the maximum cardinality

numeric range quantification ***has*** **minimum cardinality** FL

Definition: the numeric range quantification cannot be satisfied by a number less than the minimum cardinality

9.2.7 Objectifications

Issue 10569 Replace figure

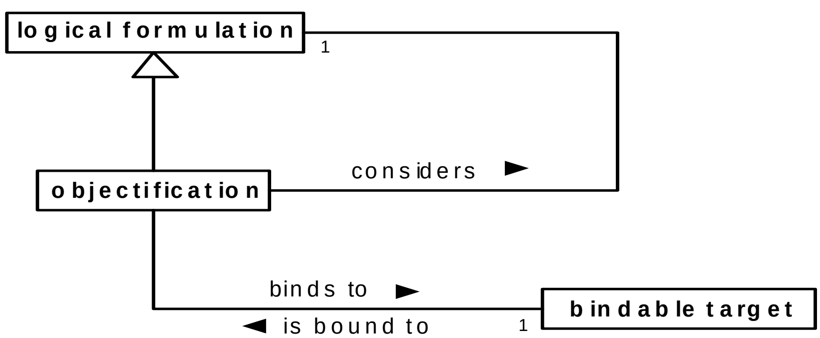


Figure 9.9

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9725 Replace "Definition"/ Add text
Issue 10596 Replace Example text
Issue 9955 REplace text

---

objectification FL

Definition: logical formulation that involves a bindable target and a considered logical formulation and that formulates the meaning:the thing to which the bindable target refers is a state of affairs that corresponds to the meaning of the considered logical formulation

Concept Type: logical formulation kind

Note: An objectification is similar to an instantiation formulation in that it is satisfied by a correspondence of a referent thing to a meaning. For an instantiation formulation the meaning is a concept. For an objectification the meaning is a proposition.

Necessity: Each objectification *considers* exactly one logical formulation.

Necessity: Each objectification *binds to* exactly one bindable target.

Necessity: Each variable that *is bound to* an objectification *is free within* the objectification.

Necessity: Each variable that *is free within* the logical formulation that *is considered by* an objectification *is free within* the objectification.

Reference Scheme: the bindable target that *is bound to* the objectification and the logical formulation *that is considered by* the objectification

Example: 'late return' defined as "actuality that a given rental is returned late".
The concept 'late return' is defined by a closed projection.
. The projection is on a first variable.
. . The first variable ranges over the concept 'actuality'.
. The projection has an auxiliary variable.
. . The auxiliary variable ranges over the concept 'rental'.
. The projection is constrained by an objectification.
. . The objectification binds to the first variable.
. . The objectification considers an atomic formulation.
. . . The atomic formulation is based on the characteristic 'rental is returned late'.
. . . . The 'rental' role is bound to the auxiliary variable.

Example: "EU-Rent reviews each corporate account at EU-Rent Headquarters."
The statement above could be formulated using a ternary fact type 'company reviews account at place,' but such a fact type is not likely ~~found~~ represented in a business vocabulary because it mixes two orthogonal binary fact types: 'company reviews account' and 'state of affairs occurs at place'. The formulation below uses the two binary fact types and employs objectification to tie them together.
The statement is formulated by a universal quantification.
. The quantification introduces a first variable.
. . The first variable ranges over the concept 'corporate account'.
. The quantification scopes over an existential quantification.
. . The existential quantification introduces a second variable.
. . . The second variable ranges over the fact type 'company reviews account'.
~~. . The existential quantification is restricted by an objectification.~~
. . . The second variable is restricted by an objectification.
. . . . The objectification binds to the second variable.
. . . . The objectification considers an atomic formulation.
. . . . . The atomic formulation is based on the fact type 'company reviews account'.
. . . . . . The 'company' role is bound to the individual concept 'EU-Rent'.
. . . . . . The 'account' role is bound to the first variable.
. . The existential quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type 'state of affairs occurs at place'.
. . . . The 'state of affairs' role is bound to the second variable.
. . . . The 'place' role is bound to the individual concept 'EU-Rent Headquarters'.

Example: "EU-Rent has reviewed each corporate account."
The fact type 'company reviews account' can be used to formulate the meaning of 'company has reviewed account' (the present perfect tense) by using objectification along with a generic fact type for the present perfect tense, 'state of affairs has occurred'. A formulation of the example statement is similar to that of the previous example but uses the fact type 'state of affairs has occurred' rather than 'state of affairs occurs at place'.

Example: "EU-Rent privately reviews each corporate account."
This example is similar to the previous two. A fact type, such as "state of affairs is private," is sometimes expressed using an adverb, such as "privately." A formulation of the example statement is similar to that of the previous two examples, but uses the fact type 'state of affairs is private'.

Example: “If a rental car is returned late because the car has a mechanical breakdown ….” In a possible formulation of this example, objectifications of “the car has a mechanical breakdown” and “the rental car is returned late” respectively formulate something for each role of the fact type ‘state of affairs causes state of affairs’.

Issue 10423 Remove text

objectification *considers* logical formulation FL

Definition: the objectification is of the state or event that corresponds to the meaning of the logical formulation

~~Synonymous Form:~~ ~~logical formulation *is considered by* objectification~~

objectification *binds to* bindable target FL

Definition: the bindable target indicates the referent state or event identified by the objectification

Synonymous Form: bindable target *is bound to* objectification

9.2.8 Projecting Formulations

Issue 9731 Revise figure
Issue 9712 Replace figure

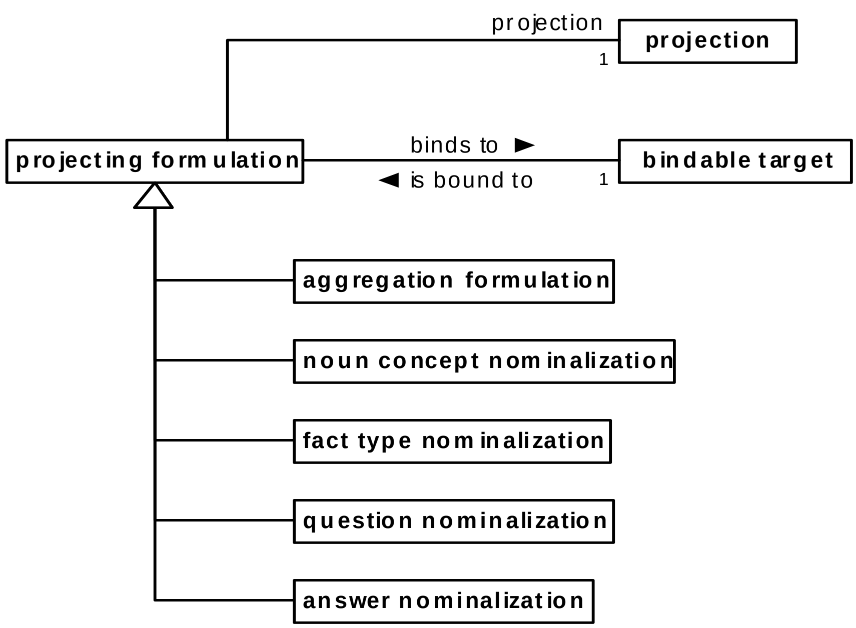


Figure 9.10

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9726 Add text

projecting formulation FL

Definition: logical formulation of a referent thing considered with respect to a particular projection

Necessity: Each projecting formulation *has* exactly one projection.

Necessity: Each projecting formulation *binds to* exactly one bindable target.

Necessity: Each variable that *is bound to* a projecting formulation *is free within* the projecting formulation.

Necessity: Each variable that *is free within* the projection *of* a projecting formulation *is free within* the projecting formulation.

Note: The concept 'projecting formulation' is abstract. See its specializations for semantics.

Example: See 'aggregation formulation', 'question nominalization', and 'answer nominalization'.

projecting formulation *has* projection FL

Definition: the projecting formulation is based on the projection

projecting formulation *binds to* bindable target FL

Definition: the bindable target indicates the referent thing considered by the projecting formulation

Synonymous Form: bindable target *is bound to* projecting formulation

Issue 9727 Revise text
Issue 10596 Replace Example
Issue 9712 Revise Definition, add Note

aggregation formulation

FL

Definition: projecting formulation that formulates the meaning: the thing to which the bindable target bound to the projecting formulation refers is the result of ~~set or multiset resulting from~~ the projection of the projecting formulation

Note: The aggregation formulation is used primarily to associate a variable with a set of things, involvements, or actualities that satisfy some condition. That is, it formulates natural language expressions of the form: "let <*variable*> be the set of all things *t* such that <some condition involving *t*>," so that <*variable*> can then be used in other formulations regarding the set. The <condition involving *t*> often includes some free variable introduced in the context in which the formulation is used.

Concept Type: logical formulation kind

Necessity: The projection *of* each aggregation formulation *is on* exactly one variable.

~~Note:~~ ~~An aggregation formulation is satisfied for each referent set or multiset that exactly matches the result of the projection, taking the result as the set or multiset of referents of the variable in the projection.~~

Reference Scheme: the bindable target that *is bound to* the aggregation formulation and the projection *of* the aggregation formulation

~~Example:~~ ~~"The number of rental cars stored at a given branch must not exceed the car storage capacity of the branch." This example considers the number of elements in a set (the set of rental cars stored at a branch). The projection of an aggregation formulation is used to define that set, and the aggregation formulation restricts the third variable below so that its referent is that set.~~

~~The statement is formulated by an obligation formulation.~~

~~. The obligation formulation embeds a first universal quantification.~~

~~. . The first universal quantification introduces a first variable.~~

~~. . . The first variable ranges over the concept 'branch.'~~

~~. . The first universal quantification scopes over a second universal quantification.~~

~~. . . The second universal quantification introduces a second variable.~~

~~. . . . The second variable ranges over the concept 'number.'~~

~~. . . . The second variable is unitary.~~

~~. . . The second universal quantification is restricted by a third universal quantification.~~

~~. . . . The third universal quantification introduces a third variable.~~

~~. . . . . The third variable is unitary.~~

~~. . . . The third universal quantification is restricted by an aggregation formulation.~~

~~. . . . . The aggregation formulation binds to the third variable.~~

~~. . . . . The aggregation formulation considers a projection.~~

~~. . . . . . The projection is on a fourth variable.~~

~~. . . . . . . The fourth variable ranges over the concept 'rental car.'~~

~~. . . . . . The projection is constrained by an atomic formulation.~~

~~. . . . . . . The atomic formulation is based on the fact type 'rental car is stored at branch.'~~

~~. . . . . The 'rental car' role is bound to the fourth variable.~~

~~. . . . . The 'branch' role is bound to the first variable.~~

~~. . . . The third universal quantification scopes over an atomic formulation.~~

~~. . . . . The atomic formulation is based on the fact type ‘set has number.’~~
~~. . . . . . The 'set' role is bound to the third variable.~~
~~. . . . . . The 'number' role is bound to the second variable.~~
~~. . . The second universal quantification scopes a fourth universal quantification.~~
~~. . . . The fourth universal quantification introduces a fifth variable.~~
~~. . . . . The fifth variable ranges over the concept ‘car storage capacity.’~~
~~. . . . . The fifth variable is unitary.~~
~~. . . . The fourth universal quantification is restricted by an atomic formulation.~~
~~. . . . . The atomic formulation is based on the fact type ‘branch has car storage capacity.’~~
~~. . . . . . The ‘branch’ role is bound to the first variable.~~
~~. . . . . . The ‘car storage capacity’ role is bound to the fifth variable.~~
~~. . . . The fourth universal quantification scopes over a logical negation.~~
~~. . . . . The logical operand of the logical negation is an atomic formulation.~~
~~. . . . . . The atomic formulation is based on the fact type ‘number1 exceeds number2.’~~
~~. . . . . . . The ‘number1’ role is bound to the second variable.~~
~~. . . . . . . The ‘number2’ role is bound to the fifth variable.~~

Example: “The number of rental cars stored at a given branch must not exceed the car storage capacity of the branch.” This example considers the number of elements in a set (the set of rental cars stored at a branch). The projection of an aggregation formulation is used to define that set, and the aggregation formulation restricts the third variable below so that its referent is that set.

The statement is formulated by an obligation formulation.
. The obligation formulation embeds a first universal quantification.
. . The first universal quantification introduces a first variable.
. . . The first variable ranges over the concept ‘branch’.
. . The first universal quantification scopes over a second universal quantification.
. . . The second universal quantification introduces a second variable.
. . . . The second variable ranges over the concept ‘number’.
. . . . The second variable is unitary.
. . . . The second variable is restricted by a third universal quantification.
. . . . . The third universal quantification introduces a third variable.
. . . . . . The third variable is unitary.
. . . . . . The third variable is restricted by an aggregation formulation.
. . . . The aggregation formulation binds to the third variable.
. . . . The aggregation formulation considers a projection.
. . . . The projection is on a fourth variable.
. . . . The fourth variable ranges over the concept ‘rental car’.
. . . . The projection is constrained by an atomic formulation.
. . . . The atomic formulation is based on the fact type
‘rental car is stored at branch’.
. . . . The ‘rental car’ role is bound to the fourth variable.
. . . . The ‘branch’ role is bound to the first variable.
. . . . . The third universal quantification scopes over an atomic formulation.
. . . . . . The atomic formulation is based on the fact type ‘set has number’.
. . . . The ‘set’ role is bound to the third variable.
. . . . The ‘number’ role is bound to the second variable.

. . . The second universal quantification scopes a fourth universal quantification.
. . . . The fourth universal quantification introduces a fifth variable.
. . . . . The fifth variable ranges over the concept ‘car storage capacity’.
. . . . . The fifth variable is unitary.
. . . . . The fifth variable is restricted by an atomic formulation.
. . . . . . The atomic formulation is based on the fact type
‘branch has car storage capacity’.
. . . . The ‘branch’ role is bound to the first variable.
. . . . The ‘car storage capacity’ role is bound to the fifth variable.
. . . . The fourth universal quantification scopes over a logical negation.
. . . . . The logical operand of the logical negation is an atomic formulation.
. . . . . . The atomic formulation is based on the fact type ‘number_1 exceeds number_2’.
. . . . The ‘number_1’ role is bound to the second variable.
. . . . The ‘number_2’ role is bound to the fifth variable.

Issue 9728 Replace text
Issue 10596 Replace text

---

noun concept nominalization ~~fact type formulation~~ FL

Definition: projecting formulation that formulates the meaning: the thing to which the bindable target bound to the projecting formulation refers is a noun concept that is defined by the projection of the projecting formulation

Concept Type: logical formulation kind

Necessity: The projection *of* each noun concept nominalization ~~fact type formulation~~ *is on* exactly one variable.

Note: In the case of variables being free within a projection of a noun concept nominalization ~~fact type formulation~~, the projection is considered to define a noun concept only in the context of there being a referent thing given for each free variable.

Note: Nouns are generally used to refer to things in the extension of the noun concept meant by the noun. Less commonly, a noun is used to mention a noun concept itself. This is referred to as a “mention” of the concept as opposed to a “use.”

Reference Scheme: the bindable target that *is bound to* the noun concept nominalization ~~fact type formulation~~ and the projection *of* the noun concept nominalization ~~fact type formulation~~

Example: EU-Rent stores at least 300 kiloliters of petrol.”
In this example, ‘petrol’ is a mention of the concept ‘petrol’ which is used in the ‘type’ role of a fact type ‘quantity is of type’.
The statement is formulated by an at-least-n quantification.
. The minimum cardinality of the quantification is 300.
. The quantification introduces a first variable.
. . The first variable ranges over the concept ‘kiloliter’.
. The quantification scopes over an existential quantification.
. . The existential quantification introduces a second variable.
~~. . The existential quantification is restricted by a noun concept formulation.~~

. . . The second variable is restricted by a noun concept nominalization ~~fact type formulation~~.
. . . . The noun concept nominalization ~~fact type formulation~~ binds to the second variable.
. . . . The noun concept nominalization ~~fact type formulation~~ considers a projection.
. . . . . The projection is on a third variable.
. . . . . . The third variable ranges over the concept 'petrol'.
. . The existential quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type 'company stores thing'.
. . . . The 'company' role is bound to the individual concept 'EU-Rent'.
. . . . The 'thing' role is bound to the first variable.

Example: "EU-Rent stores at least 300 kiloliters of medium or high grade petrol."
This example is the same as the previous example except that the mentioned concept is more complex: "medium or high grade petrol." The statement's formulation starts with that of the previous example and adds the following regarding its projection:
. . . . . The projection is constrained by a disjunction.
. . . . . . The disjunction's logical operand 1 is an atomic formulation.
. . . . The atomic formulation is based on the characteristic 'petrol is medium grade'.
. . . . The 'petrol' role is bound to the third variable.
. . . . . . The disjunction's logical operand 2 is an atomic formulation.
. . . . The atomic formulation is based on the characteristic 'petrol is high grade'.
. . . . The 'petrol' role is bound to the third variable.

Example: "EU-Rent's headcount increased by 300 in the year 2005."
The proposition is based on the fact type 'quantitative property increased by quantity in time period'. The quantitative property is the noun concept expressed as "EU-Rent's headcount." The statement is formulated by an existential quantification.
. The quantification introduces a unitary variable.
. . The variable ranges over the concept 'quantitative property'.
~~. The quantification is restricted by a noun concept formulation.~~
. . The variable is restricted by a noun concept nominalization ~~fact type formulation~~.
. . . The noun concept nominalization ~~fact type formulation~~ binds to the first variable.
. . . The noun concept nominalization ~~fact type formulation~~ considers a projection.
. . . . The projection is on a second unitary variable.
. . . . . The second variable ranges over the concept 'headcount'.
. . . . The projection is constrained by an atomic formulation.
. . . . . The atomic formulation is based on the fact type 'company has headcount'.
. . . . . . The 'company' role is bound to the individual concept 'EU-Rent'.
. . . . . . The 'headcount' role is bound to the second variable.
. The quantification scopes over an atomic formulation.
. . The atomic formulation is based on the fact type 'quantitative property increased by quantity in time period'.
. . . The 'quantitative property' role is bound to the first variable.
. . . The 'quantity' role is bound to the individual concept '300'.
. . . The 'time period' role is bound to the individual concept 'year 2005'.

Issue 9712 Add text
Issue 9948 Remove Note text

fact type nominalization

FL

Definition: projecting formulation that formulates the meaning: the thing to which the bindable target bound to the projecting formulation refers is a fact type that is defined by the projection of the projecting formulation

Concept Type: logical formulation kind

Reference Scheme: the bindable target that *is bound to* the fact type nominalization and the projection *of* the fact type nominalization

Note: A fact type nominalization formulates the (anonymous) fact type defined by a projection. In most uses of fact type nominalizations, the bindable target is a unitary variable, and the effect is to define the variable to refer to the anonymous fact type defined by the projection. It is the only referent for which the fact type nominalization will hold.

Note: In the case of variables being free within a projection of a fact type nominalization, the projection is considered to define a fact type only in the context of there being a referent thing substituted for each free variable.

Note: More information about how a projection defines a fact type is in the entry for 'closed projection *defines* fact type'. A fact type nominalization nominalizes only a fact type, not its roles. ~~which can be nominalized using a noun concept nominalization~~.

Example: "Being established by a rental booking is a characteristic attributed to each advance rental."
The characteristic expressed as "being established by a rental booking" is nominalized within the statement.
The statement is formulated by a universal quantification.
. The universal quantification introduces a first variable.
. . The first variable ranges over the concept 'advance rental'.
. The universal quantification scopes over a first existential quantification.
. . The first existential quantification introduces a second variable.
. . . The second variable ranges over the concept 'characteristic'.
. . . The second variable is restricted by an atomic formulation.
. . . . The atomic formulation is based on the fact type 'characteristic is attributed to thing'.
. . . . . The 'characteristic' role is bound to the second variable.
. . . . . The 'thing' role is bound to the first variable.
. . The first existential quantification scopes over a fact type nominalization.
. . . The fact type nominalization binds to the second variable.
. . . The fact type nominalization considers a projection.
. . . . The projection is on a third variable.
. . . . The projection is constrained by a second existential quantification.
. . . . . The second existential quantification introduces a fourth variable.
. . . . . . The fourth variable ranges over the concept 'rental booking'.
. . . . . The second existential quantification scopes over an atomic formulation.
. . . . . . The atomic formulation is based on the fact type
'rental booking establishes advanced rental'.
. . . . The 'rental booking' role is bound to the fourth variable.
. . . . The 'advanced rental' role is bound to the third variable.

Issue 9731 Remove text

~~**fact type formulation**~~ ~~FL~~

~~Definition:~~	~~projecting formulation of a referent fact type whose intension is formulated in a particular projection~~
~~Concept Type:~~	~~logical formulation kind~~
~~Necessity:~~	~~The projection *of* each fact type formulation *is* a set projection.~~
~~Note:~~	~~A fact type formulation is satisfied for each referent that is a fact type defined by the projection. For a closed projection, the projection *defines* the fact type. Otherwise, a satisfying fact type is the meaning formulated by the projection plus an understood reference to the referent of each variable that is free within the projection.~~
~~Reference Scheme:~~	~~the bindable target that *is bound to* the fact type formulation and the projection *of* the fact type formulation~~
~~Example:~~	~~In the statement, "drinking and driving violates a rental agreement," the subject of "violates" is a characteristic whose meaning is formulated as a projection over a conjunction of atomic formulations based on the characteristics 'person drinks' and 'person drives.'~~

9.2.9 Nominalizations of Propositions and Questions

Issue 10569 Replace figure

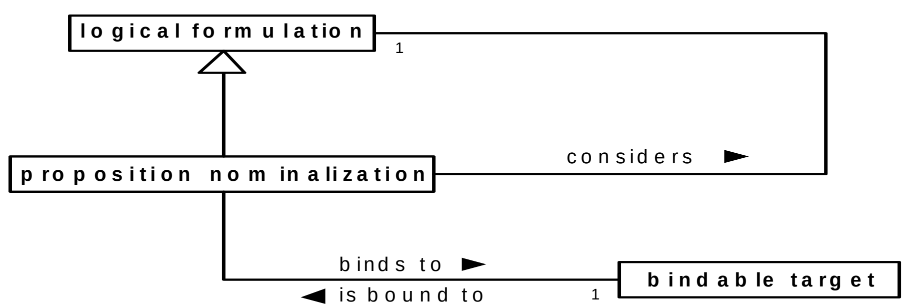


Figure 9.11

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9732 Replace Definition, Note, and Examples
Issue 10596 Replace text and add "." to some lines

proposition nominalization FL

Definition:	logical formulation that involves a bindable target and a considered logical formulation and that formulates the meaning: the thing to which the bindable target refers is the proposition that is formulated by the considered logical formulation
Concept Type:	logical formulation kind
Necessity:	Each proposition nominalization *considers* exactly one logical formulation.

Necessity:	Each proposition nominalization *binds to* exactly one bindable target.
Necessity:	Each variable that *is bound to* a proposition nominalization *is free within* the proposition nominalization.
Necessity:	Each variable that *is free within* the logical formulation that *is considered by* a proposition nominalization *is free within* the proposition nominalization.
Note:	A closed logical formulation means exactly one proposition. An open logical formulation does not mean any proposition. In the case of variables being free within a considered logical formulation, the formulation is considered to mean a proposition only in the context of there being a referent thing given for each free variable.
Note:	The truth of a nominalized proposition is not relevant to the satisfaction of the proposition nominalization.
Reference Scheme:	the bindable target that *is bound to* the proposition nominalization and the logical formulation *that is considered by* the proposition nominalization

Example:

"Each EU-Rent branch posts a sign stating that no personal checks are accepted by the branch."

The statement is formalized by a universal quantification.
. The universal quantification is on a first variable.
. . The variable ranges over the concept 'EU-Rent branch'.
. The universal quantification scopes over an existential quantification.
. . The existential quantification introduces a second variable.
. . . The second variable ranges over the concept 'sign'.
~~. . The existential quantification is restricted by a second existential quantification.~~
. . . The second variable is restricted by a second existential quantification.
. . . . The second existential quantification introduces a third variable.
. . . . . The third variable ranges over the concept 'proposition'.
~~. . . The second existential quantification is restricted by a proposition nominalization~~.
. . . . . The third variable is restricted by a proposition nominalization.
. . . . . . The proposition nominalization binds to the third variable
. . . . . . The proposition nominalization considers a logical negation.
. . . . The logical operand of the negation is a third existential quantification.
. . . . The quantification introduces a fourth variable.
. . . . The variable ranges over the concept 'personal check'.
. . . . The quantification scopes over an atomic formulation.
. . . . The atomic formulation is based on the fact type
'branch accepts monetary instrument'.
. . . . The 'branch' role is bound to the first variable.
. . . . The 'monetary instrument' role is bound to the fourth variable.
. . . . The second existential quantification scopes over an atomic formulation.
. . . . . The atomic formulation is based on the fact type 'sign states proposition'.
. . . . . . The 'sign' role is bound to the second variable.
. . . . . . The 'proposition' role is bound to the third variable.
. . The first existential quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type 'branch posts sign'.
. . . . The 'branch' role is bound to the first variable.
. . . . The 'sign' role is bound to the second variable.

Issue 10423 Remove text

proposition nominalization *considers* logical formulation

FL

Definition:	the proposition nominalization nominalizes the proposition whose meaning is formulated by the logical formulation
~~Synonymous Form:~~	~~logical formulation *is considered by* proposition nominalization~~

proposition nominalization *binds to* bindable target

FL

Definition:	the bindable target indicates the referent proposition identified by the proposition nominalization
Synonymous Form:	bindable target *is bound to* proposition nominalization

---

Issue 9734 Replace Definition, Notes, and Example
Issue 10569 Remove Necessity
Issue 10596 Replace text and add "." to some entries

---

question nominalization

Definition:	projecting formulation that formulates the meaning: the thing to which the bindable target bound to the projecting formulation refers is the question that is meant by the projection of the projecting formulation
Concept Type:	logical formulation kind
~~Necessity:~~	~~The projection *of* each question nominalization *is* a set projection.~~
Note:	See 'closed projection *means* question' for an explanation and examples of how questions are formulated.
Note:	A closed projection means at most one question. In the case of variables being free within a projection, the projection is considered to mean a question only in the context of there being a referent thing given for each free variable.
Reference Scheme:	the bindable target that *is bound to* the question nominalization and the projection *of* the question nominalization

Example:

"An agent asks each customer what car model the customer prefers."
The statement is formulated by a universal quantification.
. The quantification introduces a first variable.
. . The first variable ranges over the concept 'customer'.
. The quantification scopes over an existential quantification.
. . The existential quantification introduces a second variable.
. . . The second variable ranges over the concept 'agent'.
. . The existential quantification scopes over a second existential quantification.
. . . The second existential quantification introduces a third variable.
. . . . The third variable ranges over the concept 'question'.
~~. . . The second existential quantification is restricted by a question nominalization.~~
. . . . The third variable is restricted by a question nominalization.
. . . . . The question nominalization binds to the third variable.
. . . . . The question nominalization considers a projection.
. . . . . . The projection is on a fourth variable.
. . . . The variable ranges over the concept 'car model'.
. . . . . . The projection is constrained by an atomic formulation.
. . . . The atomic formulation is based on the fact type 'person prefers car model'.
. . . . The 'person' role is bound to the first variable.

. . . . The ‘car model’ role is bound to the fourth variable.
. . . The second existential quantification scopes over an atomic formulation.
. . . . The atomic formulation is based on the fact type ‘$person_1$ asks $person_2$ question’.
. . . . . The ‘$person_1$’ role is bound to the second variable.
. . . . . The ‘$person_2$’ role is bound to the first variable.
. . . . . The ‘question’ role is bound to the third variable.

Issue 9734 Replace Definition, Notes, and Example
Issue 10569 Remove Necessity
Issue 10596 Replace text and add “.” to some entries

---

answer nominalization

Definition: projecting formulation that formulates the meaning: the thing to which the bindable target bound to the projecting formulation refers is a proposition that is true and that completely and correctly answers the question meant by the projection of the projecting formulation

Concept Type: logical formulation kind

~~Necessity:~~ ~~The projection *of* each answer nominalization *is* a set projection.~~

Note: See ‘closed projection *means* question’ for an explanation and examples of how questions are formulated.

Note: In the case of variables being free within a projection, the projection is considered to mean a question only in the context of there being a referent thing given for each free variable.

Note: A thing referred to by a bindable target bound to an answer nominalization is a satisfactory proposition if it correctly and completely holds the result of the answer nominalization’s projection. A satisfying proposition incorporates the meaning formulated by the projection in the context of there being a referent thing given for each free variable of the projection. Further, the satisfying proposition refers to each referent of each variable in the projection. If the projection result has multiple elements, a satisfying proposition holds them all, conjunctively. If the projection result is empty, a satisfying projection indicates that it is empty.

Note: Each reference in a satisfying answer should use a defined reference scheme.

Reference Scheme: the bindable target that *is bound to* the answer nominalization and the projection *of* the answer nominalization

Example: “An agent tells each customer what special offer is available to the customer.”
The statement is formulated by a universal quantification.
. The quantification introduces a first variable.
. . The first variable ranges over the concept ‘customer’.
. The quantification scopes over an existential quantification.
. . The existential quantification introduces a second variable.
. . . The second variable ranges over the concept ‘agent’.
. . The existential quantification scopes over a second existential quantification.
. . . The second existential quantification introduces a third variable.
. . . . The third variable ranges over the concept ‘proposition’.
~~. . . The second existential quantification is restricted by an answer nominalization.~~
. . . . The third variable is restricted by an answer nominalization.
. . . . . The answer nominalization binds to the third variable.

. . . . . The answer nominalization considers a projection.
. . . . . . The projection is on a fourth variable.
. . . . The variable ranges over the concept ‘special offer’.
. . . . . . The projection is constrained by an atomic formulation.
. . . . The atomic formulation is based on the fact type ‘special offer is available to customer’.
. . . . The ‘special offer’ role is bound to the fourth variable.
. . . . The ‘customer’ role is bound to the first variable.
. . . The second existential quantification scopes over an atomic formulation.
. . . . The atomic formulation is based on the fact type ‘person_1 tells person_2 proposition’.
. . . . . The ‘person_1’ role is bound to the second variable.
. . . . . The ‘person_2’ role is bound to the first variable.
. . . . . The ‘proposition’ role is bound to the third variable.

If exactly two special offers (Gold Customer Discount and Free One-level Upgrade) are available to a customer having customer id ‘9876’, a satisfying answer for that customer would be the proposition meant by the statement: “The special offers available to the customer having the customer id ‘9876’ are the Gold Customer Discount and the Free One-level Upgrade.”

9.3 Projections

Issue 9947 Replace figure
Issue 9930 Replace figure
Issue 9712 Replace figure

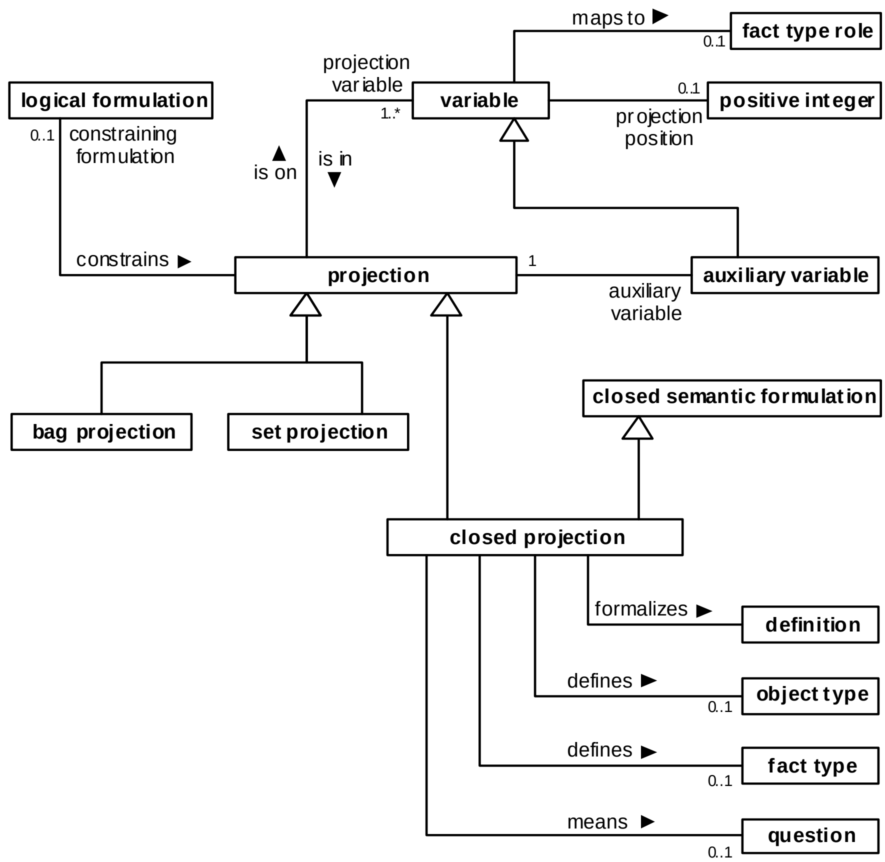


Figure 9.12

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9462 Remove text/Revise text
Issue 10596 Add text
Issue 9712 Replace text (definition and note)

projection

FL

Definition: semantic formulation that introduces one or more variables corresponding to involvements in actualities and that is possibly constrained by a logical formulation and that projects one or more of those variables

~~Definition: semantic formulation that operates over one or more variables and results in a set or multiset~~

Necessity: Each projection *is on* at least one variable.

Necessity: Each projection *is constrained by* at most one logical formulation.

Necessity: A variable that *is free within* a logical formulation that *constrains* a projection *is free within* the projection if and only if the projection *is* not *on* the variable and the variable *is* not an auxiliary variable *of* the projection.

Necessity: No projection *is* a logical formulation.

Necessity: A variable that *is in* a projection *is* not *free within* the projection.

Necessity: A variable that *is free within* a logical formulation that *restricts* another variable that *is in* a projection *is free within* the projection.

Necessity: A variable that *is free within* a logical formulation that *restricts* an auxiliary variable *of* a projection *is free within* the projection if and only if the variable *is* not the auxiliary variable.

Note: A restriction on a variable introduced by a projection cannot involve any other variable introduced by the projection.

Reference Scheme: the set of ~~each~~ variables that *are in* the projection and the set of auxiliary variables *of* the projection and the set of ~~each~~ logical formulations that *constrain* the projection

~~Note:~~ ~~A projection's result can be taken in multiple ways. Which way depends on how the projection is used. When used in an aggregating formulation or as defining a concept other than a fact type, the result elements are simply the referents of the variable in the projection. When used to define a fact type, each result element is taken as an actuality that involves the referents of the variables in the projection.~~

Note: A projection is a structure of meaning used in formulating different kinds of meanings. Each is explained separately. See the following entries: 'closed projection *defines* noun concept', 'closed projection *defines* fact type' and 'closed projection *means* question'. Also, projections are incorporated into projecting formulations, which include 'aggregation formulation', 'noun concept nominalization', 'fact type nominalization', 'question nominalization', and 'answer nominalization' each of which is explained separately with examples in previous subclauses.

Note: A projection introduces one or more variables corresponding to involvements in actualities. If the projection is constrained by a logical formulation, then for each combination of variables, one referent for each variable, the actuality is that the meaning of the constraining formulation is true. If the projection has no constraining formulation, then for each combination of variables, one referent for each variable, the actuality is that the referents exist.

That is, the basic meaning of a projection is a fact type in which all of the variables introduced by the projection correspond to roles. The basic meaning corresponds to actualities for which the following proposition holds:

t_1 is a valid referent of v_1
[AND t_2 is a valid referent of v_2
...
AND t_n is a valid referent of v_n]
[AND $S(t_1, ..., t_n)$]

where $v_1, ..., v_n$ are the variables introduced by the projection, $t_1, ..., t_n$ are things, and $S(t_1, ..., t_n)$ is the proposition formulated by the logical formulation that constrains the projection, if any, with those things substituted for the occurrences of the corresponding variables.

The meaning of a projection in some uses, however, can be restricted to refer to the involvements of the things in the roles (denoted by the projection variables) in those actualities, or to the things that have those involvements.

Note: Projections introduce variables in two ways: projection variables (variables that the projection 'is on') and auxiliary variables. Both correspond to involvements in the actualities that correspond to the basic meaning, but the result of a projection includes only the involvements that correspond to the projection variables. Auxiliary variables are used in selecting the actualities that correspond to the projection, but are not part of the intent of the projection itself.

Issue 9930 Add text
Issue 9712 Add text
Issue 11289 Replace text

projection *is on* variable

FL

Definition:	the projection introduces the variable such that satisfying referents of the variable are in the result of the projection
Synonymous Form:	variable *is in* projection
Synonymous Form:	projection *has* projection variable
~~Necessity:~~	~~No projection variable *is* a scoped-over variable.~~
Necessity:	No variable that *is in* a projection *is introduced by* a quantification.

projection *has* auxiliary variable

FL

Definition:	the auxiliary variable is introduced by the projection, but is left out of the result of the projection thereby giving the possibility of duplicates in a result
~~Necessity:~~	~~No auxiliary variable *is* a scoped-over variable.~~
~~Necessity:~~	~~No projection variable *is* an auxiliary variable.~~
Necessity:	No auxiliary variable *is introduced by* a quantification.
Necessity:	No projection *is on* an auxiliary variable.
Necessity:	Each projection that *has* an auxiliary variable *is constrained by* a logical formulation.

Issue 10423 Remove text
Issue 9713 Replace text
Issue 10596 Add Reference Scheme text to auxiliary variable
Issue 9930 Add Synonymous Form

logical formulation *constrains* projection

FL

Definition:	the logical formulation determines which referents of the variables introduced by the projection are in the result of the projection
~~Synonymous Form:~~	~~projection *is constrained by* logical formulation~~
Synonymous Form:	projection *has* constraining formulation
Note:	A logical formulation that constrains a projection restricts the results of the projection. If there is no constraining logical formulation, then there is no restriction other than what is on variables in the projection.

auxiliary variable

FL

Definition:	variable that is introduced by a projection, but which is left out of the result of the projection thereby giving the possibility of duplicate results
Necessity:	Each auxiliary variable *is of* exactly one projection.
~~Reference Scheme:~~	~~the projection that *has* the auxiliary variable and the projection position *of* the variable~~

Reference Scheme: a projection that *has* the auxiliary variable and a projection position *of* the auxiliary variable and the set of concepts that *are ranged over by* the auxiliary variable and the set of logical formulations that *restrict* the auxiliary variable and whether the auxiliary variable *is unitary*

projection position

FL

Definition: positive integer that distinguishes a variable introduced by a projection from others introduced by the same projection

Concept Type: role

variable *has* projection position

FL

Definition: the variable is introduced by a projection and has the unique projection position among the set of variables introduced by that projection

Necessity: Each variable *has* at most one projection position.

Necessity: Each variable that *is in* a projection *has* exactly one projection position.

Necessity: Each auxiliary variable *has* exactly one projection position.

set projection

FL

Definition: projection that *has* no auxiliary variable

Example: A projection formalizing the expression, "customers that are preferred," is on a single variable (customer). There is no auxiliary variable, so the result is necessarily a set.

Issue 9712 add Note

---

bag projection

FL

Definition: projection that *has* an auxiliary variable

Note: A bag projection treats the resulting set of actualities as a set of the corresponding involvements of referents of the projection variables in roles in those actualities. A thing that participates in those involvements may participate in more than one involvement and therefore have multiple "occurrences" in the projection result. In many cases, the use of the projection reduces the set of involvements to the set of things involved (and ignores the fact of multiple occurrence). But in some cases the distinguished involvements/occurrences are important.

Example: A projection formalizing the expression, "account balances of customers that are preferred," is on a variable (account balance) and has an auxiliary variable (customer). Only balances are in the result, but there can be duplicates where multiple customers have the same balance.

Issue 10569 Replace Necessity, remove Necessity
Issue 9712 Remove all Necessity text

---

closed projection

FL

Definition: projection that *is a* closed semantic formulation

~~Necessity:~~ ~~Each closed projection that *defines* a fact type *is* a set projection.~~

~~Necessity:~~ ~~Each variable that *is in* a closed projection *maps to* exactly one role *of* each fact type that *is defined by* the closed projection.~~

~~Necessity:~~ ~~If a closed projection defines a fact type then each role of the fact type is mapped from exactly one variable that is in the closed projection.~~

~~Necessity:~~ ~~A variable maps to a role of a fact type only if the variable is of a closed projection that defines the fact type.~~

~~Necessity:~~ ~~A closed projection that defines a noun concept is on at most one variable.~~

~~Necessity:~~ ~~If a closed projection that defines a noun concept is a bag projection then the noun concept is a role.~~

~~Necessity:~~ ~~Each closed projection that formalizes a definition of a concept defines the concept.~~

~~Necessity:~~ ~~If a closed projection formalizes a definition of a concept then the closed projection defines the concept.~~

~~Necessity:~~ ~~If a closed projection that defines a noun concept is a set projection that is on a variable that is unitary then the noun concept is an individual concept.~~

~~Necessity:~~ ~~Each closed projection that means at most one question.~~

Example: A projection formalizing the expression, “customers that are preferred,” is closed – there is no variable that is not introduced. But within a formulation of the expression, “Each branch must report the number of car models offered by the branch,” the projection of “car models offered by the branch” is open because it binds to a variable (branch) that is introduced outside of the projection.

Issue 10423 Remove text
Issue 9712 Add text
Issue 10622 Add text

closed projection *formalizes* definition

Definition: the definition conveys the meaning formulated by the closed projection and the closed projection refers to the concepts represented in the definition

Example: The one concept ‘local car movement’ can be defined as “one-way car movement that is in-area” or as “car movement that is in-area and that is not round-trip.” Both definitions have the same meaning, but one is formalized in reference to the noun concept ‘one-way car movement’ (defined as “car movement that is not round-trip”) and the other in reference to the characteristic ‘car movement is round-trip’. The two formulations are different but mean the same noun concept.

Necessity: Each closed projection that *formalizes* a definition *of* a noun concept *defines* the noun concept.

Necessity: Each closed projection that *formalizes* a definition *of* a fact type *defines* the fact type.

~~Synonymous Form:~~ ~~definition is formalized by closed projection~~

Issue 9712 Replace text

~~closed projection *defines* concept~~

~~FL~~

~~Definition:~~ ~~the closed projection formalizes a definition of the concept such that the result of the projection is the extension of the concept~~

~~Synonymous Form:~~ ~~concept is defined by closed projection~~

Example: ~~For a noun concept 'barred driver' having the definition, "person that must not rent or drive a car from EU-Rent," a closed projection structures the meaning of the definition. The projection is on one variable, which ranges over the concept 'person.' The satisfying referents of that variable make up the extension of the concept 'barred driver.'~~

Example: ~~For a fact type 'branch is sold out' having the definition, "the branch has no rental car available," a closed projection structures the meaning of the definition. The projection is on one variable which maps to the role 'branch.'~~

Issue 9948 Replace text, remove example

closed projection *defines* noun concept

FL

Definition: the closed projection is on exactly one variable and the closed projection formulates a set of incorporated characteristics sufficient to determine the noun concept

Necessity: Each closed projection that *defines* a noun concept *is on* at most one variable.

Necessity: If a closed projection that *defines* a noun concept *is* a set projection that *is on* a variable that *is unitary* then the noun concept *is* an individual concept.

Note: A closed projection defines a noun concept by formulating a set of incorporated characteristics that determine the noun concept. These incorporated characteristics include:

1. All characteristics of the ranged-over concept of the projection variable of the projection, if there is one.
2. If a logical formulation restricts the projection variable, the meaning of that formulation with respect to the projection variable.
3. If the projection has a constraining formulation and the projection has no auxiliary variable, the meaning of the constraining formulation with respect to the projection variable.
4. If the projection has a constraining formulation and the projection has an auxiliary variable, the characteristic of being involved in an actuality that corresponds to the "basic meaning" of the projection.

Note: When a projection defines a noun concept, it restricts the basic meaning (the set of corresponding actualities) to the involvements in those actualities that are denoted by the projection variable, and further to the things participating in those involvements – the things that play the corresponding role. If there are auxiliary variables, a given thing may participate in more than one such involvement. In many cases, however, the projection introduces only one variable and the actualities are of things having a particular property. ~~If a projection that defines a noun concept has an auxiliary variable, the noun concept is a role of the base meaning (the anonymous fact type) of the projection.~~ If a projection that defines an object type has an auxiliary variable, the object type incorporates the characteristic of being involved in an actuality that also involves a referent of the auxiliary variable, as if the auxiliary variable is existentially quantified. The characterization is from the perspective of a referent of the auxiliary variable.

Example: The ~~noun concept~~ object type 'wrecked car' defined as "car that is disabled by an accident"
A closed projection defines the ~~noun concept~~ object type.
. The projection is on a first variable.
. . The first variable ranges over the concept 'car'.
. The projection is constrained by an existential quantification.
. . The quantification is on a second variable.
. . . The second variable ranges over the concept 'accident'.

. . The quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type 'accident disables vehicle'.
. . . . The 'accident' role is bound to the second variable.
. . . . The 'vehicle' role is bound to the first variable.

~~Example:~~ ~~The role 'renter' defined as "person contractually responsible for a given rental"~~
~~The role is defined by a closed projection.~~
~~. The projection is on a first variable.~~
~~. . The first variable ranges over the concept 'person'.~~
~~. The projection has an auxiliary variable.~~
~~. . The auxiliary variable ranges over the concept 'rental'.~~
~~. The projection is constrained by an atomic formulation.~~
~~. . The atomic formulation is based on the fact type 'person is contractually responsible for rental'.~~
~~. . . The 'person' role is bound to the first variable.~~
~~. . . The 'rental' role is bound to the auxiliary variable.~~

~~This example differs from the previous example primarily in its use of an auxiliary variable (for 'rental') rather than a variable (for 'accident') introduced by an existential quantification. This difference accounts for 'renter' being a role with respect to involvement with a particular rental while 'wrecked car' is not a role with respect to a particular accident, even though at least one accident disables each wrecked car. A significance of the auxiliary variable in defining the role is that a person plays the role of 'renter' once for each corresponding referent of the auxiliary variable – once for each rental for which the person is contractually responsible. The extension of the role is a set of persons. But the role's intension also relates each renter to a particular rental, so besides the extension of the role (a set), there is a multiset of persons with each duplicate person being distinguished by a different rental.~~

Issue 9948 Replace text, remove Necessity

---

closed projection *defines* fact type

Definition: the closed projection is on one variable for each role of the fact type and the closed projection identifies enough characteristics incorporated by the fact type that all of its incorporated characteristics can be determined

Necessity: If a closed projection *defines* a fact type and the closed projection *defines* a noun concept then the fact type *is* a characteristic and the role *of* the characteristic *is coextensive with* the noun concept.

~~Necessity:~~ ~~If a closed projection that *defines* a characteristic *is* a set projection that *is on* a variable that *is unitary* then the characteristic *is* an individual concept.~~

Note: If a closed projection defines a fact type, each variable introduced by the projection, including auxiliary variables, is understood as a point of involvement in actualities that are instances of the fact type. If the projection has a constraining formulation, the meaning of the fact type for each combination of referents, one for each variable, is the proposition meant by the logical formulation. If no logical formulation constrains the projection, then the meaning of the fact type for each combination of referents is that the referents all exist.

Note: A fact type defined by a closed projection incorporates the following characteristics:

1. All characteristics of the concept 'actuality'.

2. Each instance of the fact type involves exactly one thing in each role of the fact type – see ‘variable *maps to* fact type role’ below.
3. If the projection has a constraining formulation and the projection has no auxiliary variable, the meaning of the constraining formulation with respect to the projection variables.
4. If the projection has a constraining formulation and the projection has an auxiliary variable, the meaning of the constraining formulation with respect to the projection variables and of involving a given referent of each auxiliary variable of the projection in its corresponding role of the “base meaning.”

Example: The characteristic ‘car is wrecked’ defined as “the car is disabled by an accident.” The closed projection given in the ~~first~~ example under ‘closed projection *defines* noun concept’ above as defining ‘wrecked car’ also defines this characteristic. The difference between the characteristic and the noun concept is that the extension of the noun concept is the set of wrecked cars while the extension of the characteristic is the set of actualities that a given car is wrecked. Elements of the two extensions are related one-to-one.

Example: The binary fact type ‘accident disables vehicle’ defined as “the accident causes the vehicle to be nonoperational”.

The binary fact type is defined by a closed projection.
. The projection is on a first variable.
. . The first variable ranges over the concept ‘vehicle’.
. The projection is on a second variable.
. . The second variable ranges over the concept ‘accident’.
. The projection is constrained by an existential quantification.
. . The existential quantification is on a third variable.
. . . The third variable is restricted by an objectification.
. . . . The objectification binds to the third variable.
. . . . The objectification considers an atomic formulation.
. . . . . The atomic formulation is based on the fact type ‘vehicle is nonoperational’.
. . . . . . The ‘vehicle’ role is bound to the first variable.
. . The existential quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type ‘event causes state of affairs’.
. . . . The ‘event’ role is bound to the second variable.
. . . . The ‘state of affairs’ role is bound to the third variable.

Issue 9712 Move entry / add text
Issue 9948 Replace role with fact type role

---

variable *maps to* fact type role

FL

Definition: the variable is in a closed projection that defines the fact type that has the fact type role such that for each element in the projection result the referent of the variable is involved in the fact type role in a corresponding actuality in the extension of the fact type

Synonymous Form: fact type role *is mapped from* variable

Necessity: If a closed projection *defines* a fact type then each role *of* the fact type *is mapped from* exactly one variable that *is in* the closed projection and each variable that *is in* the closed projection *maps to* exactly one role *of* the fact type.

Necessity: A variable *maps to* a fact type role only if a closed projection that *is on* the variable *defines* a fact type that *has* the fact type role.

Necessity:	Each variable *maps to* at most one fact type role.
Note:	A fact type role that is mapped from a projection variable of a closed projection incorporates the following characteristics (which are the same as if an object type is defined by the projection with the one modification that all other introduced variables are auxiliary): 1. All characteristics of the ranged-over concept of the variable, if there is one. 2. If a logical formulation restricts the variable, the meaning of that formulation with respect to the variable. 3. If the projection has a constraining formulation, the characteristic of being involved as a referent of the variable in a given actuality denoted by the constraining formulation.
Example:	The 'car' role of the characteristic 'car is wrecked' in the example above under 'closed projection *defines* fact type' is mapped from the one variable in the closed projection that defines the characteristic. Note that the role incorporates the same characteristics as the noun concept 'wrecked car', and is therefore coextensive with it.
Example:	In the binary fact type 'accident disables vehicle' in the example above under 'closed projection *defines* fact type', the 'accident' role is mapped from the first variable and the 'vehicle' role is mapped from the second variable in the projection that defines the binary fact type.

Issue 9733 Add Notes and Example
Issue 10423 Remove text
Issue 9712 Add Necessity and Note
Issue 9948 Revise text

closed projection *means* question

Definition:	the closed projection formulates the question such that the result of the projection answers the question
Necessity:	Each closed projection *means* at most one question.
~~Synonymous Form:~~	~~question *is meant by* closed projection~~
Note:	A question using an interrogative operator such as 'what', 'when', 'where', 'why', or 'how' is generally formulated by a projection on a variable that ranges over a concept that matches the operator. The interrogative 'what' is often used with a designation of a noun concept such as in "What car is available?" in which case the variable ranges over the noun concept 'car'. For each of the other operators the variable ranges over a noun concept fitting to that operator as if 'what' had been used with a designation for that concept. Examples of the correspondence of interrogative operators to noun concepts is shown below. "When is a car available?" What time "How is a car driven?" What method "Where is a car?" What location "Who can drive a car?" What person "Why is a car available?" What cause Note that definition of these nouns (underlined above) is outside the scope of SBVR. However, the concept 'cause' is a role ~~having the more general concept~~ that ranges over the concept 'state of affairs', so an answer to a 'why' question is often formulated using objectification (the last example under objectification considers one state of affairs as a cause of another).
Note:	A true/false question is typically nominalized using the interrogative operator 'whether' as in "The customer asked whether a car is available," but is asked (in English) with no such

operator: “Is a car available?”. The meaning of ‘whether’ in this context is “What truth-value does this proposition have?”. The formulation of such a question is a projection on a variable that ranges over a characteristic type (here called ‘truth-value’) whose instances are the characteristics ‘proposition *is true*’ and ‘proposition *is false*’. The projection is constrained by the truth-value being that of the proposition “a car is available” formulated using proposition nominalization.

Example: “Is a car available?”

The question is meant by a closed projection.
. The projection is on a unitary variable.
. . The variable ranges over the concept ‘truth-value’.
. The projection is constrained by a universal quantification.
. . The universal quantification introduces a second unitary variable.
. . . The second variable ranges over the concept ‘proposition’.
~~. . The universal quantification is restricted by a proposition nominalization.~~
. . . The second variable is restricted by a proposition nominalization.
. . . . The proposition nominalization binds to the second variable.
. . . . The proposition nominalization considers an existential quantification.
. . . . . The existential quantification introduces a third variable.
. . . . . . The variable ranges over the concept ‘car’.
. . . . . The existential quantification scopes over an atomic formulation.
. . . . . . The atomic formulation is based on the fact type ‘car is available’.
. . . . The ‘car’ role is bound to the third variable.
. . The universal quantification scopes over an atomic formulation.
. . . The atomic formulation is based on the fact type ‘proposition has truth-value’.
. . . . The ‘proposition’ role is bound to the second variable.
. . . . The ‘truth-value’ role is bound to the first variable.

Note: An auxiliary variable of a closed projection that means a question is relevant to formulating the meaning of the question, but the question is answered without identifying referents of the auxiliary variable.

10 Providing Semantic and Logical Foundations for Business Vocabulary and Rules

This clause lists and explains foundational concepts taken from respected works on formal logics and mathematics. A mapping is then shown from the concepts of the SBVR Logical Formulation of Semantics Vocabulary to these foundational concepts.

A conceptual model includes both a conceptual schema and a population of facts that conform to the schema. A conceptual model may cover any desired time span, and contain facts concerning the past, present, or future. This notion is distinct from changes made to a conceptual model. Any change to a conceptual model, including any change to any fact in the fact population, creates a different conceptual model. Each conceptual model is distinct and independent, although there may be relationships between conceptual models that share the same conceptual schema.

'Facts' are one of the primary building blocks of SBVR. A 'Fact' is of a particular 'Fact Type.' The lowest level logical unit in SBVR – an 'Atomic Formulation' – is a logical formulation based directly upon a fact type, involving no logical operation. An atomic formulation may be considered as an invocation of a predicate.

Issue 9447 Revise text

SBVR makes no distinction about how facts are known: for example, whether they are asserted as 'ground facts' or obtained by inference. Inferences can only be performed at one time within a particular ~~conceptual~~ fact model. SBVR does not define any kind of inference that can be made between ~~conceptual~~ fact models.

Control over the order in which inferences can be made is a common feature in the automation of inference, as found, for example, in rules engines. SBVR deals with declarative rules expressed from a business perspective. Transitions between ~~conceptual~~ fact models and the mechanization of those rules in an automated system are outside the scope of SBVR.

Closed-world assumptions are often used in automated systems, such as the well-known 'negation by failure' in the Prolog language. The business orientation of SBVR makes it natural to assume open-world semantics by default. For example, if we assume that 'Customers' have some unary fact such as 'Credit OK' then we cannot assume anything like 'Credit not OK' in the absence of this fact. SBVR permits fact types to be explicitly identified as closed where this makes business sense. For example, it may be appropriate to infer 'Credit not OK' for a subset of customers identified as 'Credit-Checked Customers' in the absence of a 'Credit OK' fact.

The detailed definition of SBVR uses the vocabulary defined in SBVR – in other words, SBVR is defined in terms of itself. This inevitably makes the SBVR definition higher order, but this does not force any modeler to produce exclusively higher-order models. Models based on SBVR can be first order if that is what is desired by the modeler.

10.1 Logical Foundations for SBVR

10.1.1 SBVR Formal Grounding Model Interpretation

10.1.1.1 Introduction

Issue 9368 Revise text

The SBVR (Semantics of Business Vocabulary and Business Rules) initiative is intended to capture business facts and business rules that may be expressed either informally or formally. Business rule expressions are classified as formal only if they are expressed purely in terms of fact types in the pre-declared schema for the business domain, as well as certain logical/mathematical operators, quantifiers, etc. Formal statements of rules may be transformed into logical formulations that are used for exchange with other rules-based software tools. Informal statements of rules may be exchanged as un-interpreted comments. The following discussion of business rule semantics is confined to formal statements of business rules. (A closer definition of terms is given as needed later throughout this clause.)

The rest of this clause is structured as follows. 10.1.1.2 provides some basic background and terminology, explaining our usage of terms such as "schema,' "model," and "fact." 10.1.1.3 reviews the approach to choosing open or closed world semantics. 10.1.1.4 provides an overview of the use of quantifiers as well as alethic or deontic modal operators in specifying business rules. 10.1.1.5 and 10.1.1.6 respectively discuss the formal semantics for static, alethic constraints and static, deontic constraints. 10.1.1.7 considers derivation rules. 10.1.1.8 examines dynamic constraints. 10.1.1.9 reviews the option for using higher-order logic.

10.1.1.2 Facts, Schemas, and Models

Issue 9368 Remove text
Issue 11264 Replace text

For any given business, the "universe of discourse" indicates those aspects of the business that are of interest. The term "business domain" is commonly used in the modeling community, with equivalent meaning. A "model," in the sense used here, is a structure intended to describe a business domain, and is composed of a conceptual *schema* (fact structure) and a *population* of ground facts (see later). A *fact* is a proposition taken to be true by the business. Population facts are restricted to elementary and existential facts (see later).

Instantiated roles of facts refer to individuals (such as "Employee 123" or "the sales department"). These individuals are considered as being of a particular type (such as "Employee" or "Department") where *type* denotes "set of possible individuals."

The conceptual schema declares the *fact types* (kinds of facts, such as "Employee works for Department") and ~~*logical*~~ *rules* ~~(typically constraints or derivation rules)~~ relevant to the business domain.

Issue 9579 Revise text
Issue 9368 Remove text/Replace text
Issue 11264 Replace text

~~Logical rules,~~ The terms 'rule' and 'business rule,' in the senses used here, are defined in Clause 12.1.2. ~~regulations or principles governing conduct, procedure, etc. Logical~~ Rules are effectively higher-level facts (i.e., facts about ~~facts~~ propositions), and in a loose sense are also sometimes considered under the generic term 'fact.' For clarity, the term "ground fact" is used here to explicitly exclude such (meta) facts.

Constraints are used to define bounds, borders, or limits on fact populations, and may be static or dynamic. A *static constraint* imposes a restriction on what fact populations are possible or permitted, for each fact population taken individually.

Static constraint
Each Employee was born on **at most one** Date

A *dynamic constraint* imposes a restriction on transitions between fact populations.

Dynamic constraint
A person's marital status may change from single to married, but not from divorced to single

Derivation rules indicate how the population of a fact type may be derived from the populations of one or more fact types or how a type of individual may be defined in terms of other types of individuals and fact types.

Derivation rules
Person_1 is an uncle of Person_2 **if** Person_1 is a brother of **some** Person_3 **who** is a parent of Person_2,
Each FemaleAustralian **is a** Person **who** was born in Country 'Australia' **and** has Gender 'Female'

A model of the kind considered here is a *fact model*, not a process model. The term *knowledge base* is sometimes used to reflect this focus (on what is known, as opposed to what must be done). At least two kinds of fact model may be specified: reality models; and in-practice models. Although both these models use the same set of fact types, they may differ in the constraints imposed on those fact types. A *reality model* of a business domain is intended to reflect the constraints that actually apply to the business domain in the real world. An *in-practice model* of a business domain reflects the constraints that the business chooses in practice to impose on its knowledge of the business domain.

Suppose the following two fact types are of interest: Employee was born on Date; Employee has PhoneNumber. In the real world, each employee is born, and may have more than one phone number. Hence the reality model includes the constraint "**Each** Employee was born on **at least one** Date" and allows that "**It is possible that the same** Employee has **more than one** PhoneNumber." Now suppose that the business decides to make it optional whether it knows an employee's birth date. Suppose also that the business is interested in knowing at most one phone number for any given employee. In this case, the in-practice model excludes the reality constraint "**Each** Employee was born on **at least one** Date," but it includes the following constraint that doesn't apply in the reality model: **Each** Employee has **at most one** PhoneNumber.

Constraint differences between reality and in-practice models have some restrictions (for instance, in-practice uniqueness constraints need to be at least as strong as the corresponding real world uniqueness constraints, and if a fact type role is optional in the real world it is optional in the in-practice world, but the converse need not apply).

Reality schemas are sometimes constructed first to help determine in-practice schemas. Although a population may be added to any schema to form a model, it is common to add populations only to in-practice schemas. So in-practice models are more common than reality models. The possibility of incomplete knowledge arises for both reality and in-practice models but is more prevalent with in-practice models since these tend to include more optional aspects. Adoption of open or closed world assumptions is discussed in 10.1.1.3.

Example of incomplete knowledge
The business might know just some of a given employee's phone numbers

Issue 9368 Remove text

We use the term "fact model" or "knowledge base" in a broad sense. Conceptually, the fact model is represented by a set of sentences, each of which connotes either a ~~logical~~ rule or a ground fact. The fact model may be fully automated (as in, say, a database system), manual (as in, say, a paper record system), or semi-automated. The knowledge may even be stored in human memory (belonging to the business domain experts who may be collectively regarded as the authoritative source of those business facts that are of interest). However, the knowledge must ultimately be expressible by sentences communicated between humans.

Issue 9368	Remove text
Issue 9579	Revise text
Issue 9447	Revise text
Issue 11264	Replace text

A fact model is ~~specified as~~ a conceptual model of the business domain, using a suitable high level vocabulary and language that is readily understood by the business domain experts. Typically this language will be a formal subset of a natural language. In particular, the language is not a machine-oriented technical language (such as C# or Java ~~or SQL~~) that might be used to implement a system to enforce at least some of the business rules included in the model. Business domain models are meant to capture the relevant business rules, not to implement them. Whether a given business rule is implemented at all, or how it might be implemented (automated, semi-automated, or manual) are not issues here. Typically however, it is expected that many business rules specified in a business domain model will likely be enforced in an automated way; and in such cases, the ~~logical~~ rules need to be formally expressed.

Any fact model passes through a sequence of *states*, each of which includes a set of *ground facts*, which are either elementary or existential. Roughly speaking, an *elementary fact* is a declaration that an individual has a property, or that one or more individuals participate in a relationship, where the fact cannot be split into simpler facts with the same individuals (without information loss).

Examples of elementary facts

The Country named 'Australia' is large

The Prime Minister named 'John Howard' was born in the Country named 'Australia'

An elementary fact may be treated as an instantiation of a typed, irreducible predicate of interest to the business, except that multiple fact type readings using different predicates, possibly based on different orderings of the individuals, are considered to express the same fact if they mean the same. Individuals are typically denoted by definite descriptions.

The sentences (1) and (2) below express the same fact:

(1) The President named 'Mary McAleese' governs the Country that has the Country Name 'Ireland.'

(2) The Country that has the Country Name 'Ireland' is governed by the President named 'Mary McAleese.'

"The President named 'Mary McAleese'" is treated here as shorthand for "The President who has the President Name 'Mary McAleese'"

Instead of definite descriptions, proper names may be used if they function as individual constants in the business domain. Lexical individuals denote themselves. Individual constants may also be introduced as abbreviations of definite descriptions.

Example of a self-denoting lexical individual

The country code 'US'

We use the term “fact” in the sense of “proposition taken to be true by the business” (i.e., the business members are prepared to act as if they believed the proposition is true; their attitude toward the proposition is one of epistemic commitment). This sense of epistemic commitment does not require any special interpretation of logical operators, or use of epistemic or doxastic logic. The logical connectives (and, or, not, if-then, etc.) may be interpreted just like truth functional operators (conjunction, disjunction, negation, material implication, etc.) in 2-valued classical logic. An *existential fact* is used to simply assert the existence of an individual,

Example of an existential fact
There is a Country that has the Country Code ‘US’

A *fact type* may be identified by one or more fact type readings that declare typed predicates.

Examples of fact type readings

The President named 'Mary McAleese' governs the Country that has the Country Name 'Ireland'

    is an instance of the fact type

President governs Country

The Country that has the Country Name 'Ireland' is governed by the President named 'Mary McAleese'

    is an instance of the fact type

Country is governed by President

Subclause 10.1.1 uses initial capitals to denote types of individuals (other styles may be used for this purpose), and in general allows predicates in mixfix notation.

Example of mixfix notation
President visited Country on Date

More conventional but less readable syntaxes may also be used.

Example of more conventional notation

President governs Country

    may be expressed as

governs(*x*:President; *y*:Country)

Each predicate has a fixed arity, so variadic predicates are not supported.

For example, the unary "smokes" predicate in 'Person smokes' is considered to be different from the binary "smokes" predicate in 'Person smokes Cigar Brand.'

Note that we do not identify untyped predicates simply by their name and arity.

For example, the “has” in 'Person has Disease' is considered to be a different predicate from the "has" in 'Disease has Cure.'

Issue 9368 Remove text
Issue 11264 Replace text

The fact model includes both the conceptual schema and the ground fact population (set of fact instances that instantiate the fact types in the schema). The conceptual schema includes a generic component and a domain-specific component. The generic component is common to all conceptual schemas: this includes relevant axioms from logic and mathematics[1]. The domain-specific component includes the concept definitions and declarations of the ground fact types and ~~logical~~ business rules relevant to the specific business domain.

Trivially, each fact model includes existential facts to declare the existence of generic constants such as numbers, but we ignore these in our discussion, confining the use of “population” to the domain-specific population of interest. With that understanding, the fact model at any point in time may be declared as a set of sentences that collectively express the conceptual schema and the fact population of the domain-specific fact types in the conceptual schema.

Although in practice the conceptual schema may evolve over time (if the business domain changes its structure or scope of interest) we ignore schema evolution here, treating the conceptual schema as fixed. Schema evolution may be handled as a metametalevel concern. Model exchange must be enabled between a system supporting SBVR and other systems identified as desirable targets for interoperability. Any exchange of a fact model takes place at a given point in time, and at that time the conceptual schema is fixed (later exchanges may be used to update the fact model as required). Also, when a necessity is originally stated, the intent is that by default the ~~logical~~ rule should stay in force.

In contrast to the conceptual schema, the (domain-specific) fact population is typically highly variable.

For example, the fact type "Employee works on Project" may initially have no instances, but over time thousands of employees may be added or removed from various project teams.

Figure 10-1 provides a simplified picture of this situation, indicating that the fact model of sentences expressing population facts (instances of domain-specific fact types) is a varset (variable-set) whose population at any given time is a set of facts.

1. For a detailed discussion of one way to formalize this, see [Halp1989]. A fact model is specified as a set of sentences in a language based on predicate logic with identity. An interpretation is defined in the usual way (e.g., each predicate symbol maps onto a relation over the domain of individuals) and a model (not the same as fact model) is an interpretation where all the sentences are true.

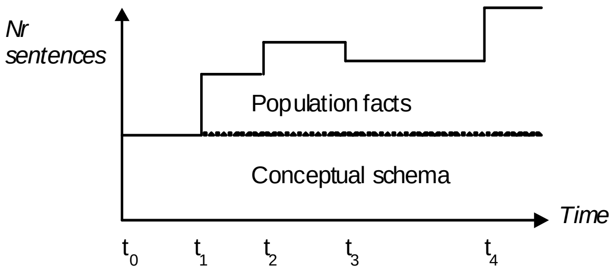


Figure 10.1 - Evolution of the fact model (schema plus ground fact instances)

The fact model may be initially empty or pre-populated with some facts. The fact model may expand or shrink over time as facts are added or removed from it. At any point in time, the fact model includes a set of facts. Figure 10-2 depicts this situation in more detail, using a labeled box to denote a fact instance (f1 = fact 1, etc.).

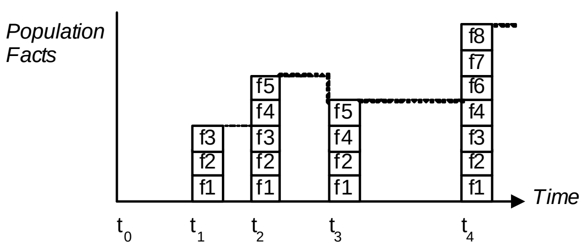


Figure 10.2- Evolution of the ground fact population

In treating a fact model as a varset of facts that typically changes over time, we allow facts to be added or deleted (see Figure 10-22). We might delete a fact because we revise our decision on whether it is (taken to be) true (for instance, we might discover a mistake), or because we decide that fact is no longer of interest. Now consider the following description by [Anto2001] of non-monotonic logic.

> The term “non-monotonic logic” covers a family of formal frameworks devised to capture and represent *defeasible inference*, i.e., that kind of inference of everyday life in which reasoners draw conclusions tentatively, reserving the right to retract them in the light of further information. Such inferences are called “non-monotonic” because the set of conclusions warranted on the basis of a given knowledge base does not increase (in fact, it can shrink) with the size of the knowledge base itself. This is in contrast to classical (first-order) logic, whose inferences, being deductively valid, can never be “undone” by new information.

On the surface, it would appear that we are committing to a non-monotonic logic, given that we allow facts to be deleted in going from one state to another. However it seems reasonable to formalize those ~~logical~~ business rules that are static constraints in terms of classical, non-monotonic logic.

> For example, we might formalize the static constraint that each person was born on some date as an SBVR logical formulation of the formula $\forall x$:Person $\exists y$:Date *x* was born on *y.*

In classifying the ~~logical~~ rule as a static constraint, we assert that it is true for each state of the fact model, taken individually. This seems to be enough, from the point of view of exchanging fact models, which always involves just one state at that time. Note also that the characterization of fact models as variable sets of sentences does not claim that propositions change their truth value over time. We regard propositions to be atemporal: they are timelessly true or false, so never change their truth value.

At least superficially, it is possible that a sentence in one fact model state expresses a different proposition from that expressed by the same sentence in another fact model state. For example, the meaning of time-deictic sentence occurrences depends on the time they were uttered or inscribed.

> For instance, given the static constraint that each person lives in at most one country, we might assert for the fact model state 1 that Terry lives in Australia, for fact model state 2 we delete "Terry lives in Australia" and add that Terry lives in Utah, and for fact model state 3 we delete "Terry lives in Utah" and add that Terry lives in Australia. This does not involve any change in proposition truth values, because different propositions were being asserted in the different states. Here the verb phrase "lives in" means "currently lives in," where 'currently' may be unpacked into a time-indexed expression that includes the time of that fact model state.

10.1.1.3 Open/Closed World Semantics

Adopting *closed world* semantics basically means that all relevant facts are known (either as primitives – not defined in terms of other things – or derivable). So if a proposition cannot be proved true, it is assumed to be false. This *closed world assumption* entails *negation by failure*, since failure to find a fact implies its negation. *Open world semantics* allows that some knowledge may be incomplete; so if a proposition and its negation are both absent, it is unknown whether the proposition is true.

In modeling any given business domain, attention can be restricted to propositions *of interest* to that domain. If a proposition is not relevant to that domain, it is not included as a fact there, but we do not assume it is false; rather we simply dismiss it from consideration. For any business domain, we have a *finite set of types of individuals and fact types* (typed predicates), and any type of individual or fact type outside this set is simply disregarded.

It is a practical issue whether one's knowledge pertaining to the population of a given fact type is complete or not, since this may impact how the business derives other facts (e.g., negations) or how it reacts to query results (e.g., whether to treat "not" as "not the case" or merely "not known to be the case"). So we regard the issue of open/closed world semantics to be relevant to the fact model itself, not just automated implementations of the fact model.

Many implementations treat "not" in the closed-world sense of either "not known" (as a primitive or derivable fact), i.e., negation as failure, or "not known as a primitive fact," i.e., semi-positive negation. For instance, Prolog-based rule engines rely on negation by failure, and the "not" in SQL means "not recorded in a base table or derivable in a view."

> SQL example,
>
> Figure 10-3 depicts the relational schema and a sample population for a database fragment used to store the employee number and name of each employee, as well as the cars they drive (if any).

employee (empNr, empName)

Drives (empNr, carRegNr)

Employee			**Drives**	
empNr	*empName*		*empNr*	*carRegNr*
1	John Smith		1	ABC123
2	Ann Jones		2	AAA246
3	John Smith		2	DEF001

Figure 10.3- A sample database storing some facts about employees

Suppose we want to know the employee number and name of each employee. In SQL we might formulate this query as **select** * **from** Employee, which returns the three rows of data shown in the Employee table. This result returns the employee number and name of those employees referenced in the database. Whether this includes all the employees in the business domain depends on whether the database is complete with respect to the population of the elementary fact type Employee has EmployeeName. If it is complete, the fact type is closed, and we may treat the SQL query as equivalent to our intended query about the business domain. If it is not complete, then the fact type is open, and we may need to take into account that there may be more employees than listed in the result.

Knowledge about completeness is typically not stored in databases, although in principle it could be. Users typically adopt the closed world assumption when interpreting data in relational databases. If independently of the database system they know how complete the data is, they may take that into account in deciding how completely the query results from the database system relate to the real world of their business domain.

Suppose we want to know the employee number of each employee who does not drive a car for the database shown in Figure 10-3. In SQL we might formulate this query as **select** empNr **from** Employee **where** empNr **not in** (**select** empNr **from** Drives). This returns just one employee number (viz. 3). Whether this covers all the non-driver employees in the business domain depends on whether the population of the two fact types (Employee has EmployeeName and Employee drives Car) is complete or not. Again, this knowledge about completeness could be stored in the database, but typically isn't, in which case users need to rely on their own knowledge about completeness to decide whether the data returned is complete or not.

The approach adopted here is fact-based (as opposed to attribute-based), where each fact type is modeled as a type of relationship, never as an attribute. Annex J provides extended examples of fact types expressed in this way using a popular fact-based modeling approach.

Example fact-based representation of a database schema

The information structure implied by the database schema shown in Figure 10-3 can be expressed as a set of fact types and constraints as follows, using the capitalized mixfix notational style described earlier:

Types of individuals

Employee

Car

Employee Number

Employee Name

Car Registration Number

(Note that here Employee and Car represent the kind of real world individuals that typically change state. Employee Number, Employee Name and Car Registration Number represent simple self-identifying lexical constants.)

Fact types

Employee has Employee Number

Employee has Employee Name

Car has Car Registration Number

Employee drives Car

Constraints

Each Employee has exactly one Employee Number.

For each Employee Number, at most one Employee has that Employee Number.

Each Employee has exactly one Employee Name.

Each Car has exactly one Car Registration Number.

For each Car Registration Number, at most one Car has that Car Registration Number.

It is possible that the same Employee drives more than one Car and that more than one Employee drives the same Car.

Completeness claims about a schema can be clarified by referring to whether fact type roles are mandatory and whether instances of fact type roles are unique. A fact type role is mandatory if, for each state of the fact model, each instance in the population of the associated type of individual must play that fact type role. A fact type role (or combination of fact type roles) is unique if, for each state of the fact model, each individual that instantiates the fact type role (or each sequence of individuals that instantiates the fact type role sequence) does so once only.

In the schema given above:

each Employee has exactly one Employee Name (mandatory fact type role) but it is optional whether an Employee drives a car.

each Employee has exactly one Employee Name: the Employee fact type role is unique in this fact type but the Employee Name fact type role is not (an Employee has only one Employee Name, but the same Employee Name could refer to more than one Employee).

To consider completeness claims, we can express additional requirements in terms of the fact model populations of types of individuals and the sequences of fact type roles they play in the population of fact types. A schema, as described earlier, is useful for clarifying the conditions under which completeness claims may be made.

Referring again to the Employee-Car schema, for any state of the fact model, let *pop*(*I*) denote the fact model population of the type of individual *I* in that state, and let *pop*(*F*) denote the fact model population of the fact type role sequence for the fact type *F* in that state. If the fact model is complete with regard to capturing the real world business domain, then for each state of the fact model the following three additional conditions are satisfied:

(1) pop(Employee)	=	set of employees in the (real world) business domain (at that time)
(2) pop(Car)	=	set of cars in the business domain
(3) pop(Employee drives Car)	=	set of (employee, car) pairs from pop(Employee) × pop(Car) where that employee drives that car in the business domain.

Requirements (1) and (2) declare that the fact model population of the Employee and Car types of individuals always matches that of the business domain being modeled. We may regard this as asserting the closed world assumption for those types of individuals. Requirement (3) asserts that for those employees and cars that are included in the fact model, if they drive a car then this fact is known. In combination, requirements (1) – (3) entail the closed world assumption for the drives fact type (if an employee drives a car in the business domain, this is known in the fact model).

Given the schema, and requirement (1), the closed world assumption is implied for the employee name fact type. This follows because of the mandatory and uniqueness constraints on the first fact type role (employee is closed, so we have all the employees; having a name is mandatory, so we have at least one name for each employee; the uniqueness constraint means that each employee has at most one name; so for all employees we now have all their names). Note that open world semantics still applies to the employee name fact type; in the presence of (1) and the constraints, this is equivalent to closed world semantics for that fact type.

For any given schema, the business might have complete knowledge about some parts and incomplete knowledge about other parts. So in practice, a mixture of open and closed world assumptions may apply. We use the term "*local closure*" (or "relative closure") for the application of the closed world assumption to just some parts of the overall schema. One might assume open world semantics by default, and then apply local closure to specific parts as desired; or alternatively, assume closed world semantics by default and then apply "local openness." We adopt the former approach as it seems more realistic when modeling real business domains.

Closure (i.e., local closure) may be explicitly asserted for any type of individual, on a one-by-one basis, to declare that for each state the fact model population agrees with that of the population of that type of individual in the actual business domain. The relevant meta-fact type is: "type of individual is closed." It may be reasonable to assume closure for types of individual by default, but it seems unrealistic to assume closure for predicates.

Closure may also be asserted for fact types. *Semi-closure* is with respect to the fact model population of the types of individual playing a fact type role in the predicate. If closure has also been declared for these types, then (full) closure also holds for the fact type (i.e., closure with respect to the domain population of the types of individuals). The relevant meta-fact types are: “fact type is semi-closed” and “fact type is closed.” The meta-fact type “concept is closed” applies to both types of individuals and fact types, since both are concepts.

As seen earlier, closure for a fact type is sometimes implied. A *functional fact type role* is the complete argument of a uniqueness constraint. For schemas whose functional fact type roles are also functional in the business domain, the following implications hold. If a predicate includes a mandatory, functional fact type role, then that predicate is semi-closed by implication (as in the employee name example earlier). This result may be generalized to the case of a mandatory fact type role that has a frequency constraint of exactly *n* (although some attribute-based approaches do not deal reliably with various n-ary cases). If a type of individual has a set of functional fact type roles that are disjunctively mandatory and mutually exclusive (in other words, they are spanned by an exclusive-or constraint), then the predicates that include those fact type roles are semi-closed by implication. If the type of individual has also been declared complete in such cases, then (full) closure applies.

For many fact types in a business domain, especially those without functional fact type roles, it is impractical to include all the negative instances as primitive facts.

For example, for the fact type “Employee drives Car,” there might be many thousands of cars, so one would normally not explicitly include negated facts such as “Employee 1 does **not** drive Car ‘AAA246’.”

In some cases however, especially with functional fact type roles or when the population is small, it is practical to include negated facts as base facts.

Example

To provide a concrete example of the alternative, we can consider the unary fact type 'Person smokes,' and three instances of Person: Fred, Sue, and Tom (for simplicity we will ignore reference schemes and assume that a person may be identified by their first name).

Assume that we know that Fred smokes. If we use open-world semantics, then it is unknown whether Sue or Tom smoke. If we apply closed world semantics, then the absence of facts that Sue or Tom smoke entails that they don't smoke.

If, for each Person, it is known whether that person smokes or not, then we could adopt one of two approaches to model our business domain.

(a) Use two unary fact types, such as 'Person smokes' and 'Person is a nonsmoker,' with an exclusive-or constraint between the fact types. In other words, a Person must play one fact type role or the other, but cannot play both.

(b) Use a binary fact type such as 'Person has Smoker Status' where Smoker Status is indicated by some suitable code such as 'S' or 'NS' (for smoker or nonsmoker respectively), together with the constraint that a Person has exactly one Smoker Status.

In each of these cases, negated facts are explicitly treated as primitive facts and the predicates are given open world semantics. Semi-closure is implied because of the constraints.

Now consider a business domain where we know that Fred smokes, and that Sue doesn't smoke, but are unsure whether Tom smokes. In this case we have three alternative approaches that we could consider.

(a) Use two unary fact types, such as 'Person smokes' and 'Person is a nonsmoker,' with an exclusion constraint between the fact types. In other words, a Person may play one fact type role or the other (but not both) or may play neither fact type role. For the given scenario, we would have the facts 'Fred smokes,' 'Sue is a nonsmoker' and no information for Tom.

(b) Use a binary fact type such as 'Person has Smoker Status' where Smoker Status is indicated by some suitable code such as 'S' or 'NS' (for smoker or nonsmoker respectively), together with the constraint that a Person has zero or one Smoker Status value. For the given scenario we would have the facts 'Fred has Smoker Status 'S,'' 'Sue has Smoker Status 'NS,'' and no information for Tom.

(c) Use a binary fact type such as 'Person has Smoker Status' where Smoker Status is indicated by some suitable code such as 'S,' 'NS,' or '?' (for smoker, nonsmoker, or unknown, respectively), together with the constraint that a Person has exactly one Smoker Status. In this case we treat the 'unknown' value ('?') like any other value using 2-valued logic, rather than adopt a generic null based on 3-valued logic, as in SQL. For the given scenario we would have the facts “Fred has Smoker Status 'S,'” “Sue has Smoker Status 'NS,'" and “Tom has Smoker Status '?'."

The above discussion indicates some ways of declaring and inferring various kinds of closure in the underlying fact model, based on a default, open world semantics. Here, all business rules that are parsed as formal are given a logical formulation based on the fact types in the underlying model. When people formulate queries on the model population, they may either adopt whatever closure guarantees are formally captured in the model, or instead informally rely on their own knowledge about closure to decide whether the data returned is complete or not. Such informal knowledge is outside the fact model, and does not impact the formal semantics of the logical formulation used in exchanging fact models.

In addition to specifying fact models at a conceptual level, languages may be defined for querying these models directly at a conceptual level. These may include features such as the ability to specify projections in the scope of negation, as well as projections in the scope of the "whether-or-not" operator which is used to perform conceptual left outer joins [Bloe1996. Bloe1997] . Further details are outside the scope of this subclause.

10.1.1.4 Quantifiers and Modalities

Issue 9368 Remove text

Static constraints apply to each state of the fact model, taken individually. These may typically be expressed as logical formulations that are equivalent to formulae in 2-valued, first-order predicate calculus with identity. The 2-valued restriction applies because the fact types on which the ~~logical~~ rules are based are elementary (irreducible), so their instances never involve nulls. For convenience, we can use mixfix notation for predicates, and predefine some numeric quantifiers in addition to ∀ and ∃. Table 10-1 summarizes the pre-defined quantifiers.

Table 10.1- Quantifiers

Symbol	*Example*	*Name*	*Meaning*
$\forall$	$\forall x$	Universal Quantifier	For each and every *x*, taken one at a time
$\exists$	$\exists x$	Existential Quantifier	At least one *x*
$\exists^{1}$	$\exists^{1} x$	Exactly-one quantifier	There is exactly one (at least one and at most one) *x*
$\exists^{0..1}$	$\exists^{0..1} x$	At-most-one quantifier	There is at most one *x*
$\exists^{0..n}$ ($n \geq 1$)	$\exists^{0..2} x$	At-most-*n* quantifier	There is at most *n* *x* *Note:* *n* is always instantiated by a number ≥ 1. So this is really a set of quantifiers (*n* = 1, etc.)
$\exists^{n..}$ ($n \geq 1$)	$\exists^{2..} x$	At-least-*n* quantifier	There is at least *n* *x* *Note:* *n* is always instantiated by a number ≥ 1. So this is really a set of quantifiers (*n* = 1, etc.)
$\exists^{n}$ ($n \geq 1$)	$\exists^{2} x$	Exactly-*n* quantifier	There is at exactly (at least and at most) *n* *x* *Note:* *n* is always instantiated by a number ≥ 1. So this is really a set of quantifiers (*n* = 1, etc.)
$\exists^{n..m}$ ($n \geq 1$, $m \geq 2$)	$\exists^{2..5} x$	Numeric range quantifier	There is at least *n* and at most *m* *x*

The additional existential quantifiers are easily defined in terms of the standard quantifiers.

For example, the exactly-two quantifier $\exists^2$ may be defined as follows. Let x, x_1, x_2 be individual variables and Φx be a well formed formula with no free occurrences of x_1, x_2. Then:

$$\exists^2 x\ \Phi x \ =_{df}\ \exists x_1 \exists x_2\ [\Phi x_1\ \&\ \Phi x_2\ \&\ x_1 \neq x_2\ \&\ \forall y(\Phi y \supset (y = x_1 \vee y = x_2))]$$

Definition schemas for the other quantifiers may be found on page 4-11 of [Halp1989].

The ~~logical~~ rule formulations covered here may use any of the basic alethic or deontic modal operators shown in Table 10.2. These modal operators are treated as proposition-forming operators on propositions (rather than actions). Other equivalent readings may be used in whatever concrete syntax is used to originally declare the logical rule (e.g., "necessary" might be replaced by "required," and "obligatory" might be replaced by "ought to be the case"). Derived modal operators may also be used in the surface syntax, but are translated into the basic modal operators plus negation (~).

For example, "It is impossible that p" is defined as "It is not possible that p" ($\sim\Diamond p$), and "It is forbidden that p" is defined as "It is not permitted that p" ($\boldsymbol{F}p =_{df} \sim\boldsymbol{P}p$).

Issue 9475/9945Replace Table

Table 10.2 - Quantifiers

Modality		Modal Formula		applying modal negation rules ... = (Logically Equivalent) Modal Formula	
		Formula	Reading (Verbalized as):	Formula	Reading (Verbalized as):
alethic	necessity	$\Box p$	It is necessary that p	$\sim\Diamond\sim p$	It is not possible that not p
	the negation of necessity: **non-necessity**	$\sim\Box p$	It is not necessary that p	$\Diamond\sim p$	It is possible that not p
	possibility	$\Diamond p$	It is possible that p	$\sim\Box\sim p$	It is not necessary that not p
	the negation of possibility: impossibility	$\sim\Diamond p$	It is not possible that p It is impossible that p	$\Box\sim p$	It is necessary that not p
	contingency	$\Diamond p\ \&\ \sim\Box p$	It is possible but not necessary that p	$\sim(\sim \Diamond p \vee \Box p)$	It is neither impossible nor necessary that p
deontic	obligation	**O**p	It is obligatory that p	$\sim\boldsymbol{P}\sim p$	It is not permitted that not p

Table 10.2 - Quantifiers

	the negation of obligation: non-obligation	~**O**p	It is not obligatory that *p*	***P***~*p*	It is permitted that not *p*
	permission	**P**p	It is permitted that *p*	~***O***~*p*	It is not obligatory that not *p*
	the negation of permission: prohibition	~**P**p **F**p	It is not permitted that *p* It is prohibited that *p* It is forbidden that *p*	***O***~*p*	It is obligatory that not *p*
	optionality	***P****p* & ~***O****p*	It is permitted but not obligatory that *p*	~ (~***P****p* v ***O****p*)	It is neither prohibited nor obligatory that *p*

Table Legend:

□	necessity
◊	possibility
O	obligation
P	permission
F	forbidden
≡	logically equivalent
&	and
v	or (inclusive-or)
~	not
p	some proposition

Issue 10444 Revise text

The following *modal negation rules* apply: it is not necessary that ≡ it is possible that not ($\sim\Box p \equiv \Diamond\sim p$); it is not possible that ≡ it is necessary that not ($\sim\Diamond p \equiv \Box\sim p$); it is not obligatory that ≡ it is permitted that it is not the case that ($\sim\boldsymbol{O}p \equiv \boldsymbol{P}\sim p$); it is not permitted that ≡ it is obligatory that it is not the case that ($\sim\boldsymbol{P}p \equiv \boldsymbol{O}\sim p$). In principle, these ~~logical~~ rules could be used with double negation to get by with just one alethic and one deontic ~~modal~~ operator (e.g., $\Diamond p$ could be defined as $\sim\Box\sim p$, and $\boldsymbol{P}p$ could be defined as $\sim\boldsymbol{O}\sim p$).

Every constraint has an associated *modality,* determined by the logical modal operator that functions explicitly or implicitly as its main operator. We can distinguish between positive, negative, and default verbalizations of constraints. In positive verbalizations, an alethic modality of necessity is often assumed (if no modality is explicitly specified), but may be explicitly prepended.

For example, the following static constraint

C1 **Each** Person was born in **at most one** Country.

may be explicitly verbalized with an alethic modality thus:

C1' **It is necessary that each** Person was born in **at most one** Country.

We interpret this in terms of *possible world semantics*, as introduced by Saul Kripke and other logicians in the 1950s. A proposition is necessarily true if and only if it is true in all possible worlds. With respect to a *static constraint* declared for a given business domain, a possible world corresponds to a *state of the fact model* that might exist at some point in time.

The constraint C1 in the example above means that for each state of the fact model, each instance in the population of Person is born in at most one country.

A proposition is possible if and only if it is true in at least one possible world. A proposition is impossible if and only if it is true in no possible world (i.e., it is false in all possible worlds).

In the example above, constraint C1 may be reformulated as the following negative verbalization:

C1" **It is impossible that the same** Person was born in **more than one** Country.

In practice, both positive and negative verbalizations are useful for validating constraints with domain experts, especially when illustrated with sample populations that provide satisfying examples or counter-examples respectively. The approach described here does not stipulate a high level language for ~~logical~~ rule verbalization, so many alternative verbalizations may be used.

Many business constraints are deontic rather than alethic in nature. To avoid confusion, we recommend that, when declaring a deontic constraint, the deontic modality always be explicitly included.

Consider the following static, deontic constraint.

C2 **It is obligatory that each** Person is a husband of **at most one** Person.

If this ~~logical~~ rule were instead expressed simply as "**each** Person is a husband of **at most one** Person," it would not be obvious that a deontic interpretation was intended. The deontic version indicates a condition that *ought* to be satisfied, while recognizing that the condition *might* not be satisfied. Including the obligation operator makes the ~~logical~~ rule much weaker than a necessity claim, since it allows that there could be some states of the fact model where a person is a husband of more than one wife (excluding same-sex unions from instances of the husband relationship). For such cases of polygamy, it is important to know the facts indicating that the person has multiple wives. Rather than reject this possibility, we allow it and then typically perform an action that is designed to minimize the chance of such a situation arising again (e.g., send a message to inform legal authorities about the situation).

Constraint C2 may be reformulated as either of the following negative verbalizations:

C2' **It is forbidden that the same** Person is a husband of **more than one** Person.

C2" **It is not permitted that the same** Person is a husband of **more than one** Person.

In practice, most ~~logical~~ statements of business rules include only one modal operator, and this operator is the main operator of the whole ~~logical~~ rule ~~expression~~ statement. For these cases, we simply tag the constraint as being of the modality corresponding to its main operator, without committing to any particular modal logic. Apart from this modality tag, there are some basic modal properties that may be used in transforming the original high level expression of the ~~logical~~ rule into a standard logical formulation. At a minimum, these include the modal negation rules.

We also make use of equivalences that allow one to move the modal operator to the front of the formula.

> For example, suppose the user formulates ~~logical~~ rule C1 instead as:
>
> **For each** Person, **it is necessary that that** Person was born in **at most one** Country.
>
> The modal operator is now embedded in the scope of a universal quantifier. To transform this ~~logical~~ rule formulation to a standard logical formulation that classifies the ~~logical~~ rule as an alethic necessity, we move the modal operator before the universal quantifier, to give:
>
> **It is necessary that each** Person was born in **at most one** Country.

For such tasks, we assume that the Barcan formulae and their converses apply, so that $\Box$ and $\forall$ are commutative, as are $\Diamond$ and $\exists$. In other words:

$$\forall x \Box \mathrm{F}x \equiv \Box \forall x \mathrm{F}x$$

$$\exists x \Diamond \mathrm{F}x \equiv \Diamond \exists x \mathrm{F}x$$

While these commutativity results are valid for all normal, alethic modal logics, some philosophical concerns have been raised about these equivalences (e.g., see subclauses 4.6-4.8 of [Girl2000]).

> As a deontic example, suppose the user formulates ~~logical~~ rule C2 instead as:
>
> **For each** Person, **it is obligatory that that** Person is a husband of **at most one** Person.
>
> Using a deontic variant of the Barcan equivalences, we commute the $\forall$ and ***O*** operators, thus transforming the ~~logical~~ rule formulation into the deontic obligation:
>
> **It is obligatory that each** Person is a husband of **at most one** Person.

So far, our ~~logical~~ rule examples have included just one modal operator, which (perhaps after transformation) also turns out to be the main operator. Ignoring dynamic aspects, we may handle such cases without needing to commit to the formal semantics of any specific modal logic. The only impact of tagging a ~~logical~~ rule as a necessity or obligation is on the ~~logical~~ rule enforcement policy. Enforcement of a necessity rule should never allow the necessity rule to be violated. Enforcement of an obligation rule should allow states that do not satisfy the obligation rule ~~condition~~, and take some other remedial action: the precise action to be taken is not specified in SBVR, as it is out of scope ~~for the proposal (logical rule enforcement and logical rule management are to be addressed in separate proposals)~~. At any rate, a business person ought to be able to specify a deontic rule first at a high level, without committing at that time to the precise action to be taken if the condition is not satisfied; of course, the action still needs to be specified later in refining the ~~logical~~ rule to make it fully operational.

10.1.1.5 Static, Alethic Constraints

Issue 9368 Remove text

~~Logic~~ Rule formulations may make use of two alethic modal operators: $\Box$ = it is necessary that; $\Diamond$ = it is possible that. Static constraints are treated as alethic necessities by default, where each state of the fact model corresponds to a possible world.

> Given the fact type Person was born in Country, the constraint "**Each** Person was born in **at most one** Country" may be captured by an SBVR logical formulation that may be automatically translated to the formula $\forall x$:Person $\exists^{0..1}y$:Country *x* was born in *y*. This formula is understood to be true for each state of the knowledgebase. Pragmatically, the ~~logical~~ rule is understood to apply to all future states of the fact model, until the ~~logical~~ rule is revoked or changed. This understanding could be made explicit by prepending the formula with $\Box$ to yield the modal formula $\Box\forall x$:Person $\exists^{0..1}y$:Country *x* was born in *y*.

For compliance with Common Logic, formulae such as those in the preceding example could then be treated as irregular expressions, with the modal necessity operator treated as an uninterpreted symbol (e.g., using "[N]" for $\Box$). However we leave this understanding as implicit, and do *not* commit to any particular modal logic.

For the model theory, we omit the necessity operator from the formula. Instead, we merely tag the ~~logical~~ rule as a necessity. The implementation impact of the alethic necessity tag is that any attempted change that would cause the model of the business domain to violate the constraint must be dealt with in a way that ensures the constraint is still satisfied (e.g., reject the change, or take some compensatory action).

Typically, the only modal operator in an explicit ~~logical~~ rule formulation is $\Box$, and this is at the front of the ~~logical~~ rule formulation. This common case was covered earlier. If an alethic modal operator is placed elsewhere in the ~~logical~~ rule formulation, we first try to "normalize" it by moving the modal operator to the front, using transformation rules such as the modal negation rules ($\sim\Box p \equiv \Diamond\sim p$; $\sim\Diamond p \equiv \Box\sim p$) and/or the Barcan formulae and their converses ($\forall x\Box\Phi x \equiv \Box\forall x\Phi x$ *and* $\exists x\Diamond\Phi x \equiv \Diamond\exists x\Phi x$, i.e., $\Box$ and $\forall$ are commutative, as are $\Diamond$ and $\exists$).

> For example, the embedded formulation "$\forall x$:Person $\Box$ $\exists^{0..1}y$:Country *x* was born in *y*" (**For each** Person, **it is necessary that that** Person was born in **at most one** Country.) may be transformed into "$\Box\forall x$:Person $\exists^{0..1}y$:Country *x* was born in *y*" (**It is necessary that each** Person was born in **at most one** Country.).

We also allow use of the following equivalences: $\Box\Box p \equiv \Box p$; $\Diamond\Diamond p \equiv \Diamond p$; $\Box\Diamond\Box\Diamond p \equiv \Box\Diamond p$; $\Diamond\Box\Diamond\Box p \equiv \Diamond\Box p$. These hold in S4, but not in some modal logics, e.g., K or T [Girl2000, p. 35].

To make life interesting, SBVR also allows a single ~~logical~~ rule formulation to include multiple occurrences of modal operators, including the nesting of a modal operator within the scope of another modal operator. While this expressibility may be needed to capture some real business rules, it complicates attempts to provide a formal semantics.

In extremely rare cases, a formula for a static ~~logical~~ rule might contain an embedded alethic modality that cannot be eliminated by transformation. For such cases, we could retain the modal operator in the ~~logical~~ rule formulation and adopt the formal semantics of a particular modal logic. There are many normal modal logics to choose from (e.g., K, K4, KB, K5, DT, DB, D4, D5, T, Br, S4, S5) as well as many non-normal modal logics (e.g., C2, ED2, E2, S0.5, S2, S3). For a discussion of these logics, and their inter-relationships, see [Girl2000] (esp. pp. 48, 82). For SBVR, if we decide to retain the embedded alethic operator for such cases, we choose S4 for the formal semantics. The possibility of schema evolution along with changes to necessity constraints may seem to violate S4, where the accessibility relationship between possible worlds is transitive, but we resolve this by treating such evolution as a metametalevel concern. Alternatively, we may handle such very rare cases by moving the embedded alethic operators down to domain-level predicates (e.g., is necessary) in a similar fashion to the way we deal with embedded deontics (see later).

10.1.1.6 Static, Deontic Constraints

Issue 9368 Remove text

Constraint formulations may make use of the standard deontic modal operators (***O*** = it is obligatory that; ***P*** = it is permitted that) as well as ***F*** = it is forbidden that (defined as ~***P***, i.e., "It is not permitted that").

If the ~~logical~~ rule formulation includes exactly one deontic operator, ***O***, and this is at the front, then the ~~logical~~ rule may be formalized as ***O****p*, where *p* is a first-order formula that is tagged as obligatory (rather than necessary). For the purposes of this subclause, this tag is assigned only the following informal semantics: it ought to be the case that *p* (for all future states of the fact model, until the constraint is revoked or changed). The implementation impact is that it is possible to have a state in which the ~~logical~~ rule ~~'s condition~~ is violated (i.e., not satisfied), in which case some appropriate action (currently unspecified) ought to be taken to help reduce the chance of future violations. ~~A later submission to the OMG is intended to address logical rule enforcement, including the specification of appropriate actions in response to deontic rule "violations."~~

From a model-theoretic perspective, a model is an interpretation where each *non-deontic* formula evaluates to true, and the model is classified as a *permitted model* if the *p* in each deontic formula (of the form ***O****p*) evaluates to true, otherwise the model is a *forbidden model* (though it is still a model). Note that this approach removes any need to assign a truth value to expressions of the form ***O****p*.

For example, suppose the fact type Person is a husband of Person is declared to be many to many, but that each role of this fact type has a deontic uniqueness constraint to indicate that the fact type *ought* to be 1:1. The deontic constraint on the husband fact type role verbalizes as: **It is obligatory that each** Person is a husband of **at most one** Person. This formalizes as $\boldsymbol{O}\forall x$:Person $\exists^{0..1}y$:Person x is a husband of y, which may be captured by entering the ~~logical~~ rule body as $\forall x$:Person $\exists^{0..1}y$:Person x is a husband of y and tagging the ~~logical~~ rule body as deontic. The other deontic constraint (each wife should have at most one husband) may be handled in a similar way. A more detailed treatment of this example is included in Annex J.

Note that some formulae allowed by SBVR are illegal in some deontic logics (e.g., iterating modal operators such as ***OP****p* is forbidden in von Wright's deontic logic), and deontic logic itself is "rife with disagreements about what should be the case" [Girl2000, p. 173].

If a deontic modal operator is embedded later in the ~~logical~~ rule formulation, we first try to "normalize" the formula by moving the modal operator to the front, using transformation ~~logical~~ rules such as $p \supset \boldsymbol{O}q \;.\!\equiv\!.\; \boldsymbol{O}(p \supset q)$ or deontic counterparts to the Barcan formulae.

In some cases, a formula for a static ~~logical~~ rule might contain an embedded deontic modality that cannot be eliminated by transformation. In this case, we still allow the business user to express the ~~logical~~ rule at a high level using such embedded deontic operators, but *where possible* we transform the formula to a first-order formula without modalities by *replacing the modal operators by predicates at the business domain level*. These predicates (e.g., is forbidden) are treated like any other predicate in the domain, except that their names are reserved, and they are given some basic additional formal semantics to capture the deontic modal negation rules: it is not obligatory that ≡ it is permitted that it is not the case that ($\sim\boldsymbol{O}p \equiv \boldsymbol{P}\sim p$); it is not permitted that ≡ it is obligatory that it is not the case that ($\sim\boldsymbol{P}p \equiv \boldsymbol{O}\sim p$). For example, these ~~logical~~ rules entail an exclusion constraint between the predicates is forbidden and is permitted.

This latter approach may also be used as an alternative to tagging a ~~logical~~ rule body as deontic, thereby (where possible) moving deontic aspects out of the metamodel and into the business domain model.

For example, consider the following ~~logical~~ rule:

Car rentals ought not be issued to people who are barred drivers at the time the rental was issued.

This deontic constraint may be captured by the following textual constraint on the domain fact type CarRental is forbidden:

CarRental is forbidden **if**

CarRental was issued at Time **and**

CarRental was issued to Person **and**

Person is a barred driver at Time.

The fact type Person is a barred driver at Time is derived from other fact types (Person was barred at Time, Person was unbarred at Time) using the derivation rule:

Person is a barred driver at Time_1 **iff**

Person was barred at **a** Time_2 <= Time_1 **and**

Person was **not** unbarred at **a** Time_3 **between** Time_2 **and** Time_1.

The deontic constraint may be formalized by the first-order formula: $\forall x$:CarRental $\forall y$:Person $\forall t$:Time [(x was issued at t & x was issued to y & y is a barred driver at t) $\supset$ x is forbidden]. This schema allows for the possible existence of forbidden car rentals; if desired, some fact types could be added to describe actions (e.g., sending messages) to be taken in reaction to such an event.

As a second example, consider the following deontic rule:

It is forbidden that more than three people are on the EU-Rent Board.

Suppose the underlying schema includes the fact type: Person is on Board. This may be used to define the derived fact type Board has NrMembers using the derivation rule: nrMembers **of** Board = **count each** Person **who** is on Board. Objectify this derived fact type as BoardHavingSize, and then add the fact type BoardHavingSize is forbidden. The deontic constraint may now be captured by the following textual constraint on the derived fact type:

BoardHavingSize is forbidden **if**

BoardHavingSize is of **a** Board

that has BoardName ‘EU-Rent Board’

and has NrMembers > 3.

As a third example, our earlier schema for current marriage may be recast by.objectifying the fact type Person is a husband of Person as CurrentMarriage, and recognizing the link fact types Person is a husband in CurrentMarriage and Person is a wife in CurrentMarriage. The deontic constraints may now be formulated as textual constraints on the fact type CurrentMarriage is forbidden as follows:

CurrentMarriage is forbidden **if**

a Person_1 **who** is a husband in CurrentMarriage

is a husband of **more than one** Person_2.

CurrentMarriage is forbidden **if**

a Person_1 **who** is a wife in CurrentMarriage

is a wife of **more than one** Person_2.

Extended treatments of the examples above are provided in Annex J.

The approach to objectification described here works for those cases where a fact (proposition taken to be true) is being objectified (which covers the usual cases of nominalization, including the EU-Rent Board and current marriage examples discussed earlier), but it does not handle cases where no factual claim is being made of the proposition.

SBVR is intended to cater for ~~logical~~ rules that embed possibly non-factual propositions. However, there does not appear to be any simple solution to providing explicit, formal semantics for such ~~logical~~ rules.

As a nasty example, consider the following ~~logical~~ business rule:

It is not permitted that some department adopts a ~~logical~~ rule that says it is obligatory that each employee of that department is male.

This example includes the mention (rather than use) of an open proposition in the scope of an embedded deontic operator. One possible, though weak, solution is to rely on reserved domain predicates to carry much of the semantics implicitly. For example, suppose the schema includes the following fact types: Person is male; Person works for Department; Department adopts Logic Rule. Objectify Department adopts ~~Logical~~ Rule as ~~Logic~~RuleAdoption, and add the following fact types: ~~Logic~~RuleAdoption is forbidden; ~~Logic~~Rule obligates the actualization of PossibleAllMaleState; PossibleAllMaleState is actual. This uses the special predicates "obligates the actualization of" and "is actual," as well as a type of individual "PossibleAllMaleState" which includes all conceivable all-male-states of departments, whether actual or not. The derived fact type PossibleAllMaleState is actual may be defined using the derivation rule:

PossibleAllMaleState is actual **iff**

PossibleAllMaleState is of **a** Department **and**

each Person **who** works for **that** Department is male.

i.e., $\forall x$:PossibleAllMaleState [x is actual $\equiv \exists y$:Department (x is of y & $\forall z$:Person (z works for y $\supset$ z is male))].
The deontic constraint may now be captured by the following textual constraint on the fact type ~~Logic~~RuleAdoption is forbidden:

~~Logic~~RuleAdoption is forbidden **if**

~~Logic~~RuleAdoption is by **a** Department

and is of **a** ~~Logic~~Rule

that obligates the actualization of **a** PossibleAllMaleState

that is of **the same** Department.

i.e., $\forall x$:~~Logic~~RuleAdoption $\forall y$:Department $\forall z$:~~Logic~~Rule $\forall w$:PossibleAllMaleState [(x is by y & x is of z & z obligates the actualization of w & w is of y) $\supset$ x is forbidden]

The formalization of the deontic constraint works, because the relevant instance of PossibleAllMaleState exists, regardless of whether or not the relevant depart actually is all male. The "obligates the actualization of" and "is actual" predicates embed a lot of semantics, which is left implicit. While the connection between these predicates is left informal, the derivation rule for PossibleAllMaleState is actual provides enough semantics to enable human readers to understand the intent. An extended treatment of this example is provided in Annex J.

Alternatively, we could capture the structure of the ~~logical~~ rule using the current semantic formulation machinery, and then adopt one of two extremes: (1) treat the ~~logical~~ rule overall as an uninterpreted sentence, or informal comment, for which humans are to provide the semantics; (2) translate the semantic formulation directly into higher-order logic, which permits logical formulations (which connote propositions) to be predicated over. The complexity and implementation overhead of option (2) would seem to be very substantial.

We could try to push such cases down to first-order logic by providing the equivalent of the semantic formulation machinery as a predefined package that may be imported into a domain model, and then identifying propositions by means of a structured logical formulation. But that seems a fudge, because in order to assign formal semantics to such expressions, we must effectively adopt the higher-order logic proposal mentioned in the previous paragraph.

Pat Hayes has indicated his intent to add support for reification as an extension to Common Logic at some future date. This support is intended to cater for objectification of propositions that are already being asserted as facts (i.e., propositions being used), as well as propositions for which no factual claim is made (i.e., propositions being mentioned). When available, his treatment for the latter case may offer a better solution for the problem under consideration. His intent is to allow quantification and predication over propositions (or expressions that declare propositions), regardless of whether truth claims are being asserted of those propositions, while still retaining a first-order approach. We might be able to adopt whatever he proposes in this regard to provide a formal semantics for such problematic ~~logical~~ rules.

10.1.1.7 Derivation Rules

Issue 9368 Remove text

The SBVR approach supports ~~logical~~ rules for deriving types of individuals (subtype definitions) or fact types using either 'if-and-only-if' (equivalence) formulations for full derivation, or 'if' ~~rules~~ for partial derivation. A subtype may be fully derived (defined in terms of fact type roles played by its supertype), asserted (without a derivation rule), or partly derived.

Here is one simple example of each kind of derivation rule, stated first using a high-level textual language, as described earlier, and then recast as a predicate logic formula. The transformation from a semantic formulation structure in a high level language into predicate logic is straightforward.

Derivation rule for fully derived subtype:

Each Australian **is a** Person **who** was born in Country 'AU.'

$\forall x$ [Australian x $\equiv$ (Person x & $\exists y$:Country $\exists z$:CountryCode (x was born in y & y has z & z = 'AU'))]

Derivation rule for partly derived subtype:

Person_1 **is a** Grandparent **if** Person_1 is a parent of **some** Person_2 who is a parent of **some** Person_3.

$\forall x$:Person [Grandparent x $\subset$ $\exists y$:Person $\exists z$:Person (x is a parent of y & y is a parent of z)]

Derivation rule for fully derived fact type:

Person_1 is an uncle of Person_2 **iff** Person_1 is a brother of **some** Person_3 **who** is a parent of Person_2.

$\forall x$:Person $\forall y$:Person [x is an uncle of y $\equiv$ $\exists z$:Person (x is a brother of z & z is a parent of y)]

Derivation rule for partly derived fact type:

If a Patient smokes **then that** Patient is cancer-prone.

$\forall x$:Patient (smokes x $\supset$ cancer-prone x)

10.1.1.8 Dynamic Constraints

Dynamic constraints apply restrictions on possible transitions between business states. The constraint may simply compare one state to the next.

Salaries should never decrease.

Alternatively, the constraint may compare states separated by a given period.

Invoices ought to be paid within 30 days of being issued.

The invoice ~~logical~~ rule might be formally expressed in a high level rules language thus, assuming the fact types Invoice was issued on Date and Invoice is paid on Date are included in the conceptual schema:

For each Invoice, **if that** Invoice was issued on Date_1
then it is obligatory that

that Invoice is paid on Date_2 **where** Date_2 <= Date_1 + 30 days.

This might now be normalized to the following formulation, moving the deontic operator to the front:

It is obligatory that each Invoice **that** was issued on Date_1 is paid on Date_2
where Date_2 <= Date_1 + 30 days.

There are two issues here. First, what ~~logical~~ transformation rules did we rely on to license the transformation of the ~~logical~~ rule? It would seem that we require an equivalence rule such as $p \supset \boldsymbol{O}q \;.\!\equiv\!.\; \boldsymbol{O}(p \supset q)$. While this formula is actually illegal in some deontic logics, it does seem intuitively acceptable. At any rate, the preliminary transformation work in normalizing a ~~logical~~ rule formulation might involve more than just the Barcan equivalences or their deontic counterparts. In principle, this issue might be ignored for interoperability purposes, so long as the business domain expert is able to confirm that the final, normalized formulation (perhaps produced manually by the business rules modeler) agrees with their intended semantics; it is only the final, normalized formulation that is used for exchange with other software tools.

The second issue concerns the dynamic nature of the ~~logical~~ rule. While it is obvious how one may actually implement this logical rule in a database system, capturing the formal semantics in an appropriate logic (e.g., a temporal or dynamic logic) is a harder task. One possibility is to provide a temporal package that may be imported into a domain model, in order to provide a first-order logic solution. Another possibility is to adopt a temporal modal logic (e.g., treat a possible world as a sequence of accessible states of the fact model). It may well be reasonable to defer decisions on formal semantics for dynamic rules to a later version of the SBVR standard.

10.1.1.9 Higher-order Logic

Currently, SBVR allows users to either stay with first-order logic, or adopt higher-order logic restricted to Henkin semantics (e.g., for dealing with categorization types). In general, standard higher-order logic allows quantification over uncountably many possible predicates (or functions). If D = the domain of individuals, then the range of any unary predicate variable R is the entire power set P(D) (i.e., the set of all subsets of D), the range of any binary predicate variable is the Cartesian product P(D) × P(D), and so on for higher arity predicates. If D includes a denumerable (countable infinite, i.e., $|D| = \aleph_0$) set, such as the natural numbers, then P(D) is uncountably infinite. In contrast, Henkin semantics restricts quantifiers to range over only individuals and those predicates (or functions) that are specified in the universe of discourse (a.k.a. business domain), where the n-ary predicates/functions (n > 0) range over a fixed set of n-ary relations/operations. By restricting the ranges of predicate and function variables, the Henkin interpretation retains certain desirable first-order properties (e.g., completeness, compactness, and the Skolem-Löwenheim theorems) that are lost in the standard interpretation of higher-order logic.

Common Logic adopts the Henkin restriction on quantifier ranges, but does not adopt the Axiom of Comprehension, which states that for each property there exists a set of elements having that property, i.e., for any formula $\varphi(x)$ where x (possibly a vector) is free in φ, $\exists A \forall x[x \in A \equiv \varphi(x)]$. The intent of the Comprehension axiom (to ensure that every formula specifies a set) may also be achieved by using lambda abstraction to name the set, e.g., $\lambda x.\varphi(x)$, which is equivalent to the set comprehension $\{x|\ \varphi(x)\}$. The Axiom of Comprehension leads to Russell's paradox (substituting $x \notin x$ for $\varphi(x)$ generates a contradiction since $\{x|\ x \notin x\}$ is simultaneously a member of itself and not a member of itself). The paradox may be avoided either by rejecting the comprehension axiom (e.g., replacing it by the weaker axiom of separation, as in Zermelo-Fraenkel set theory) or by restricting the language so that formulae such as $x \notin x$ are illegal (as in Russell's type theory, where a set may belong only to a set of higher order).

Here we use ~~uses~~ set comprehensions (in a restricted sense) to define projections on schema path expressions, as a way to specify result sets.

For example, given the fact type Employee(EmpNr) works for Company(Name), the query "Who works for Microsoft?" corresponds to the following set comprehension:

{*x*:Employee | ∃ *y*:Company; *z*:CompanyName (*x* works for *y* & *y* has *z* & *z* = 'Microsoft')}

The formal semantics of such conceptual queries is based on that of the Conquer language, which provides a sugared version of sorted finitary first-order logic with set comprehension [Anto2001].

The use here of set comprehension is quite restricted. Any expression we use to define a set must ultimately be expressible only in terms of some basic logical operators (e.g., &) as well as predefined ground fact types which must be either elementary or existential. Hence we adopt a limited version of the axiom of comprehension. Common Logic is open to extensions that adopt restricted versions of the comprehension axiom. To avoid Russell's paradox, we treat formulae such as $x \notin x$ as illegal. The "is an instance of" predicate caters for set membership, but is constrained to be irreflexive, and the formation rules do not permit expressions of the form $x \in x$ – in other words, we cannot make statements involving self-membership. We do not adopt a type theory such as Russell's type theory, where each set may belong only to a set of a higher type.

The decision on whether to use higher-order types mainly impacts the following three aspects of fact modeling: categorization schemes, un-normalized structures, and crossing levels/metalevels within the same model. In [Halp2004], some ways are suggested to avoid higher-order types, by treating types as intensional individuals whose instances may sometimes be in 1:1 correspondence (but not identical) to subtypes, by requiring subtype definitions to be informative, by remodeling (including demotion of metadata to data), and by treating types as individuals in separate models. For further discussion, see [Halp2004].

Acknowledgement: We gratefully acknowledge the assistance of Pat Hayes (http://www.ihmc.us/users/user.php?UserID=phayes) in addressing some of the logical semantics topics in this document.

10.1.2 Formal Logic & Mathematics in General

Formal Logic and Mathematics Vocabulary

Language: English

Issue 9721 Add text

---

acceptable world

Definition: any state (situation) of some given universe of discourse (domain) that is implicitly characterized, by someone with legal authority over that domain, as consistent with some set of goals of that authority pursued by exercise of that authority

Issue 9475/9945 Change text

alethic modality

Source: CDP

Definition: Historically, any of the five central ways or modes in which a given proposition might be true or false: necessity (and non-necessity), possibility (and impossibility), and contingency.

Note: (1) Although these “modes” have historically been thought of as ways in which a proposition might be true, we think of them as ways in which one might think of the truth of a proposition: e.g., that a proposition be qualified with the alethic modality “necessity” does not imply it is a fact, but only signifies that the semantic community is considering it (takes it to be) necessarily true. For some issues arising from the former approach, cf. CDP, s.v. *intensional logic*. For a thorough critique of it, see PEIL. The four “modal negation equivalences” (MLP, p. 3), such as $\Box p \equiv \sim\Diamond\sim p$, still hold under the latter approach (cf. LEVS, p. 135), which is the more useful one in the fields of linguistic semantics and linguistic pragmatics.

Note: (2) The four alethic modalities which we consider most basic, and to which the four “modal negation equivalences” (MLP, p. 3) apply, are necessity, possibility, and their respective negations (non-necessity and impossibility). We also define a fifth modality, contingency for the idea “neither impossible nor necessary.” (CDP)

Note: (3) Alethic modal logic differs from deontic modal logic in that the former deals with people’s estimate(s) of the possible truth of some proposition, whereas deontic modal logic deals with people’s estimate(s) of the social desirability of some particular party’s making some proposition true.

antecedent

Source: adapted from GFOL

Definition: The wff in [or more specifically, the proposition-wff in or else the proposition denoted by] the if-clause of an implication.

Note: Interpolation ours. Otherwise the definition is from GFOL.

argument

Source: GFOL

Definition: a [logical-] subject-term for a predicate.

Note: Interpolation in square brackets ours. By “logical subject” we mean an object playing a role (i.e., an object filling an object hole) in a logical predicate. Thus there may be one or more logical-subject-terms in a logical predicate.

arity

Source: IMRD (pp. 10, 64)

Definition: A logical predicate’s number of roles (i.e., of object holes).

Note: A function may be thought of as a relation; accordingly, we treat a function as a logical predicate. MATH defines arity of a function thus: “The number of arguments taken by something, usually applied to functions: an *n*-ary function is one with an arity of *n*, i.e., it takes *n* arguments. Unary is a synonym for 1-ary, and binary is a synonym for 2-ary.”

atomic formula

Source: GFOL ["atom"]

Definition: In predicate logic, a wff without quantifiers or connectives.

Note: (1) This definition is from the cited source s.v. atom, which we deem a synonym.

Note: (2) LSO says of atomic formula: "The simplest sort of wff of a formal language; an atomic formula of the language of predicate logic is a predicate letter followed by zero or more name letters." Yet it can also be a propositional variable or a propositional constant, depending on context.

consequent

Source: GFOL

Definition: The wff in [or more specifically, the proposition-wff in or else the proposition denoted by] the then-clause of an implication.

Note: Interpolation ours.

contingency

Definition: alethic modality that is the conjunction of possibility and non-necessity

Note: Contingency ("it is possible but not necessary that *p*") is the modal equivalent of "it is neither impossible nor necessary that *p*": $(\Diamond p \,\&\, \sim \Box p) \equiv \sim (\sim \Diamond p \vee \Box p)$.

deontic modality

Source: CDP ["deontic operator"]; LEVS (pp. 276-77); LSO (p. 302); MLP (pp. 170-76)

Definition: Any of the five central ways or modes in which one might think of the social desirability of a certain other person(s)'s making true some proposition, that is, the social desirability that the act(s) be performed, by a certain other person(s), that would make the proposition true; viz., obligation (and its negation, non-obligation), permission (and its negation, nonpermission (forbidden/prohibition)), and optionality.

Note: (1) The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

Note: (2) Alethic modal logic differs from deontic modal logic in that the former deals with people's estimate(s) of the possible truth of some proposition, whereas deontic modal logic deals with people's estimate(s) of the social desirability of some particular party's making some proposition true.

Note: (3) The four deontic modalities that we consider most basic, and to which the four "modal negation equivalences" apply, are obligation, permission, and their respective negations (non-obligation and prohibition). We also define a fifth modality, optionality, for the idea "neither prohibited nor obligatory."

domain

Source: GFOL

Definition: Of an interpretation of a formal language of predicate logic, the set of objects that may serve as the assigned referents of the constants of the language, the arguments of functions, and the arguments of predicates.

domain grammar

Source: META (p. 4); HALT89 (sec. 3.2); IMRD (pp. 27-30)

Definition: The formation rules determining what is a wff in a given domain-specific formal language.

Note: Another term for that which is called in ORM "conceptual schema." The definition given above is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

first-order instance

Source: GFOL

Definition: The objects or elements taken as the [logical] subjects of the predicates of first-order predicate logic.

Definition: [CLARIFIED DEFINITION] object or element taken as a logical subject of a predicate of first order logic.

Note: And the distinguishing characteristic of "first-order" predicate logic, in turn, is the additional restriction, re the formation of wffs, that subjects of predicates cannot themselves be types or predicates, but rather only individuals (or individual-constants, individual-variables, or function-expressions). See first-order type.

first-order type

Source: LSO (pp. 280-84) [and "type system"]; META (p. 140); TTGG (p. 5)

Definition: A type whose extension includes no types or predicates, only first order instances, in accordance with the grammatical restrictions in first-order predicate logic.

Note: The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

formal model

Source: based on GFOL ["model"]; META (pp. 5,6, 148-49)

Definition: An *interpretation* supplies semantics (referents) for a given formal language, in relation to some domain or universe. It specifies referents for the nonlogical symbols occurring in the formal language. A *formal model* of a given wff or set of wffs in a formal language is an interpretation of the language for which the wffs are considered true.

implication

Source: GFOL

Definition: expression of the form, "if A, then B," when A and B stand for wffs or propositions. The wff in the if-clause is called the antecedent (also the implicans and protasis). The wff in the then-clause is called the consequent (also the implicate and apodosis). Also called a conditional, or a conditional statement.

Note: In SBVR we treat "implication" as if it is "material implication" (i.e., 'p ➞ q' is equivalent to '~p v q').

impossibility

Definition: alethic modality that is the negation of possibility

Note:	A *derived modal operator* for 'impossibility' may be used in the surface syntax, but it is translated into the basic modal operator for 'possibility' plus negation (~) (i.e., "It is impossible that *p*" is defined as "It is not possible that p": ~◊*p*).
Note:	Impossibility ("it is impossible that *p*") is the modal equivalent of "it is necessary that not *p*": ~◊*p* ≡ □~*p*.

integer

Source:	GFOL ["integers"]
	The natural numbers supplemented by their negative counterparts. The set {...-3, -2, -1, 0, 1, 2, 3...}.

logical variable

Source:	GFOL
Definition:	A symbol whose referent varies or is unknown. A place-holder, as opposed to an abbreviation or name (a constant).
Note:	This definition is from the cited source s.v. variable, which we deem a synonym.

member

Source:	DEAN (p. 6); GFOL ["membership"]
Definition:	An element belonging to a set.
Note:	The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

modal logic

Source:	SEP
Definition:	Narrowly construed, modal logic studies reasoning that involves the use of the expressions 'necessarily' and 'possibly.' However, the term 'modal logic' is used more broadly to cover a family of logics with similar rules and a variety of different symbols.

necessity

Source:	CDP
Definition:	A modal property that qualifies an assertion of a whole proposition just when it is not considered possible that the proposition is false.
Note:	The definition given is not quoted directly from any source. Rather, we have based our definition on passages mainly in the above-cited source. See also alethic modality.
Note:	Necessity ("it is necessary that *p*") is the modal equivalent of "it is not possible that not *p*": □*p* ≡ ~◊~*p*.
Note:	The following *modal negation rules* apply: "it is not necessary that *p*" ≡ "it is possible that not *p*": ~□*p* ≡ ◊~*p*. See non-necessity.

non-necessity

Definition:	alethic modality that is the negation of necessity.
Note:	Non-necessity ("it is not necessary that *p*") is the modal equivalent of "it is possible that not *p*": ~□*p* ≡ ◊~*p*.

non-obligation

Definition: deontic modality that is the negation of obligation.

Note: Non-obligation ("it is not obligatory that *p*") is the modal equivalent of "it is permitted that not *p*": $\sim\boldsymbol{O}p \equiv \boldsymbol{P}{\sim}p$.

obligation

Source: CDP ["deontic logic"]; MLP (pp. 170-76)

Definition: One of the four main deontic modalities, which qualifies as socially obligatory the making true a certain proposition (i.e., the doing a certain act) by a certain party or parties.

Note: The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

Note: Obligation ("it is obligatory that p") is the modal equivalent of "it is not permitted that not p": $\boldsymbol{O}p \equiv \sim\boldsymbol{P}{\sim}p$

Note: The following modal negation rules apply:
"it is not obligatory that p" ≡ "it is permitted that not p": $\sim\boldsymbol{O}p \equiv \boldsymbol{P}{\sim}p$. See non-obligation.

optionality

Definition: deontic modality that is the conjunction of permission and non-obligation

Note: Optionality ("it is permitted but not obligatory that *p*") is the modal equivalent of "it is neither prohibited nor obligatory that *p*": $(\boldsymbol{P}p \,\&\, \sim\boldsymbol{O}p \equiv \sim(\sim\boldsymbol{P}p \vee \boldsymbol{O}p)$.

permission

Source: CDP ["deontic logic"]; MLP (pp. 170-76)

Definition: One of the four main deontic modalities, which qualifies as socially permissible the making true a certain proposition (i.e., the doing a certain act) by a certain party or parties.

Note: The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

Note: Permission (it is permitted that *p*") is the modal equivalent of "it is not obligatory that not *p*": $\boldsymbol{P}p \equiv \sim\boldsymbol{O}{\sim}p$.

Note: The following model negation rules apply:
"it is not permitted that *p*" ≡ "it is obligatory that not *p*": $\sim\boldsymbol{P}p \equiv \boldsymbol{O}{\sim}p$. See prohibition.

population

Source: IMRD (p. 164)

Definition: The extension of a type (whether object type, fact type, or role) for a given state of the business domain.

possibility

Source: CDP

Definition: A modal property that qualifies an assertion of a whole proposition just when it is considered possible that the proposition is true.

Note: The definition given is not quoted directly from any source. Rather, we have based our definition on passages mainly in the above-cited source. See also alethic modality.

Note: Possibility ("it is possible that *p*") is the modal equivalent of "it is not necessary that not *p*": $\Diamond p \equiv \sim\Box\sim p$.

Note: The following *modal negation rules* apply:
"it is not possible that *p*" ≡ "it is necessary that not *p*": $\sim\Diamond p \equiv \Box\sim p$. See impossibility.

Issue 9721 Add text

possible world

Definition: any state (situation) of some given universe of discourse (domain) that is implicitly characterized, by an accepted expert on that domain, as logically consistent with some set of laws seen by that expert as applying to that domain

Note: "Possible world" means "logically possible world," and not "physically possible world." Included within the sense of "possible world" is any "possible situation;" therefore, the notion includes the "possible states" of any given set of objects of interest - which set is commonly called the "Universe of Discourse" (or "UoD"), a.k.a. the "domain" (or "business domain"). Thus, in the context of a static constraint declared for a given business domain, a "possible world" would correspond to (but not be identical to) a state of the domain's fact model that could exist at some point in time.

predicate

Source: GFOL

Definition: Intuitively, whatever is said of the subject[s] of a sentence. A function from individuals (or a sequence of individuals) to truth-values.

Note: Interpolation in square brackets ours. A predicate is distinguished from others by sentence structure, not by proposition/meaning (see IMRD, pp. 63-66). Propositions or meanings distinguish fact types, each of which may have 1 or more predicates.

Issue 9475/9945 Change text

prohibition

Source: CDP ["deontic logic"]; MLP (pp. 170-76)

Definition: One of the four main deontic modalities nonpermissibility, which qualifies as socially not permissible the making true a certain proposition (i.e., the doing a certain act) by a certain party or parties.

Definition: deontic modality that is the negation of permission

Note: See also permission. The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

Note: A *derived modal operator* for 'prohibition' may be used in the surface syntax, but it is translated into the basic modal operator for 'permission' plus negation (~). (i.e., "It is prohibited that *p*" is defined as "It is not permitted that *p*": ~***P****p).*

Note: A *derived modal operator* for 'forbidden' may be used in the surface syntax, but it is translated into the basic modal operator for 'permission' plus negation (~). (i.e., "It is forbidden that *p*" (***F****p*) is defined as "It is not permitted that *p*": ~***P****p).*

Note: Prohibition ("it is prohibited that *p*") is the modal equivalent of "it is obligatory that not *p*": $\sim\boldsymbol{P}p \equiv \boldsymbol{O}\sim p$.

proposition

Source: DL (p. 4)

Definition: That which is asserted when a sentence is uttered or inscribed.

Note: Generally understood as "the meaning of" a declarative sentence. GFOL defines it thus: "In logic generally (for some), the meaning of a sentence that is invariant through all the paraphrases and translations of the sentence."

propositional operator

Source: PLTS

Definition: An operator (or connective) joins … statements [i.e., propositions or proposition-wffs] into compounds…. Connectives include conjunction, disjunction, implication and equivalence. Negation is the only operator that is not a connective; it affects single statements [i.e., propositions or proposition-wffs] only, and does not join statements [i.e., propositions or proposition-wffs] into compounds.

Note: By "proposition-wff" we mean a proposition-constant or proposition-variable, or a predicate supplied with arguments so as to yield a proposition.

quantifier

Source: GFOL

Definition: In predicate logic, a symbol telling us … how many objects (in the domain) [instantiate] the predicate…. The quantifier applies to, or binds, variables which stand as the arguments of predicates. In first-order logic these variables must range over individuals; in higher-order logics they may range over predicates.

Note: Interpolation in square brackets ours.

restricted higher-order instance

Source: HALT2004 (pp. 2-4, 7); MEN97 (pp. 378-80)

Definition: instance of a restricted higher-order type.

Note: The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

restricted higher-order type

Source: HALT2004 (pp. 2-4, 7, 8); MEN97 (pp. 378-80)

Definition: A *higher-order type* includes an instance that is itself a type. For SBVR, we *restrict higher-order types* to Henkin semantics, limiting the range of predicates/functions over which we may quantify to a fixed set, rather than allowing full range over power-sets. This restriction retains useful properties of first-order logic (e.g., completeness).

Note: The definition given is not quoted directly from any source, since we have not found the term defined as such anywhere. Rather, we have based our definition on passages mainly in the above-cited sources.

set

Source: GFOL

Definition: Intuitively, a collection of elements (called members). In a set, the order of members is irrelevant, and repetition of members is [also irrelevant]. The intuitive notion of a set leads to paradoxes, and there is considerable mathematical and philosophical disagreement on how best to refine the intuitive notion.

Note: Interpolation in square brackets ours.

state of affairs

Source: CDP

Definition: A possibility, actuality or impossibility of the kind expressed by a nominalization of a declarative sentence. E.g., “This die comes up six” may be nominalized by “that this die comes up six” or “this die’s coming up six.” The resulting nominalizations might be interpreted as naming corresponding propositions or states of affairs.

subset

Source: GFOL

Definition: set all of whose members belong to a second set (a superset of the subset).

type

Source: adapted from HALT2004 (p. 8); cf. TTGG (p. 84)

Definition: named set of possible instances, where for any given state of the business domain, exactly one subset of the type is the population of the type in that state.

Note: At any given time, the population of a type is the set of instances of that type that exist in the business domain (i.e., that are referenced within facts that are known and are of interest to the business) at that time. It follows that if two types are equal, then for each state of the business domain they must have the same population.

Note: “Possible instances” here means “instances which are considered part of the type’s population, for some state of the business domain.”

Note: Because it is a formal object that behaves quite differently in first-order predicate logic than in second-order predicate logic (and differently still in third order, and so on), the definition of “type” proves to be anaphoric, having a different denotation depending on whether, in the situation where used, the intended formalization is first-order, second-order, or other-order. In our definitions of first-order type and restricted higher order type, at least some of this indefiniteness is removed (by the specifying of either first-order logic or restricted higher-order logic).

unbound variable

Source: GFOL

Definition: free variable [which, in GFOL, is defined thus:] In predicate logic, an individual variable at least one of whose occurrences in a wff does not lie within the scope of a quantifier on the same letter.

Issue 9721 Add text

Universe of Discourse

Definition: set of objects of interest, including their states, relationships, and situations and forming the context of a given discussion

wff

Source: GFOL

Definition: Acronym of “well-formed formula.” A string of symbols, each from the alphabet of a formal language, that conforms to the grammar of the formal language. In predicate logic, a closed wff is a wff with no free occurrences of any variable; either it has constants in place of variables, or its variables are bound, or both. Also called a sentence. An open wff is a wff with at least one free occurrence of a variable.

world

Source: CSILL

Definition: A universe, whether real, imaginary, or hypothetical.

Note: From CSILL: The truth-conditional approach to meaning allows model theory to be extended to the study of natural languages. Sentences and their parts are mapped on to elements of a model, which represents the truth-conditions for the sentences. In possible world semantics, models are not restricted to domains of real entities but include possible objects; that is, model theory can provide truth-conditions in terms of possible worlds, thus allowing meaningful expressions without requiring ontological commitment.

10.2 Formal Logic Interpretation Placed on SBVR Terms

Issue 9959 Remove all tables and text in Clause 10.2 and 10.3 and replace with the following.

This Clause specifies how the SBVR concepts in the table below, as defined in Clauses 8, 9, 11 and 12, are to be interpreted in terms of formal logic as defined in ISO 24707 “Information technology - Common Logic (CL) - A framework for a family of logic-based languages.” Equivalent concepts in OWL are also shown in the table where possible.

The ISO 24707 interpretation of SBVR concepts shown in the table below implements the formal logic grounding principles set forth in Clause 10.1

NOTE: The cells that are empty will be specified in a future revision of this specification.

NOTE: All SBVR Terms are "meanings" where all CL Terms are “representations of meanings.” Therefore there is a one-to-many relationship between SBVR Terms as meanings and CL Terms as representations of meanings; i.e., there can be multiple CL representations of one SBVR meaning.

SBVR Term	ISO CL Term (or equivalent expression)	OWL Term (or equivalent expression)	Comment
BASICS - Foundation			
fact	sentence with an interpretation 'taken to be' true NOTE: The mapping is many (sentences) to one (meaning)	OWL statement (*s, p, o*) interpreted as being true; individual	
fact type (3+ary) + (unary fact type)	unary predicate defining the type for a functional term or atomic sentence	---	
fact type (binary fact type)	unary predicate defining the type for a functional term or atomic sentence that has exactly two arguments	Class description defining RDF property or OWL object property (note: may only apply to OWL Full)	Need 2 RDF/ OWL properties related by inverse of = one binary fact type
fact type ***has*** fact type role	argument role in functional term or atomic sentence	---	
fact type ***has*** fact type role (binary fact type)	argument role in functional term or atomic sentence that has exactly two arguments	the range of an rdf:Property or owl:ObjectProperty; alternatively, may be specified using a restriction on the property in OWL	
fact type role	unary predicate defining the role of a name/term that is an argument	RDF/OWL subject or object	
fact type role ***ranges over*** object type (role ranges over object type)	term over which argument ranges	value restriction on property	
fundamental concept			
individual concept	name	individual	
object type	unary predicate	class	
proposition	sentence with an interpretation	OWL statement (*s, p, o*); individual	
proposition ***is false***	sentence with an interpretation = false	OWL statement (*s, p, o*) interpreted as being false; individual	
proposition ***is true***	sentence with an interpretation = true	OWL statement (*s, p, o*) interpreted as being true; individual	
reference scheme	approximately term		
reference scheme ***extensionally uses*** role			

reference scheme *is for* concept			
reference scheme *simply uses* role			
reference scheme *uses* characteristic			
situational role	unary predicate defining the role of a name/term that is an argument	RDF/OWL subject or object	
situational role *ranges over* fundamental concept (role ranges over object type)	term over which argument ranges	value restriction on property	
BASICS - Extension in Model			
	NOTE: There are two kinds of extensions in SBVR: 1. Real things that never appear in an SBVR Model themselves 2. Model extensions: a. Individual concepts as model instances of object types (fundamental concepts only) b. facts as model instances of fact types		
concept$_1$ *is coextensive with* concept$_2$ (fact type)	(forall (p1 p2) (if (and (binary fact type p1) (binary fact type p2)) (iff (is coextensive with p1 p2) (forall (x y) (iff (p1 x y) (p2 x y))))))	owl:equivalentProperty	
concept$_1$ *is coextensive with* concept$_2$ (noun concept)	(forall (c1 c2) (if (and (noun concept c1) (noun concept c2)) (iff (is coextensive with c1 c2) (forall (x) (iff (c1 x) (c2 x))))))	owl:equivalentClass	
concept *has* extension (verb concept / fact type)	"sentence type" has extension		
concept *has* extension (noun concept)	((forall (x)(iff (concept x) (or (= aaa-1 x) ... (= aaa-n x))))	enumeration of a class (OWL one Of)	
extension	extension	class	
proposition *corresponds to* state of affairs	approximately sentence denotation		
concept *classifies* thing (concept has instance)	atom (concept thing)	can be specified via an rdf:type statement (*i.e.*, thing rdf:type concept.)	
set	set		
BASICS - Intension: Characteristic			

characteristic	(see unary fact type)	(see unary fact type	(see unary fact type)
characteristic *is essential to* concept			
characteristic type			
concept *has* implied characteristic			
concept *has* necessary characteristic			
concept *incorporates* characteristic	sentence (forall (u)(implies(characteristic u)(concept u)))	rdfs:subClassOf	
delimiting characteristic			
essential characteristic			
implied characteristic			
intension	intension		
necessary characteristic			
BASICS - Intension: Categorization			
categorization scheme			
categorization type			
category			
concept type	unary predicate	class	
$concept_1$ *specializes* $concept_2$ (binary fact type)	(forall (p1 p2) (if (and (binary fact type p1) (binary fact type p2)) (iff (specializes p1 p2) ((forall (x y) (if (p1 x y) (p2 x y)))))))	rdfs:subPropertyOf + disjoint	
$concept_1$ *specializes* $concept_2$ (noun concept)	(forall (c1 c2) (if (specializes c1 c2) (forall (x) (if (c1 x) (c2 x))))) (forall (c1 c2) (if (and (specializes c1 c2) (specializes c2 c3)) (specializes c1 c3)))	rdfs:subClassOf + disjoint	One way from SBVR to CL
more general concept			
segmentation			
BASICS - Modal Logic			
element of guidance *authorizes* state of affairs			

element of guidance *obligates* state of affairs			
element of guidance *prohibits* state of affairs			
operative business rule			
proposition *is necessarily true*			
proposition *is obligated to be true*			
proposition *is permitted to be true*			
proposition *is possibly true*			
rule			
structural rule			
BASICS - Misc.			
quantity$_1$ *is less than* quantity$_2$	functional term with operator “is less than” and arguments quantity1 and quantity2		
integer	atom (integer x)	xsd:integer	There are no explicitly defined types in CL; there is specific set of XML schema datatypes available for use iwth RDF and OWL
nonnegative integer	atom (nonnegative integer x)	xsd:nonNegativeInteger	
number	atom (number x)		
positive integer	atom (positive integer x)	xsd:positiveInteger	
quantity			
SEMANTIC FORMULATIONS			
aggregation formulation			
antecedent			
at-least-n-quantification		restriction, owl:minCardinality n	

at-least-n-quantification *has* minimum cardinality			
at-most-n-quantification		restriction, owl:maxCardinality n	
at-most-n-quantification *has* maximum cardinality			
at-most-one-quantification		restriction, owl:maxCardinality 1	
atomic formulation	atomic sentence or atom	if unary - rdf:type if binary - rdf;triple nothing not 3+	
atomic formulation *has* role binding			
atomic formulation *is based on* fact type			
auxiliary variable			
bag projection			
binary logical operation			
binary logical operation *has* logical operand 1			
binary logical operation *has* logical operand 2			
bindable target			
cardinality		owl:cardinality	
closed logical formulation	sentence with an interpretation		
closed logical formulation *formalizes* statement			
closed logical formulation *means* proposition			
closed projection			
closed projection *defines* fact type			
closed projection *defines* noun concept			

closed projection ***means*** question			
closed semantic formulation			
conjunction	conjunction with at least two conjuncts	owl:intersectionOf about the extension of a concept and not about the meaning of a sentence	
consequent			
disjunction	disjunction with at least two disjuncts	owl:unionOf *	
equivalence	biconditional	roughly owl:equivalentProperty	
exactly-n quantification		restriction, owl:cardinality n	
exactly-n quantification ***has*** cardinality			
exactly-one quantification		restriction, owl:cardinality 1	
exclusive disjunction	negation of biconditional	---	
existential quantification	quantified sentence of type existential	restriction, owl:someValuesFrom	
implication	implication	---	
implication ***has*** antecedent			
implication ***has*** consequent			
inconsequent			
instantiation formulation	atomic sentence or atom	rdf:type	
instantiation formulation ***binds to*** bindable target			
instantiation formulation ***considers*** concept			
logical formulation	sentence		
logical formulation ***constrains*** projection			
logical formulation kind			
logical formulation ***restricts*** variable		owl:Restriction - for specific kinds of restrictions (value, number)	

logical negation	negation	roughly owl:complementOf	
logical operand	argument of a functional term		
logical operand 1	argument of a functional term, first in sequence		
logical operand 2	argument of a functional term, second in sequence		
logical operation	term representing the operation for a functional term		
logical operation ***has*** **logical operand**			
maximum cardinality		owl:maxCardinality	
minimum cardinality		owl:minCardinality	
modal formulation	irregular sentence	---	
modal formulation ***embeds*** **logical formulation**			
nand formulation	negation of conjunction	---	
necessity formulation			
nor formulation	negation of disjunction	---	
noun concept formulation			
numeric range quantification		restriction, owl:minCardinality n AND restriction, owl:maxCardinality m	
numeric range quantification ***has*** **maximum cardinality**			
numeric range quantification ***has*** **minimum cardinality**			
objectification			
objectification ***binds to*** **bindable target**			
objectification ***considers*** **logical formulation**			
obligation formulation			
permissibility formulation			

possibility formulation			
projecting formulation			
projecting formulation ***binds to*** bindable target			
projecting formulation ***has*** projection			
projection			
projection ***has*** auxiliary variable			
projection ***is on*** variable			
projection position			
quantification	quantified sentence		
quantification ***introduces*** variable	approximately binding sequence for quantified sentence		
quantification ***scopes over*** logical formulation	body for quantified sentence		
role binding	binding sequence		
role binding ***binds to*** bindable target	binding		
role ***has*** role binding			
scope formulation			
semantic formulation			
set ***has*** cardinality			
set projection			
universal quantification	quantified sentence of type universal	restriction, owl:allValuesFrom	
variable	name/term	individual or blank node	
variable ***has*** projection position			
variable ***is free within*** semantic formulation			
variable ***is unitary***		approximately a functional property	
variable ***ranges over*** concept		---	

whether-or-not formulation	truth function operation	---	
whether-or-not formulation *has* consequent			
whether-or-not formulation *has* inconsequent			
SEMANTIC FORMULATION - Nominalization			
answer nominalization			
fact type nominalization			
proposition nominalization			
proposition nominalization *binds to* bindable target			
proposition nominalization *considers* logical formulation			
question nominalization			
FACT MODELS			
concept *is closed in* conceptual schema			
conceptual schema			
conceptual schema *includes* concept			
conceptual schema *includes* fact model			
fact model *includes* fact			
fact model *is based on* conceptual schema			
fact type *is internally closed in* conceptual schema			

10.3 Requirements for Formal Logic Conformance

10.3.1 General Requirements for Formal Logic Interpretation

Necessity:	Each concept and element of guidance represented in an interchange file that conforms to clause 2.2.5 or 2.2.6 is in a single body of shared meanings of a semantic community.
Necessity:	Each body of shared meanings represented in an interchange file that conforms to clause 2.2.5 or 2.2.6 is considered independently of others, with the exception that there can be adoption between communities and semantic equivalence.
Necessity:	Each conceptual schema of a fact model that conforms to clause 2.2.5 or 2.2.6 is for at most one body of shared meanings.
Necessity:	Given a fact model, a compliant interchange file that conforms to clause 2.2.5 or 2.2.6 includes a representation of every fact that is in that fact model.

10.3.2 Enforcing a Restricted Higher Order Interpretation

Necessity:	Each instance of a concept in a fact model that uses a higher order interpretation is consistent with Henkin semantics.
Note:	If a fact model is inconsistent with Henkin semantics, there is generally a mapping by which one or more fact models with a restricted higher order interpretation can be produced.

10.3.3 Enforcing a First Order Interpretation

Necessity:	Each instance of a concept in a fact model that uses a first order interpretation is a first-order instance.
Note:	If fact model is inconsistent with a first order interpretation, there is generally a mapping by which one or more fact models with a first order interpretation can be produced.
Note:	A body of shared meanings that conforms to 10.3.2 always conforms to 10.3.2 "vacuously," that is, no role has an instance that is a meaning.

~~This document defines how the SBVR concepts as defined for the Meaning and Representation Vocabulary and for the Logical Formulation of Semantics Vocabulary may be mapped to terms of formal logic.~~

~~NOTE: In this subclause, the following textual styles have semantic importance:~~

- ~~Logical Formulation of Semantics Vocabulary designations and Meaning and Representation Vocabulary designations are in *italics*~~
- ~~Formal logic terms are underlined~~

~~10.2 Mapping of SBVR Business Terms to Formal Logic~~

~~The following tables show the suggested mappings to formal logics terms for the selected subset of (important) SBVR business terms. Any formatted term that is used in the "Mapping to Formal Logic" is either a formal logic term (underlined) or an SBVR business term (*italics*) that is mapped somewhere else in this table.~~

~~10.2.1 Concepts with Different Mappings for 'First-Order' and 'Restricted Higher-order~~

~~SBVR Term~~	~~'First Order' Mapping to Formal Logic~~	~~'Restricted Higher Order' Mapping to Formal Logic~~	~~Comment~~
~~object type~~	~~first-order type~~	~~restricted higher-order type~~	
~~instance~~	~~first-order instance~~	~~restricted higher-order instance~~	

~~Concepts with a Single Mapping for 'First-Order' and 'Restricted Higher-order'~~

~~Mapping to Formal Logic~~

~~Issue 9447 Revise text~~
~~Issue 9609 Remove text (see fact entry)~~
~~Issue 9258 Replace text (logical negation entry)~~

~~SBVR Term~~	~~Mapping to Formal Logic~~	~~Comment~~
~~state of affairs~~	~~state of affairs~~	
~~antecedent~~	~~antecedent~~	
~~atomic formulation~~	~~atomic formula~~	
~~cardinality~~	~~nonnegative integer that is the number of members in a set~~	
~~concept~~	~~type~~	
~~conceptual fact model~~	~~formal model~~	
~~conceptual schema~~	~~domain grammar~~	
~~consequent~~	~~consequent~~	
~~equivalence / material equivalence~~	bidirectional ~~implication~~.	
~~extension~~	~~population~~	

~~fact~~	~~logical formulation that is taken as true, in the sense that it means a proposition that is taken as true. Each actuality instantiates an unordered set of one or more roles, but for each ordering of roles predicate readings may be given, e.g., 'Terry likes Norma' and 'Norma is liked by Terry' express the same fact. There is no formal way of determining whether two primitive (non-derived) predicates have the same meaning, but equivalences between predicates may be explicitly asserted if known.~~	~~How one ascertains what is true, whether by assertion, observation or other means, is outside the scope of the SBVR specification. However, taking a formulation as true must be consistent with epistemic commitment. The term 'fact' is here defined to be consistent with the operations of truth-functional logic, which produce results based on true and false.~~
~~fact type~~	~~type of actuality, e.g., Person likes Person.~~	
~~implication / material implication~~	~~implication~~	
~~inconsequent~~	~~logical formulation that is a logical operand irrelevant to the logical result of a logical formulation such as of a whether-or-not formulation.~~	~~A term invented for SBVR.~~
~~integer~~	~~integer~~	
~~logical formulation~~	~~an abstract expression of a well-formed logical formula (wff)~~	
~~logical negation~~	~~logical formulation that applies the logical "NOT" operation (~) to a negand logical operand~~	
~~maximum cardinality~~	~~cardinality that is a maximum in a range of cardinalities, such as for an at-most-n quantification~~	
~~minimum cardinality~~	~~cardinality that is a minimum in a range of cardinalities, such as for an at-least-n quantification~~	
~~modal formulation~~	~~logical formulation (wff) that applies a modality to an embedded logical formulation (wff).~~	~~The embedded wff must either be or else contain (embed directly or indirectly) a closed wff.~~
~~Issue 9721: Remove entries~~		
~~necessity~~	~~logical formulation whose main (top level) operator is the alethic (it is necessary that). The SBVR term actually means the proposition expressed by such a logical formulation.~~	
~~obligation~~	~~logical formulation whose main (top level) operator is the deontic O (it is obligatory that). The SBVR term actually means the proposition expressed by such a logical formulation.~~	

~~permissibility~~	~~logical formulation whose main (top level) operator is the deontic P (it is permitted that). The SBVR term actually means the proposition expressed by such a logical formulation.~~	
~~possibility~~	~~logical formulation whose main (top level) operator is the alethic ◊ (it is possible that). The SBVR term actually means the proposition expressed by such a logical formulation.~~	
~~proposition~~	~~proposition~~	
~~quantification~~	~~logical formulation (wff) that applies a logical quantification operation (i.e., a quantifier) to a logical variable.~~	
~~reference scheme~~	~~chosen way of identifying instances of a given concept~~	
~~role~~	~~an object hole in a predicate~~	
~~role binding~~	~~connection of an atomic formulation to a literal value or variable for a particular role of the fact type that is the basis of the atomic formulation~~	
~~set~~	~~set~~	
~~statement~~	~~The communication act of uttering or inscribing a sentence to declare a proposition~~	
~~variable~~	~~logical variable~~	

~~Concepts that Inherit their Formal Logic Mapping from their Generalization~~

~~Generalization Providing Formal Logic Mapping~~

~~Issue 9258 Add "binary logical operation"/ Remove text~~
~~Issue 9721 Replace "claim" with "formulation"~~
~~Issue 9731 Remove "fact type formulation"~~

~~SBVR Term~~	~~Generalization Providing Formal Logic Mapping~~	~~Comment~~
~~actuality~~	~~state of affairs~~	
~~aggregation formulation~~	~~logical formulation~~	
~~at-least-n quantification~~	~~quantification~~	

~~at-most-n quantification~~	~~quantification~~	
~~at-most-one quantification~~	~~quantification~~	
~~auxiliary variable~~	~~variable~~	
~~bag projection~~	~~projection~~	
~~binary fact type~~	~~fact type~~	
~~characteristic~~	~~fact type~~	
~~closed logical formulation~~	~~logical formulation~~	
~~closed projection~~	~~projection~~	
~~closed semantic formulation~~	~~semantic formulation~~	
~~concept type~~	~~object type~~	
~~condition 1~~	~~logical operand~~	
~~condition 2~~	~~logical operand~~	
~~conjunct 1~~	~~logical operand~~	
~~conjunct 2~~	~~logical operand~~	
~~conjunction~~	~~logical formulation~~	
~~disjunct 1~~	~~logical operand~~	
~~disjunct 2~~	~~logical operand~~	
~~disjunction / exclusive disjunction~~	~~logical formulation~~	
~~exactly-n quantification~~	~~quantification~~	
~~exactly-one quantification~~	~~quantification~~	
~~exclusive disjunct 1~~	~~logical operand~~	
~~exclusive disjunct 2~~	~~logical operand~~	
~~exclusive disjunction~~	~~logical formulation~~	
~~existential quantification~~	~~quantification~~	
~~fact type formulation~~	~~logical formulation~~	
~~individual concept~~	~~object type~~	
~~instantiation formulation~~	~~logical formulation~~	

~~logical formulation kind~~	~~object type~~	
~~logical operand~~	~~logical formulation~~	
~~logical operand 1~~	~~logical operand~~	
~~logical operand 2~~	~~logical operand~~	
~~logical operation~~	~~logical formulation~~	
~~nand formulation~~	~~logical formulation~~	
~~necessity claim formulation~~	~~modal formulation~~	
~~negand~~	~~logical operand~~	
~~nonnegative integer~~	~~integer~~	
~~nor formulation~~	~~logical formulation~~	
~~fact type formulation~~	~~logical formulation~~	
~~numeric range quantification~~	~~quantification~~	
~~objectification~~	~~logical formulation~~	
~~obligation claim formulation~~	~~modal formulation~~	
~~permissibility claim formulation~~	~~modal formulation~~	
~~positive integer~~	~~integer~~	
~~possibility claim formulation~~	~~modal formulation~~	
~~projecting formulation~~	~~logical formulation~~	
~~projection~~	~~semantic formulation~~	
~~projection position~~	~~positive integer~~	
~~proposition nominalization~~	~~logical formulation~~	
~~set projection~~	~~projection~~	
~~universal quantification~~	~~quantification~~	
~~whether-or-not formulation~~	~~logical formulation~~	
~~binary logical operation~~	~~logical formulation~~	

~~Concepts whose Formal Logic Mapping is specified in their Specializations~~

~~Issue 9832 Remove text~~
~~Issue 9721 Remove modality~~

SBVR Term	Specialization(s) Providing Formal Logic Mapping	Comment
~~bindable target~~	~~variable, text~~	
~~modality~~	~~alethic modality~~, ~~deontic modality~~	
~~semantic formulation~~	~~logical formulation~~, ~~projection~~	
~~thing~~	~~(Every other concept is a specialization of *thing*.)~~	~~The current definition of *thing* is: "anything perceivable or conceivable."~~

11 Business Vocabulary

Issue 9958 Change forms of expression to fact type forms
Issue 9941 Replace symbol with designation

The following vocabulary provides words for describing business vocabularies along with the designations and fact type forms ~~forms of expression~~ they contain. A full description of a business vocabulary involves its relationship to semantic communities and speech communities, its relationship to other vocabularies, the concepts represented, their definitions and other information about them.

Vocabulary for Describing Business Vocabularies

Language: English

Included Vocabulary: Meaning and Representation Vocabulary

11.1 Business Meaning

11.1.1 Communities, Meanings & Vocabularies

Issue 9939, 9958, 9607 Replace figure
Issue 9960 Add “See Annex H.” to figures in this clause.
Issue 9930 Replace figure, replace figure caption to all figures in this Clause
Issue 9955 Replace figure, add Reference Scheme to community, add text, replace text

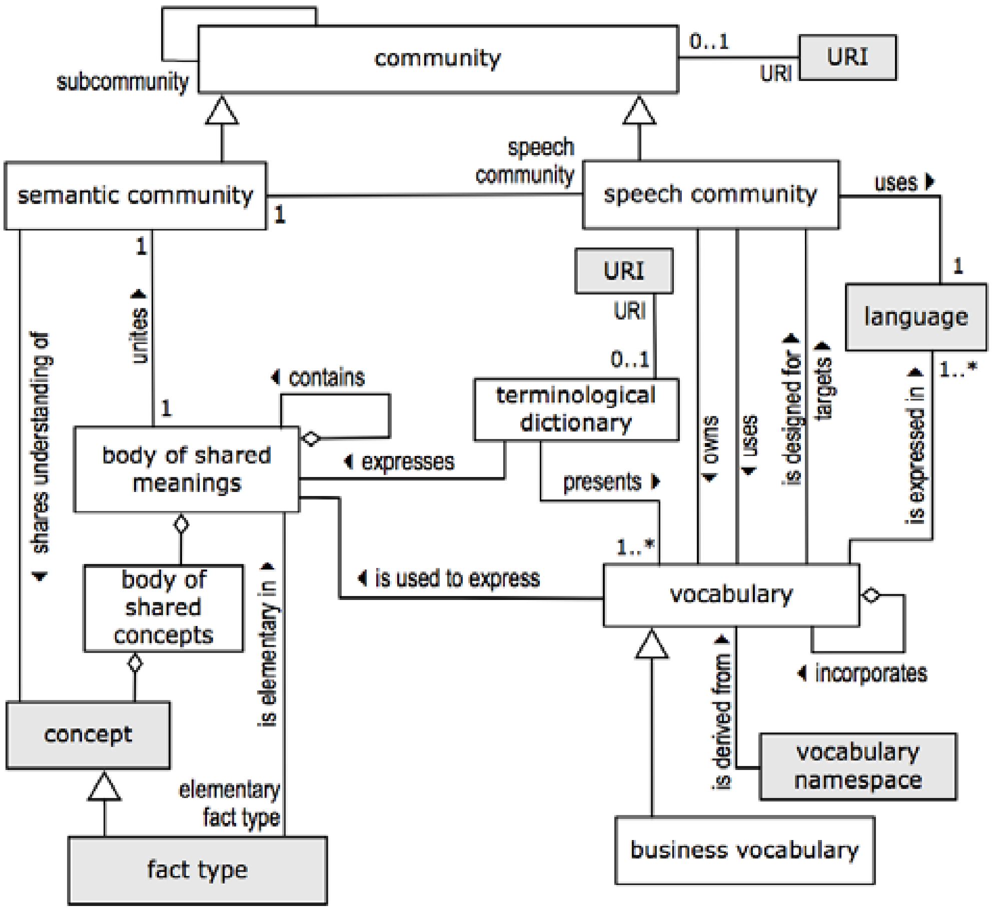


Figure 11.1

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

11.1.1.1 Communities

community

Definition:	group of people having a particular unifying characteristic in common
Dictionary Basis:	group of people having a religion, race, profession, or other particular characteristic in common [NODE 'community']
Reference Scheme:	a URI *of* the community
Example:	The Car Rental Community -- people who work in the car rental business
Example:	The EU-Rent Community -- all EU-Rent employees
Example:	The EU-Rent German Community -- employees of EU-Rent's German division

community *has* URI

Definition:	the URI uniquely identifies the community

Necessity: Each URI *is* the URI *of* at most one community.

semantic community

Definition: community whose unifying characteristic is a shared understanding (perception) of the things that they have to deal with

Example: The EU-Rent Community -- those who share the body of concepts about general and specific things of importance to the EU-Rent business.

speech community

Definition: subcommunity *of* a given semantic community whose unifying characteristic is the vocabulary and language that it uses

Dictionary Basis: group of people sharing a characteristic vocabulary, and grammatical and pronunciation patterns for use in their normal intercommunication [W3ID 'speech community']

Example: The EU-Rent German Community shares the German-based vocabulary of designations used in EU-Rent's business. The designations include German words for EU-Rent's concepts plus designations adopted from other languages.

speech community *uses* language

Definition: the speech community communicates in the language

Necessity: Each speech community *uses* exactly one language.

Issue 10423 Replace text/Remove text

---

semantic community *has* speech community

~~**semantic community *is of* speech community**~~

~~Synonymous Form:~~ ~~semantic community *has* speech community~~

Necessity: Each speech community *is of* exactly one semantic community.

subcommunity

Concept Type: role

Definition: community that *is* a distinct grouping within another community

Dictionary Basis: distinct grouping within a community [NODE 'sub-community']

~~**subcommunity *is of* community**~~

community *has* subcommunity

Definition: the subcommunity is a distinct grouping within the community

~~Synonymous Form:~~ ~~community *has* subcommunity~~

11.1.1.2 Bodies of Shared Meanings

Issue 9711 Revise text
Issue 9607 Remove text

---

body of shared meanings

Definition: set of concepts and elements of guidance for which there is a shared understanding in a given semantic community

~~Dictionary Basis: set containing all objects or elements and of which all other sets are subsets [NODE 'universal set']~~

Example: The EU-Rent Car Rental Business has a body of shared meanings which contains the set of concepts of general and specific things of importance to the EU-Rent car rental business

---

Issue 10504 Add entry and definition
Issue 11300 Remove entry

~~body of shared meanings *includes* body of shared guidance~~

~~Definition: the body of shared guidance is the set of all elements of guidance in the body of shared meanings uniting a semantic community that takes the elements of guidance as true~~

Issue 9467 Remove text
Issue 9607 Remove entry, add entry
Issue 9955 Revise text

~~semantic community *understands* body of shared meanings~~

~~Synonymous Form: body of shared meanings *is understood by* semantic community~~

~~Necessity: Each semantic community *understands* exactly one body of shared meaning.~~

~~Necessity: Each body of shared meaning *is understood by* exactly one semantic community.~~

body of shared meanings *unites* semantic community

Definition: the body of shared meanings *is* the set of concepts and elements of guidance for which there is a shared understanding in the semantic community

Necessity: Each semantic community *is united by* exactly one body of shared meanings.

Necessity: Each body of shared meanings *unites* exactly one semantic community.

Note: Understanding the body of shared meanings that unites a semantic community is an obligation for participation in the semantic community. Communication within the community is based on an assumption of mutual understanding of the body of shared meaning.

body of shared meanings *includes* body of shared concepts

body of shared concepts

Definition: all of the concepts within a body of shared meanings

body of shared concepts *includes* concept

Concept Type: partitive fact type

Synonymous Form: concept *is included in* body of shared concepts

semantic community *shares understanding of* concept

Synonymous Form: concept *has shared understanding by* semantic community

body of shared meanings$_1$ *contains* body of shared meanings$_2$

Concept Type: partitive fact type

Definition: the body of shared meanings *includes* everything in the other body of shared meanings

elementary fact type

Definition: fact type whose facts cannot be split into smaller units of information that collectively provide the same information as the original

Concept Type: role

Example: branch *has* storage capacity

Example: service depot *is included in* local area

Example: rental car *has* fuel level *at* date/time

Example: Counter-example (this would ***not*** be considered an elementary fact type): car manufacturer *delivers* consignment *to* branch. This is not elementary because a consignment is always from at most one car manufacturer and is always to at most one branch. So the counter-example is equivalent to the combination of two binary fact types: car manufacturer *delivers* consignment and consignment *is delivered to* branch.

fact type *is elementary in* body of shared meanings

Definition: within the body of shared meanings, the fact type cannot be decomposed into a set of two or more fact types that collectively have the same meaning as the fact type

Synonymous Form: body of shared meanings *has* elementary fact type

Necessity: Each elementary fact type *of* a body of shared meanings *is in* the body of shared meanings.

Necessity: A fact of an elementary fact type of a body of shared meanings is not equivalent to the conjunction of two or more Facts of other fact types in the body of shared meanings.

Issue 9955 Change title

11.1.1.3 Vocabularies and Terminological Dictionaries

Issue 9958 Change forms of expression to fact type forms
Issue 9955 Delete text

---

vocabulary

Definition: set *of* designations and fact type forms ~~forms of expression~~ primarily drawn from a single language to express concepts within a body of shared meanings

Dictionary Basis: sum or stock of words employed by a language, group, individual, or work, or in a field of knowledge [MWCD 'vocabulary ']

~~Reference Scheme:~~ ~~a URI *of* the vocabulary~~

Example: The sets of designations represented in EU-Rent's internal glossaries, in the natural languages in which the company does business, together with the vocabularies it has adopted, including those defined in:
* Industry standard glossaries for car rental business,

* Standard (e.g., ISO) glossaries of business terms,
* Authoritative dictionaries for the relevant natural languages.

Issue 9939 Add text
Issue 9955 Delete entry

~~vocabulary *has* URI~~

Issue 9467 Remove text
Issue 9955 Replace text / add text / revise text /. remove text

speech community *owns* vocabulary

Definition:	the speech community determines the contents of the vocabulary
~~Synonymous Form:~~	~~vocabulary *is owned by* speech community~~
Note:	The speech community that owns a vocabulary has the authority to change the content of the vocabulary.

speech community *uses* vocabulary

~~Synonymous Form:~~	~~vocabulary *is used by* speech community~~
Note:	A speech community may use a vocabulary that is owned by a different speech community.

~~vocabulary *targets* speech community~~

~~Synonymous Form:~~	~~speech community *is targeted by* vocabulary~~

vocabulary *is designed for* speech community

Synonymous Form:	vocabulary *targets* speech community
Definition:	the vocabulary is created for use by a speech community that does not own the vocabulary
Example:	A speech community of specialists (such as accountants of engineers) creates a “layman’s vocabulary” for their specialization, to be used in discourse with general management.
Example:	The legal department of a company creates a vocabulary to be used for legal documents, such as contracts.

vocabulary *is expressed in* language

Definition:	the designations of the vocabulary are primarily within the language
Synonymous Form:	language *expresses* vocabulary
Synonymous Form:	vocabulary *uses* language
Necessity:	Each vocabulary *is expressed in* at least one language.
Note:	Typically, the language would be a natural language, but not necessarily. See ‘language’.

~~vocabulary *includes* designation~~

~~Concept Type:~~	~~partitive fact type~~
~~Synonymous Form:~~	~~designation *is included in* vocabulary~~

~~vocabulary *includes* fact type form~~

~~vocabulary *includes* form of expression~~

~~Concept Type: partitive fact type~~

~~Synonymous Form: fact type form form of expression *is included in* vocabulary~~

vocabulary$_1$ *incorporates* vocabulary$_2$

Concept Type: partitive fact type

Definition: the vocabulary$_1$ *includes* each designation and fact type form that *is included in* the vocabulary$_2$

Note: When more than one vocabulary is included, a hierarchy of inclusion can provide priority for selection of definitions.

Synonymous Form: vocabulary$_2$ *is incorporated into* vocabulary$_1$

business vocabulary

Definition: vocabulary that is under business jurisdiction

vocabulary *is used to express* body of shared meanings

Definition: the vocabulary includes designations and fact type forms of the concepts in the body of shared meanings

vocabulary namespace *is derived from* vocabulary

Definition: the designations and fact type forms of the vocabulary namespace are from the vocabulary

Note: This specification does not require any particular process of derivation. But a typical process is that all designations and fact type forms that are directly distinguishable by their expressions are put into one vocabulary namespace. In the case of one or more designations or fact type forms being undistinguishable except by their subject fields, an additional vocabulary namespace is derived specifically for those subject fields.

terminological dictionary

Definition: collection of representations including at least one designation or definition of each of a set of concepts from one or more specific subject fields, together with other representations of facts related to those concepts

Source: based on ISO 1087-1 English (3.7.1) [‘terminological dictionary’]

Reference Scheme: a URI *of* the terminological dictionary

Note: Terminological dictionaries include designations and fact type forms representing concepts, and definitions, descriptions, descriptive examples, notes, structural rule statements and other representations of information about the concepts.

terminological dictionary *has* URI

Definition: the URI uniquely identifies the terminological dictionary

Necessity: Each URI *is* the URI *of* at most one terminological dictionary.

terminological dictionary *presents* vocabulary

Definition: the terminological dictionary sets forth representations related to the designations and fact type forms of the vocabulary

Necessity: Each terminological dictionary *presents* at least one vocabulary.

terminological dictionary *expresses* body of shared meanings

Definition: the terminological dictionary includes representations of the concepts in the body of shared meanings

11.1.2 Concepts & Characteristics

Issue 10633 Replace figure
Issue 10572 Replace figure
Issue 11297 Replace figure

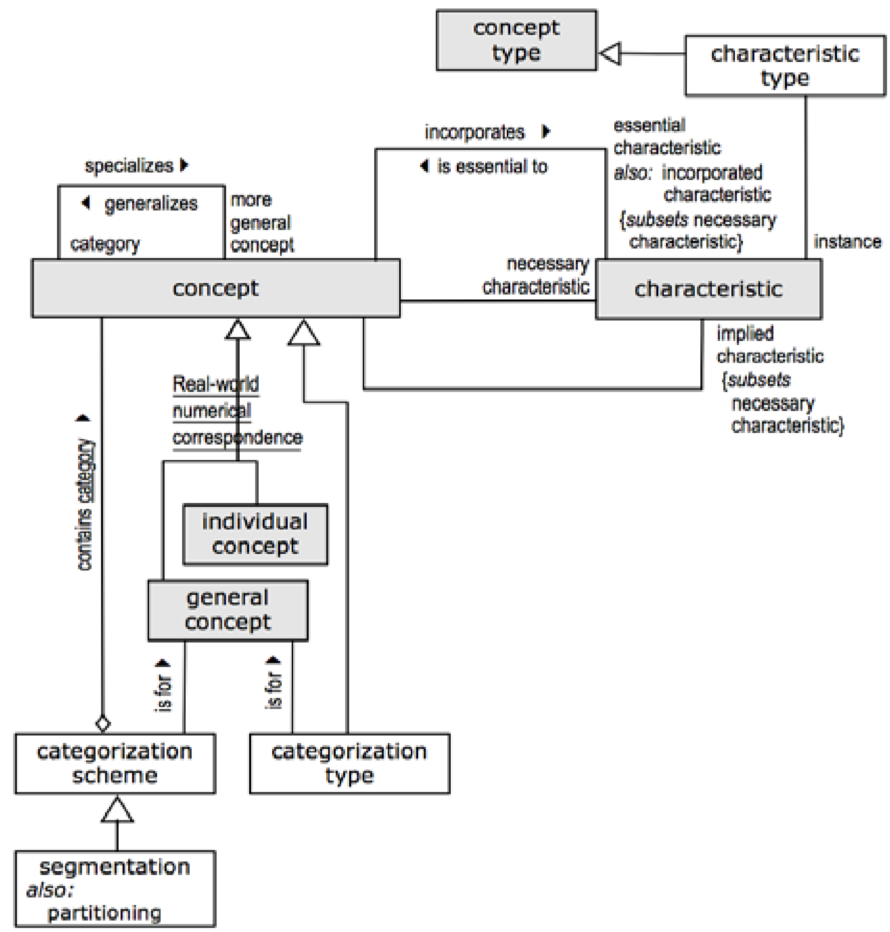


Figure 11.2

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

11.1.2.1 Kinds of Concept

Issue 9948 Move Necessity statement from General Concept
Remove general concept entry

Real-world Numerical Correspondence

Definition:	the categorization scheme *of* the concept 'concept' that *classifies* a concept based on whether or not the concept always corresponds to one specific real-world individual
Necessity:	The concept 'individual concept' *is included in* Real-world Numerical Correspondence.
Necessity:	The concept 'general concept' *is included in* Real-world Numerical Correspondence.

~~general concept~~

Source: ~~based on ISO 1087-1 (English) (3.2.3) ['general concept']~~

Definition: ~~concept that conceivably corresponds to a number of things by reason of their having something in common~~

Note: ~~A definition is given along with the source in order to state the definition in terms of SBVR.~~

Note: ~~ISO 1087-1 (English) partitions 'concept' into 'individual concept' (necessarily corresponding to one thing at a time) and 'general concept' (possibly corresponding to two or more things at a time).~~

Necessity: ~~No general concept *is* an individual concept.~~

Example: ~~The concept 'rental car' corresponding to cars that are rented~~

Example: ~~The concept 'rental customer' corresponding to EU-Rent customers that rent cars~~

11.1.2.2 Kinds of Characteristic

Issue 10572 Add text, remove text, remove entry
Issue 10574 Add entries

---

essential characteristic

Source: ISO 1087-1 (English) (3.2.6) ['essential characteristic']

Definition: characteristic which is indispensable to understanding a concept

Synonym: incorporated characteristic

Concept Type: role

characteristic *is essential to* concept

Definition: ~~the concept *incorporates* the characteristic and the characteristic is essential to understanding the concept~~

Synonymous Form: ~~concept *incorporates* essential characteristic~~

See: concept *incorporates* characteristic

Synonymous Form: concept *has* essential characteristic

Concept Type: is-property-of fact type

necessary characteristic

Definition: characteristic that is always true of each instance *of* a given concept

Concept Type: role

concept *has* necessary characteristic

Definition: the necessary characteristic is always true of each instance *of* the concept

Example: If the characteristic 'car is small' is a necessary characteristic of the concept 'compact car', then every compact car is always small

implied characteristic

Definition: necessary characteristic *of* a given concept that *is* not *incorporated by* the concept

Concept Type: role

Necessity: A concept has an implied characteristic only if it follows by logical implication from some combination of incorporations of characteristics by concepts and/or structural rules that the characteristic is always attributed to each instance of the concept.

concept *has* implied characteristic

Definition: the implied characteristic *is* a necessary characteristic *of* the concept and the concept does not *incorporate* the implied characteristic

delimiting characteristic

Source: ISO 1087-1 (English) (3.2.7) ['delimiting characteristic']

Definition: essential characteristic used for distinguishing a concept from related concepts

Concept Type: role

Note: Delimiting characteristics of a concept are inherited as essential characteristics by all categories of that concept.

~~concept *has* delimiting characteristic~~

~~Definition:~~ ~~the characteristic *is essential to* the concept and the characteristic serves to distinguish the concept from others~~

~~Concept Type:~~ ~~is-property-of-fact-type~~

characteristic type

Source: ISO 1087-1 (English) (3.2.5) ['type of characteristics']

Definition: category of [the concept] 'characteristic' which serves as a criterion of subdivision when establishing concept systems

General Concept: concept type

Necessity: Each instance *of* each characteristic type *is* a characteristic.

Example: The extension of the characteristic type 'color' includes the characteristics 'thing is blue', 'thing is red', 'thing is green', etc.

Issue 11153 Add text

11.1.2.3 Categorization Schemes

category

Source: ISO 1087-1 (English) (3.2.16) ['specific concept']

Definition: concept in a generic relation having the broader intension

Concept Type: role

Dictionary Basis: secondary or subordinate category [NODE 'subcategory']

Note: The broader intension of a category means that the category *incorporates* more characteristics than its more general concept. Thus, it is possible that a category has a smaller extension than its more general concept.

more general concept

Source: ISO 1087-1 (English) (3.2.15) ['generic concept']

Definition: concept in a generic relation having the narrower intension

Concept Type: role

Note: The narrower intension of a more general concept means that the more general concept *incorporates* fewer characteristics than any of its categories. Thus, it is possible that a more general concept has a larger extension than its categories.

Issue 9454 Add text

---

concept$_1$ *has* more general concept$_2$

See: concept$_1$ *specializes* concept$_2$

Synonymous Form: concept$_2$ *has* category$_1$

categorization scheme

Definition: scheme for partitioning things in the extension *of* a given general concept into the extensions *of* categories *of* that general concept

Example: The general concept 'person' categorized by age range and gender into categories 'boy', 'girl', 'man', 'woman'.

Dictionary Basis: an orderly combination of related parts [AH (3) 'scheme']

categorization scheme *is for* general concept

Definition: the general concept is divided into category(s) by the categorization scheme

Necessity: Each categorization scheme *is for* at least one general concept.

Synonymous Form: general concept *has* categorization scheme

categorization scheme *contains* category

Definition: the category is included in the categorization scheme as one of the categories divided into by the scheme

Synonymous Form: category *is included in* categorization scheme

Concept Type: partitive fact type

Necessity: Each category that *is included in* a categorization scheme that *is for* a general concept *is* a category *of* that general concept.

segmentation

Definition: categorization scheme whose contained categories are complete (total) and disjoint with respect to the general concept that has the categorization scheme

Synonym: partitioning

partitioning

See: segmentation

categorization type

Definition: concept whose instances are, or are in one-to-one correspondence with, meaningful-to-the-business categories of another concept

Note: A categorization type is either partial or complete. It is complete if it necessarily categorizes everything of the general concept that it is for.

Example: EU-Rent's categorization type for EU-Rent's concept of 'branch' whose instances are categories of branch: 'airport branch', 'agency', and 'city branch'.

categorization type *is for* general concept

Synonymous Form: general concept *has* categorization type

Issue 10633 Remove text

~~**category *is inactive***~~

~~Necessity:~~ ~~A category *is inactive* if and only if the category *is* a concept that plays no role in a fact type.~~

11.1.3 Kinds of Definition

Issue 10572 Replace figure, add text
Issue 9952 Replace figure
Issue 9955 Replace figure
Issue 10790 Replace figure / add text

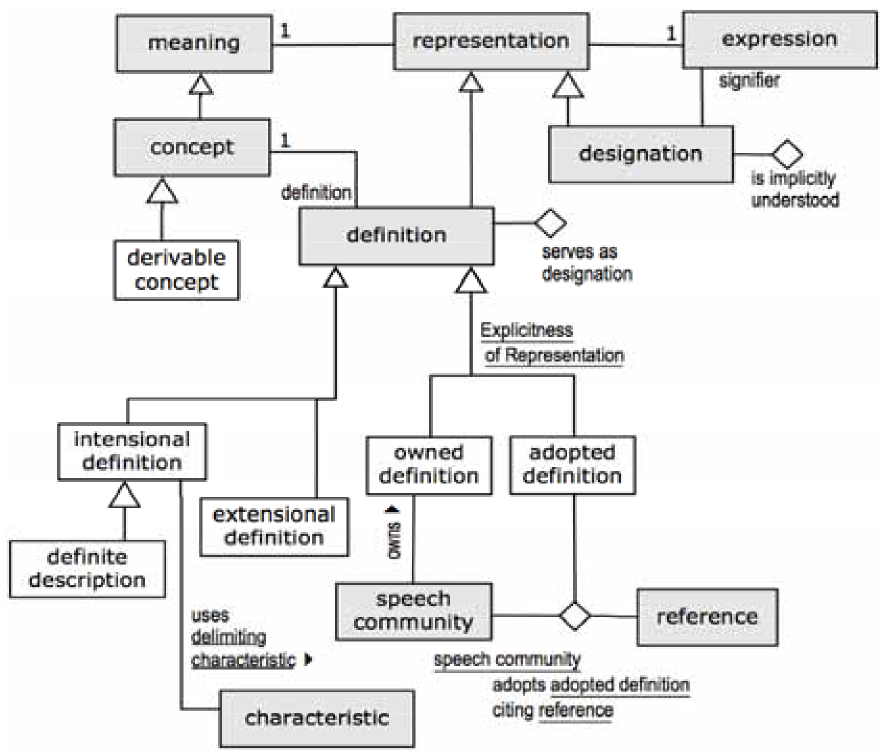


Figure 11.3

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

---

intensional definition

Source:	ISO 1087-1 (English) (3.3.2) ['intensional definition']
Definition:	definition which describes the intension of a concept by stating the superordinate concept and the delimiting characteristics
General Concept:	definition
Necessity:	No intensional definition *is* an extensional definition.

intensional definition *uses* delimiting characteristic

Definition:	the delimiting characteristic serves to distinguish the concept defined by the intensional definition from other concepts

definite description

Definition:	intensional definition of an individual
Example:	the car movement that has the movement id "UK-12345-abc-xyz"
Necessity:	Each definition *of* an individual concept *is* a definite description.
Necessity:	Each definite description *is* the definition *of* an individual concept.
Necessity:	Each definite description uses a reference scheme for the individual.

extensional definition

Source: ISO 1087-1 (English) (3.3.3) ['extensional definition']

Definition: description of a concept by enumerating all of its subordinate concepts under one criterion of subdivision

General Concept: definition

Necessity: No extensional definition *is* an intensional definition.

Explicitness of Representation

Definition: the categorization scheme *of* the concept 'definition' that *classifies* a definition based on whether it is owned by its speech community or adopted by its speech community

owned definition

Definition: definition that a speech community 'owns' and is responsible for creating and maintaining

Necessity: The concept 'owned definition' *is included in* Explicitness of Representation.

Example: EU-Rent 'owns' its definition of the concept of 'barred driver'.

speech community *owns* owned definition

Issue 9955 Replace text, remove text

---

adopted definition

Definition: definition that a speech community adopts from an external source by providing a reference to the definition

Necessity: The concept 'adopted definition' *is included in* Explicitness of Representation.

Necessity: Each adopted definition must be for a concept in the body of shared meanings *of* the semantic community *of* the speech community.

~~Necessity:~~ ~~The speech community that *adopts* an adopted definition does not *own* the source vocabulary *of* the adopted definition.~~

~~Reference Scheme:~~ ~~source vocabulary~~

Example: EU-Rent adopts definition 2b of 'law' from Merriam-Webster Unabridged, using the terms 'law' (primary) and 'statute' for the concept.

Note: The primary term used for the concept does not have to be the same as the primary term in the source. For example, EU-Rent might have taken the definition of 'law' from MWU, but used 'statute' as the primary term for the concept.

~~source vocabulary~~

~~Concept Type:~~ ~~role~~

~~Definition:~~ ~~vocabulary that *is the source of* an adopted definition~~

~~Note:~~ ~~The speech community that owns a source vocabulary and any speech community that adopts definitions from the source vocabulary are usually in different semantic communities.~~

~~speech community *adopts* adopted definition *from* source vocabulary~~

speech community *adopts* adopted definition *citing* reference

Definition: the speech community agrees that the definition identified by the reference can serve as the adopted definition

Issue 9952 Replace text

~~definition is used as term of concept~~

~~Definition:~~ ~~the definition of the concept that has no term defined is used as the term for that concept~~

~~Note:~~ ~~In the case of a concept for which no term is given, the concept is represented by its definition.~~

definition *serves as designation*

Definition: the definition acts as a designation of the concept defined by the definition

Note: In the case of a concept for which no designation is given, the concept is represented by its definition.

~~term is implicitly understood~~

~~Definition:~~ ~~the term is generally understood within by its owning community without an explicit definition~~

designation *is implicitly understood*

Definition: the designation is generally understood by its owning community without an explicit definition for the concept it designates

derivable concept

Definition: concept whose extension can be determined from its definition or from rules

11.1.4 Conceptualization Decisions

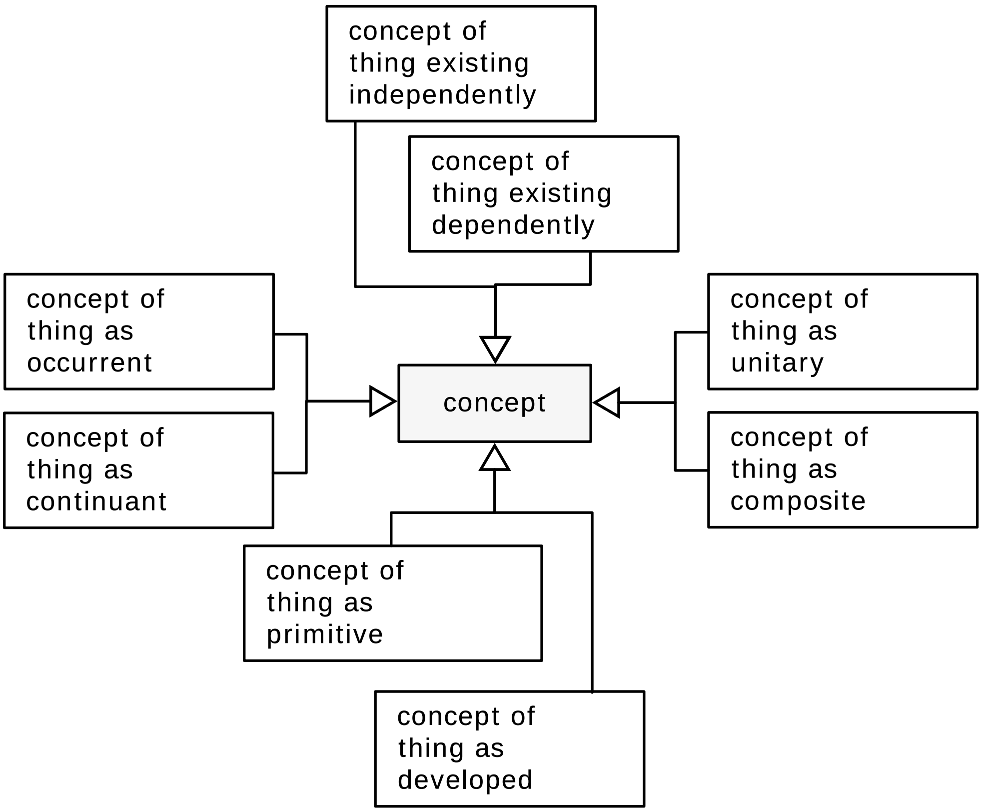


Figure 11.4

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

concept of thing as unitary

Definition: concept that conceptualizes its instances as not being made up of discrete parts or elements

Note: A thing is conceptualized as unitary if a semantic community doesn't think of it as having components, even though some other community may be aware of and concerned about its decomposition.

Example: EU-Rent finance department treats a car as unitary, while its maintenance staff treat it as composite.

concept of thing as composite

Definition: concept that conceptualizes its instances as being made of discrete parts or elements that have corresponding concepts in their own right

Necessity: No concept of thing as unitary *is* a concept of thing as composite.

concept of thing as primitive

Definition: concept that conceptualizes its instances as not being developed or derived from anything else

Dictionary Basis: not developed or derived from anything else [NODE 'primitive']

concept of thing as developed

Definition: concept that conceptualizes its instances as being developed or derived from something else

Necessity: No concept of thing as primitive *is* a concept of thing as developed.

concept of thing as occurrent

Definition: concept that conceptualizes its instances as existing only at a point in time

Dictionary Basis: the fact of something existing or being found in a place or under a particular set of conditions [NODE 'occurrence' 2] + the fact or frequency of something happening [NODE 'occurrence' 1]

concept of thing as continuant

Definition: concept that conceptualizes its instances as existing over a period of time

Dictionary Basis: a thing that retains its identity even though its states and relations may change. [NODE 'continuant' 2]

Necessity: No concept of thing as occurrent *is* a concept of thing as continuant.

concept of thing existing independently

Definition: concept that conceptualizes each instance to exist independently of other things such that existence cannot be ended by the ending of the existence of any other thing

concept of thing existing dependently

Definition: concept that conceptualizes each instance as existing only as long as one or more other things continue to exist

Necessity: No concept of thing existing independently *is* a concept of thing existing dependently.

11.1.5 Fact Type Templating

Issue 9455 Replace figure
Issue 9948 Replace figure

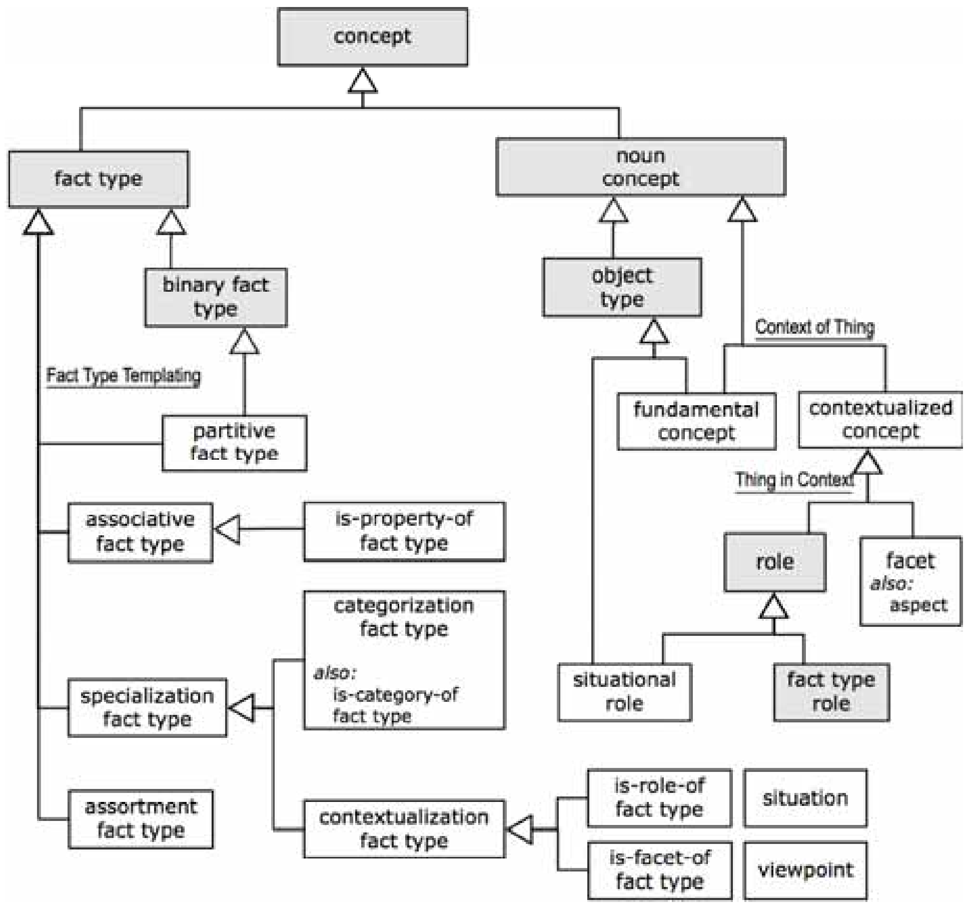


Figure 11.5

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

11.1.5.1 Kinds of Fact Type

Fact Type Templating

Definition: the categorization scheme *of* the concept 'fact type' that *classifies* a fact type *based on* the semantic nature *of* the fact type

associative fact type

Definition: fact type that *has* more than one role and that has a nonhierarchical subject-oriented connection drawn from experience, based on practical rather than theoretical considerations

Source: based on ISO 1087-1 (English) (3.2.23) ['associative relation', 'pragmatic relation']

Necessity: The concept 'associative fact type' *is included in* Fact Type Templating.

Example: The fact type 'additional driver is authorized in rental'.

Example: The fact type 'car manufacturer supplies car model'.

Example: The fact type ‘car manufacturer delivers consignment to branch’.

is-property-of fact type

Definition: associative fact type that is defined with respect to a first given concept and a second given concept such that each instance *of* the fact type *is* an actuality that an instance *of* the first concept constitutes an essential quality of an instance *of* the second concept

Dictionary Basis: an essential quality [SOED ‘property’]

Dictionary Basis: so important as to be indispensable. Something so important to a thing’s very nature that without it that thing would not be the same thing. Something that cannot be removed without destroying the thing itself or its distinguishing character [MWDS & NODE ‘essential’]

Dictionary Basis: an intelligible mark or indication by means of which a thing may be identified or its constitution understood [MWDS ‘quality’]

Necessity: Each instance *of* an is-property-of fact type *is* an actuality that a thing has a particular property.

Necessity: The concept ‘is-property-of fact type’ *is included in* Fact Type Templating.

Example: The fact type ‘engine size is property of car model’.

Issue 9948 Replace text

---

partitive fact type

Definition: ~~fact type that *has* two roles and~~ binary fact type where each instance *is* an actuality that a given part is in the composition of a given whole

Source: based on ISO 1087-1 (English) (3.2.22) [‘partitive relation’]

Necessity: Each instance *of* a partitive fact type *is* an actuality that a part is in the composition of a whole.

Necessity: The concept ‘partitive fact type’ *is included in* Fact Type Templating.

Example: The fact type ‘country is included in region’. An example of an instance of that fact type is that Sweden is included in Scandinavia.

Example: The fact type ‘branch is included in local area’.

Example: The fact type ‘car model is included in car group’.

specialization fact type

Definition: categorization fact type or contextualization fact type

General Concept: fact type

Note: The essential property is that, for these kinds of fact types, an instance of a more specific concept is one and the same thing as an instance of a more general concept.

Necessity: The concept ‘specialization fact type’ *is included in* Fact Type Templating.

Issue 9958 Change form of expression to fact type form

---

assortment fact type

Definition: fact type that is defined with respect to a given general concept and a given individual concept such that each instance of the fact type is an actuality that the one instance of the individual concept is an instance of the general concept

Dictionary Basis: to place in the same group with others : associate in a class [MWU (3) "assort"]

Necessity: Each instance of an assortment fact type is an actuality that the instance of a given individual concept is an instance of a given general concept.

Necessity: The concept 'assortment fact type' is included in Fact Type Templating.

Example: A fact type with the fact type form ~~form of expression~~ 'Euro is a currency'. The one instance of the fact type would be Euro being a currency.

Example: A fact type with the fact type form ~~form of expression~~ 'Ford Motor Company is a car manufacturer'. The one instance of the fact type would be the Ford Motor Company being a car manufacturer.

Example: A fact type with the fact type form ~~form of expression~~ 'Switzerland is a country'. The one instance of the fact type would be Switzerland being a country.

11.1.5.2 Contextualization

Issue 9446 Add text
Issue 9958 Change form of expression to fact type form

---

aspect

See: facet

facet

Definition: concept that incorporates only those characteristics of another concept being contextualized which are relevant to a given viewpoint

General Concept: contextualized concept

Dictionary Basis: a particular way in which some thing may be considered; its particular nature, appearance, or quality; the particular part or feature of it [NODE 'aspect']

Necessity: The concept 'facet' is included in Thing in Context.

Synonym: aspect

situation

Definition: set of circumstances that provides the context from which roles played by instances of a concept may be understood or assessed

Dictionary Basis: a set of circumstances in which one finds oneself; a state of affairs [NODE 'situation']

Dictionary Basis: the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood or assessed [NODE 'context']

Note: A situation typically pertains for some period of time, during which changes may occur.

Example: The situation "breakdown during rental" is the set of circumstances that starts with the breakdown of a car while on rental and continues until the broken-down car, having been replaced by another car, has been returned to an EU-Rent location.

viewpoint

Definition: perspective from which something is considered

categorization fact type

Definition: fact type that is defined with respect to a given concept and another concept that *is* a category *of* that concept such that each instance *of* the fact type *is* an actuality that a particular instance *of* the concept *is* also an instance *of* the category

Synonym: is-category-of fact type

General Concept: specialization fact type

Necessity: No categorization fact type *is* a contextualization fact type.

Example: A fact type with the fact type form ~~form of expression~~ "customer is of the category 'high-end customer'." An instance of the fact type would be a particular customer being a high-end customer.

Example: A fact type with the fact type form ~~form of expression~~ "customer is of the category 'risky customer'." An instance of the fact type would be a particular customer being a risky customer.

contextualization fact type

Definition: is-role-of fact type or is-facet-of fact type

General Concept: specialization fact type

Necessity: The concept 'contextualization fact type' *is included in* Fact Type Templating.

is-role-of fact type

Definition: fact type that is defined with respect to a given concept, a given role and a given category *of* the concept 'situation' such that each instance *of* the fact type *is* an actuality that a particular instance *of* the concept plays the role in a particular instance *of* the category *of* 'situation'

Necessity: Each instance *of* an is-role-of fact type *is* an actuality that a particular thing plays a role in a particular situation.

Necessity: Each is-role-of fact type is defined with respect to a concept, a role and a category *of* the concept 'situation'.

Necessity: The concept 'is-role-of fact type' *is included in* Fact Type Templating.

Example: A fact type with the fact type form ~~form of expression~~ "rental car plays the role 'replacement car' in the fact type 'breakdown during rental has replacement car'." An instance of the fact type would be a particular breakdown during a particular rental having a particular replacement car. Note that a separate fact type relates a breakdown during rental to a rental.

Example: A fact type with the fact type form ~~form of expression~~ "branch plays the role 'pick-up branch' in the fact type 'rental has pick-up branch'." An instance of the fact type would be a particular rental having a particular pick-up branch.

Note: A fact type is understood to be an is-role-of fact type based on a pattern of meaning, not on a fact type form ~~form of expression~~. There is no requirement of any particular wording of the fact type form ~~form of expression~~ of an is-role-of fact type.

is-facet-of fact type

Definition: fact type that is defined with respect to a given concept, a given facet and a given category *of* the concept 'viewpoint' such that each instance *of* the fact type *is* an actuality that a particular instance *of* the category *of* the concept 'viewpoint' gives consideration to the facet *of* a particular instance *of* the concept

General Concept: contextualization fact type

Necessity: Each instance *of* an is-facet-of fact type *is* an actuality that a particular viewpoint gives consideration to a facet *of* particular thing.

Necessity: Each is-facet-of fact type is defined with respect to a concept, a facet and a category *of* the concept 'viewpoint'.

Necessity: The concept 'is-facet-of fact type' *is included in* Fact Type Templating.

Example: A fact type with the fact type form ~~form of expression~~ 'financial accounting considers rental car as asset'. An instance of the fact type would be a particular financial accounting considering a particular rental car to be an asset.

Example: A fact type with the fact type form ~~form of expression~~ 'rental considers person as driver'. An instance of the fact type would be a particular rental considering a particular person to be a driver.

Note: A fact type is understood to be an is-facet-of fact type based on a pattern of meaning, not on a fact type form ~~form of expression~~.

Note: A given community may choose to include only one facet.

11.1.5.3 Contextualized Concepts

Context of Thing

Definition: the segmentation *of* the concept 'noun concept' that *classifies* a noun concept *based on* whether the noun concept's real-world individuals are perceived by the semantic community as in their uninvolved essence or as to their involvement in a situation or from a viewpoint

---

Issue 9948 Replace text, add text, remove fact type role entry

---

fundamental concept

Definition: ~~noun concept~~ object type whose real-world individuals are perceived by a given semantic community as being in their essence, apart from any situation in which they are involved or viewpoint from which they are considered

Dictionary Basis: a property or group of properties of something without which it would not exist or be what it is [NODE 'essence']

Concept Type: concept type

Necessity: No fundamental concept *is* a contextualized concept.

Necessity: The concept 'fundamental concept' *is included in* Context of Thing.

Example: car (as contrasted with 'rental car')

Example: person (as contrasted with 'customer')

Note: Each semantic community decides what is within its body of shared meanings. A concept that is considered as fundamental by one community may, to another community, be a role or facet or category of a more broadly-defined concept.

contextualized concept

Definition: role or facet

General Concept: noun concept

Necessity: The concept 'contextualized concept' *is included in* Context of Thing.

Thing in Context

Definition: the segmentation *of* the concept 'contextualized concept' that *classifies* a contextualized concept *based on* whether the contextualization reflects a situation (i.e., a role) or a viewpoint (i.e., a facet)

Necessity: The concept 'role' *is included in* Thing in Context.

~~**fact type role**~~

~~Definition:~~ ~~role *of* a fact type~~

Issue 9446 Add text
Issue 9948 Replace text, add text

---

situational role

Definition: ~~role that a thing plays in a situation~~ object type that corresponds to things based on their playing a part, assuming a function, or being used in some situation

General Concept: object type, role

Concept Type: concept type

11.2 Business Representation

11.2.1 Symbolization

Issue 9456, 9467, 9958, 9930, 9941, 9952, 9948, 11298 Replace figure

---

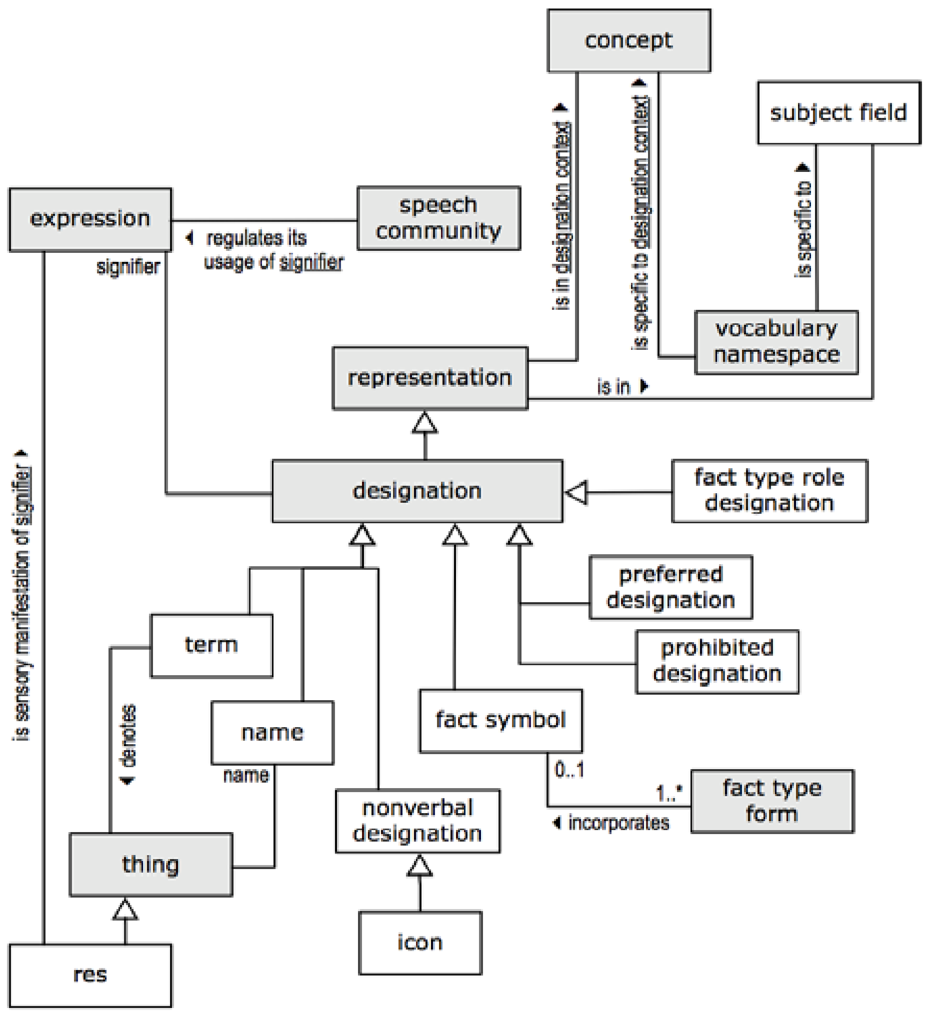


Figure 11.6

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

Issue 9941 Revise title, add entries, remove entries, revise symbol context entries

11.2.1.1 Subject Fields

subject field

Definition: field of specific knowledge

Source: ISO 1087-1 (English) (3.1.2) ['subject field']

representation *is in* subject field

Definition: the representation is recognized and used in discourse regarding the subject field

vocabulary namespace *is specific to* subject field

Definition: each designation and fact type form that *is in* the vocabulary namespace *is in* the subject field

representation *is in* designation context

Definition: the representation is recognized and used in discourse regarding the designation context

vocabulary namespace *is specific to* designation context

Definition: each designation and fact type form that *is in* the vocabulary namespace *is in* the designation context

~~symbol~~

~~Definition:~~ ~~representation *of* a concept by a signifier as *owned by* a speech community and used within a symbol context which means the concept and denotes its extension~~

~~Dictionary Basis:~~ ~~thing conventionally regarded as representing or recalling something else (by possessing analogous qualities or by association in fact or thought, especially a material object representing something abstract) [SOED 'symbol']~~

~~Necessity:~~ ~~Each symbol *realizes* exactly one designation.~~

~~Necessity:~~ ~~Each symbol *is owned by* exactly one speech community.~~

~~Necessity:~~ ~~Each symbol *is understood anywhere within* at most one symbol context.~~

~~Note:~~ ~~The 'symbol context' component of a symbol is only required for a symbol when needed for disambiguation of signifier -- i.e., in the case where a speech community uses one signifier to represent two (or more) distinct concepts.~~

~~Necessity:~~ ~~Given a speech community, signifier and symbol context, at most one symbol *is owned by* the speech community and *is understood anywhere within* symbol context and *realizes* a designation that *has* the signifier.~~

~~Reference Scheme:~~ ~~the speech community that *owns* the symbol and the designation that *realizes* the symbol and a symbol context that *qualifies understanding of* the symbol~~

~~Issue 9462 Remove text/Revise text~~

~~Reference Scheme:~~ ~~the speech community that *owns* the symbol and the designation that *realizes* the symbol and the set *of* each symbol contexts that *qualify understanding of* the symbol~~

~~symbol *realizes* designation~~

~~Issue 9467 Replace text/Remove text~~

~~speech community *owns* symbol~~

~~Synonymous Form:~~ ~~speech community *owns* symbol~~

~~Issue 9930 Remove text / Replace text~~

~~symbol *is understood anywhere within* symbol context~~

~~Synonymous Form:~~ ~~symbol context *is of* symbol~~

~~Synonymous Form:~~ ~~symbol *has* qualifying symbol context~~

~~Synonymous Form:~~ ~~symbol context *qualifies understanding of* symbol~~

~~Note:~~ ~~The sense of 'anywhere' means that the symbol is understood inside of, but not outside of, the symbol context. In other words, this specifies the one symbol context that is the largest (broadest) within which the symbol is understood.~~

designation context

Concept Type: role

Definition: concept that characterizes the domain of usage within which the expression *of* a representation has a unique meaning for a given speech community

~~Definition:~~ ~~concept that represents the scope within which a signifier has a unique meaning for a given semantic community~~

~~Necessity:~~ ~~The symbol for a symbol context provides a disambiguating context where that symbol, via the concept that it symbolizes, defines a unique context within which the signifier of a second symbol is uniquely connected to its concept.~~

Example: When EU-Rent uses the term 'site':

* within the context of the concept termed 'vehicle rental' (another EU-Rent term), it denotes EU-Rent's shared understanding of 'a place from which EU-Rent vehicles are picked up and returned'.

* within the context of the concept termed 'vehicle maintenance' (another EU-Rent term), it denotes EU-Rent's shared understanding of 'a place where EU-Rent's vehicle fleet is serviced and repaired'.

Example: When EU-Rent uses the term 'customer':

* within the context of the concept termed 'vehicle rental' (another EU-Rent term), it denotes EU-Rent's shared understanding of 'rental-customer-ness' (Definition: 'an individual who currently has an EU-Rent car on rental, or has a reservation for a future car rental, or has rented a car from EU-Rent in the past 5 years').

* within the context of the concept termed 'vehicle sales' (another EU-Rent term), it denotes EU-Rent's shared understanding of 'car-purchaser-ness' (Definition: 'an individual who has purchased at least one car from EU-Rent that is still within its warranty period').

~~representation *uses* symbol~~

~~Synonymous Form:~~ ~~symbol *is used in* representation~~

~~Example:~~ ~~EU-Rent: late rental return of a car damaged by an additional driver -- i.e., a driver, named on the rental contract, who is not the renter. Used symbols include 'late rental return', 'car', 'damaged', and 'additional driver'.~~

Issue 9941 Add nonverbal designation
Issue 9952 Replace text

11.2.1.2 Kinds of Designation

term

~~Definition:~~ ~~designation that *is for* a concept and that is a word or phrase~~

~~Source:~~ ~~based on ISO 1087-1 (English) (3.4.3 & 3.4.2) ['term' or 'appellation']~~

~~Note:~~ ~~A term is typically a common noun or noun phrase, unless it is a name.~~

~~Definition:~~ ~~designation that has a precisely-limited meaning in some uses, or is peculiar to a science, art, profession, trade, or special subject~~

~~Dictionary Basis:~~ ~~word or expression that has a precise meaning in some uses or is peculiar to a science, art, profession, or subject [MWCD 'term']~~

Source: ISO 1087-1 (English) (3.4.3) ['term']

Definition: verbal designation of a general concept in a specific subject field

General Concept: designation

Note: A term is typically formed using a common noun or noun phrase.

Example: EU-Rent agrees the word 'car' denotes its shared understanding of 'rental-car-ness' within <rental context>.

Example: EU-Rent agrees the word 'vehicle' denotes its shared understanding of 'car-ness' within <rental context>.

Example: EU-Rent agrees the word 'customer' denotes its shared understanding of 'rental-customer-ness' within <rental context>.

Example: EU-Rent agrees the word 'customer' denotes its shared understanding of 'car-purchaser-ness' within <car-sales context> -- i.e., when EU-Rent disposes of cars after they reach their mileage or age threshold.

Example: EU-Rent agrees the word 'renter' denotes its shared understanding of 'rental-customer-ness'. (within any context).

name

~~Definition:~~ ~~term that *is for* an individual concept~~

~~Source:~~ ~~based on ISO 1087-1 (English) (3.4.2) ['appellation']~~

~~Note:~~ ~~A name is often a proper noun.~~

Source: ISO 1087-1 (English) (3.4.2) ['appellation']

Definition: verbal designation of an individual concept

General Concept: designation

Necessity: No name *is* a term

Note: The expression of a name is typically a proper noun.

nonverbal designation

Definition: designation that is not expressed as words of a language

Necessity: No nonverbal designation *is* a term.

Necessity: No nonverbal designation *is* a name.

Note: A verbal designation, such as a term or name, can contain parts that are nonverbal. Some abbreviations are nonverbal while others, being expressed as words, are terms or names.

Issue 11299 Remove Necessities

icon

Definition: nonverbal designation whose signifier is a picture

Dictionary Basis: a usu. pictorial representation [MWCD 'icon']

Example:  as a designation for the concept 'u-turn'

~~Necessity:~~ ~~No icon *is* a term.~~

~~Necessity:~~ ~~No icon *is* a name.~~

Issue 9958 Change form of expression to fact type form
Issue 9952 Revise text

fact symbol

Definition: designation that is for a fact type and that is understood in an ordered context indicated by a fact type form ~~form of expression~~

~~Necessity:~~ ~~No fact symbol *is* a term.~~

Reference Scheme: a fact type form ~~form of expression~~ that *incorporates* the fact symbol

Example: In the expression, "Each customer rents a car," 'rents' is a fact symbol denoting a fact type.

Example: In the expression, "A driver of a car returns the car to a branch office," 'of' is a fact symbol for one fact type (relating a driver to a car) and 'returns to' is another fact symbol denoting a fact type (relating a driver to a car and a branch office).

Issue 9958 Change form of expression to fact type form

fact type form *incorporates* fact symbol

~~form of expression *incorporates* fact symbol~~

Definition: the fact type form ~~form of expression~~ lays out a pattern for using the fact symbol in an expression

Synonymous Form: fact symbol *is incorporated into* fact type form ~~form of expression~~

Necessity: Each fact type form ~~form of expression~~ *incorporates* at most one fact symbol.

Necessity: Each fact symbol *is incorporated into* at least one fact type form ~~form of expression~~.

Issue 9948 Add text

fact type role designation

Definition: designation that *represents* a fact type role and that *is* not a placeholder

11.2.1.3 Designations and Things in the Real-world

Issue 9467 Remove text
Issue 10790 Add text

term *denotes* thing

Definition: the thing *is* an instance *of* the concept that *is represented by* the term

~~Synonymous Form:~~ ~~thing *is denoted by* term~~

thing *has* name

Definition: the thing *is* the instance *of* the individual concept that *is represented by* the name

Synonymous Form: name *references* thing

Note: A use of an individual concept by its name denotes the thing that is in the extension of the individual concept.

Issue 9944 Change Definition text

res

Definition: thing that *is* not a meaning ~~literally anything that exists but is not the result of conceptualization~~

res *is sensory manifestation of* signifier

11.2.1.4 Designation Preference and Prohibition

Issue 9952 Revise text

preferred designation

Definition: designation that is selected by its owning speech community for a given concept from among alternative designations for that concept as being most desirable or productive

Example: EU-Rent's preferred ~~terms~~ designations for indicating the USA Dollar, Canadian Dollar, and Mexican Peso are, respectively, "USD," "CAD," and "MXN" (ISO 4217 currency codes).

Issue 10632 Remove text
Issue 9952 Revise text

prohibited designation

Definition: designation that is declared unacceptable by its owning speech community ~~creating a vocabulary that excludes the symbol~~

Example:	In EU-Rent, use of the dollar sign ($) by itself is prohibited, to avoid confusion between the USA Dollar, Canadian Dollar, and Mexican Peso.
Note:	What is prohibited is the use of a given ~~signifier~~ expression to represent a given meaning. The same ~~signifier~~ expression may be permitted, even preferred, to represent another meaning.
Necessity:	No preferred designation *is* a prohibited designation.

speech community regulates its usage of signifier

11.2.2 Forms of Business Representation

Issue 9467 Replace figure (see also Issue 9934)
Issue 10569, 9930, 9955Replace figure

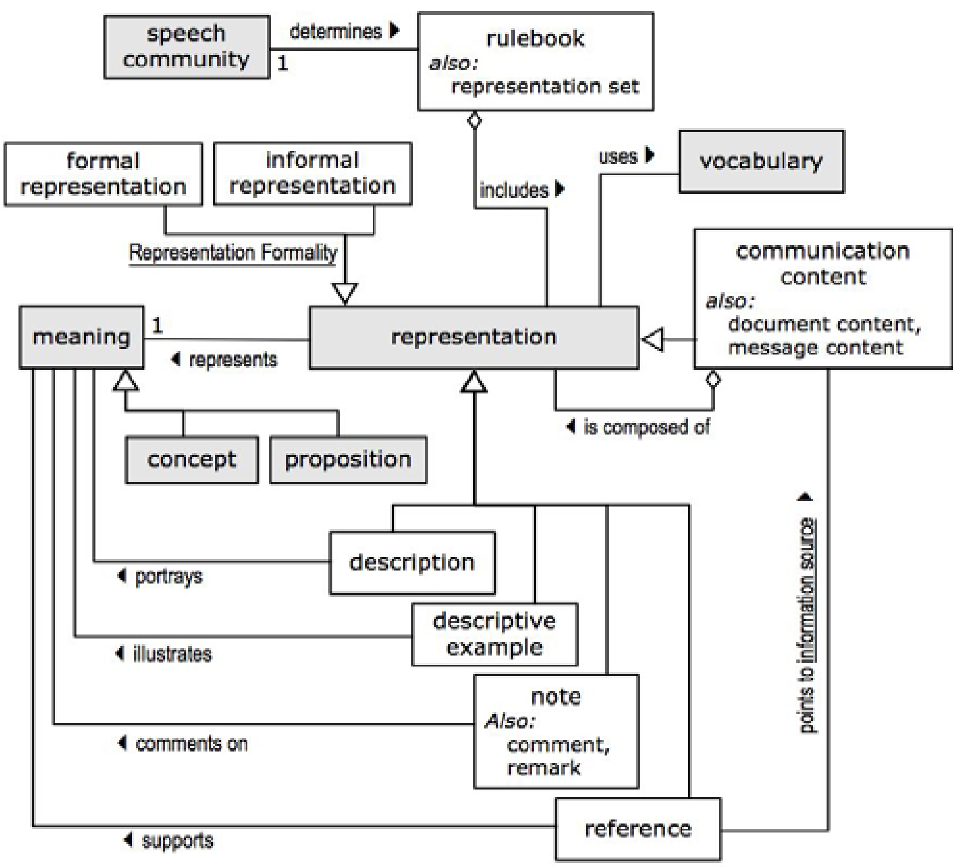


Figure 11.7

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

11.2.2.1 Representation Formality

Representation Formality

Definition: the segmentation *of* the concept 'representation' that *classifies* a representation based on whether or not it is 'formal'

informal representation

Definition: representation in which not every word is annotated ('tagged') in accordance with a notation that can be mapped to SBVR

Necessity: No informal representation *is* a formal representation.

Necessity: The concept 'informal representation' *is included in* Representation Formality.

Note: Some of the words of an informal representation may be annotated -- i.e., defined, or 'tagged', terms, names, verbs, or keywords.

formal representation

Definition: representation in which every word is annotated ('tagged') in accordance with a notation that can be mapped to SBVR

Necessity: No formal representation *is* an informal representation.

Necessity: The concept 'formal representation' *is included in* Representation Formality.

11.2.2.2 Concept Expression

Issue 9467 Remove text (then see Issue 9934)
Issue 9934 Revise text

---

description *portrays* meaning

~~**description *portrays* concept meaning**~~

~~Synonymous Form: description *portrays* concept~~

Note: The meaning of a description that portrays a concept is most likely not that concept. A description can be a statement, in which case, its meaning is a proposition.

description

Definition: representation that provides a detailed account of something, a verbal portrait

Dictionary Basis: a spoken or written representation or account of a person, object, or event [NODE 'description']

Necessity: No description that *portrays* a concept *is* a descriptive example that *illustrates* that concept.

Necessity: No description that *portrays* a concept *is* a note that *comments on* that concept.

Necessity: No description that *portrays* a concept *is* a reference that *supports* that concept.

descriptive example *illustrates* meaning

~~**descriptive example *illustrates* concept meaning**~~

~~Synonymous Form: descriptive example *illustrates* concept~~

Note: The meaning of a descriptive example is typically a proposition.

descriptive example

Definition: representation that provides descriptive material that is a sample of the thing defined

Source: based on MWCD and NODE

Dictionary Basis: one (as an item or incident) that is representative of all of a group or type [MWCD 'example']

Dictionary Basis: a thing characteristic of its kind or illustrating a general rule [NODE 'example']

Necessity: No descriptive example that *illustrates* a concept *is* a definition *of* that concept.

Necessity: No descriptive example that *illustrates* a concept *is* a description that *portrays* that concept.

Necessity: No descriptive example that *illustrates* a concept *is* a note that *comments on* that concept.

Necessity: No descriptive example that *illustrates* a concept *is* a reference that *supports* that concept.

Example: Chris Cushing is an example of EU-Rent's concept of 'rental customer'

Example: The vehicle with VIN#88744332 is an example of EU-Rent's concept of 'rental car'

Issue 10423 Replace text
Issue 10569 Replace heading and remove text

---

note *comments on* meaning

~~Synonymous Form:~~ ~~note *comments on* meaning concept~~

Note: The meaning of a note that comments on a concept is most likely not that concept. A note is typically a statement whose meaning is a proposition.

note

Definition: representation that annotates or explains

Necessity: No note that *comments on* a concept *is* a definition *of* that concept.

Necessity: No note that *comments on* a concept *is* a description that *portrays* that concept.

Necessity: No note that *comments on* a concept *is* a descriptive example that *illustrates* that concept.

Necessity: No note that *comments on* a concept *is* a reference that *supports* that concept.

Synonym: remark

Synonym: comment

comment

See: note

remark

See: note

11.2.2.3 Business Content of a Communication

communication content

Definition: representation that is a subdivision of a written composition that consists of one or more statements and deals with one point or gives the words of one speaker

Source: MWCD (1a)
Synonym: message content
Synonym: document content

document content

See: communication content

message content

See: communication content

communication content *is composed of* representation

Concept Type: partitive fact type

Issue 9467 Remove text (then see Issue 9934)
Issue 9934 Revise text

---

reference *supports* meaning

~~**reference *supports* concept meaning**~~

~~Synonymous Form: reference *supports* concept~~

reference

Definition: representation that is the mention or citation of a source of information used to direct a reader elsewhere for additional information about a given concept

Dictionary Basis: a mention or citation of a source of information in a book or article [NODE 'reference']

Necessity: No reference that *supports* a concept *is* a definition *of* that concept.

Necessity: No reference that *supports* a concept *is* a description that *portrays* that concept.

Necessity: No reference that *supports* a concept *is* a descriptive example that *illustrates* that concept.

Necessity: No reference that *supports* a concept *is* a note that *comments on* that concept.

Example: 'The Highway Code' published by HMSO, 2005.

Example: The descriptions of car models' capacity, fuel economy, and performance taken from the manufacturers' specifications.

reference *points to* information source

Definition: the communication content plays the role of an information source for the reference

information source

Concept Type: role

Definition: communication content that is used as a resource to supply information or evidence

---

Issue 9955 Add text

11.2.2.4 Sets of Business Representations

rulebook

Definition: the set *of* representations *determined by* a given speech community to represent in its language all meanings in its body of shared meanings

Synonym: representation set

Reference Scheme: the speech community that *determines* the rulebook

rulebook *includes* representation

Definition: the representation is an element of the rulebook

Synonymous Form: representation *is included in* rulebook

representation *uses* vocabulary

Definition: the representation is expressed in terms of the vocabulary

speech community *determines* rulebook

Definition: the speech community is responsible for the expression of representations that are included in the rulebook

Necessity: Each rulebook *is determined by* exactly one speech community.

Note: The speech community is responsible for translating the informal representations of the rulebook into the language of the speech community.

12 Business Rules

Vocabulary for Describing Business Rules

Language: English

Included Vocabulary: Vocabulary for Describing Business Vocabularies

---

12.1 Categories of Guidance

Issue 9475/9945 Add text

The *common sense* understanding of ‘rule’ is that a rule always tends to remove some degree of freedom. This *common sense* understanding should be contrasted with that for ‘advice’, where a degree of freedom is never removed, even potentially.

The degree of freedom removed by a rule might concern the behavior of people (in the case of an operative business rule), or their understanding of concepts (in the case of a structural rule). In the latter case, the restricting of freedom is built-in (i.e., “structural” or “by definition”). In the former case, people can still potentially violate or ignore the rule - that is a matter of free will, appropriate enforcement, and sometimes discretion (for example if the rule is offered simply as a guideline or suggestion).

Nonetheless, an operative business rule always mandates or suggests some out-of-bounds criteria for behavior, thereby potentially removing a degree of freedom. For example, the meaning of “It is prohibited that an order be paid by promissory note” indicates that workers are not completely free to accept IOUs for payment of orders. That particular degree of freedom has been removed or diminished. Depending on enforcement level, violating the rule could well invite response - which might be anything from immediate prevention and/or severe sanction, to mild tutelage. Note that other degrees of freedom have not been removed or diminished by this particular rule. For example, unless other rules pertain to how orders are paid, workers are free to accept cash, credit cards, or other means of payment - those means are allowed. The general implication is that rules indirectly prescribe what is allowable - whatever the rules do not specifically proscribe is allowed.

Advice is just the opposite of a rule. Whereas a rule always potentially removes some degree of freedom, advice always confirms or reminds that some degree of freedom does exist or is allowed. That degree of freedom might concern the behavior of people (in the case of an operative business rule), or their understanding of concepts (in the case of a structural rule).

It might be helpful to think of advice as an ‘un-rule’ or ‘no-rule’. For example, the meaning of “It is permitted that an order be paid by cash” is that such behavior is allowed - that indeed, paying by cash is acceptable. In other words, there is (or should be) no rule to the contrary.

Since advice never removes degrees of freedom, why is it sometimes useful to capture? There are many possible reasons, but probably foremost among them are to re-assure workers or others that some degree of freedom does exist; to use as a basis for admonishing workers about applying some rule that actually does not exist; or to ‘remember’ the resolutions to some rule-related issue where the outcome was in favor of ‘no rule’.

Issue 9477 Replace figure (See also Issue 9444)
Issue 9960 Add “See Annex H.” to figures in this clause
Issue 9475/9945 Replace figure
Issue 10525 Replace figure
Issue 9930 Replace figure caption to all figures in this Clause

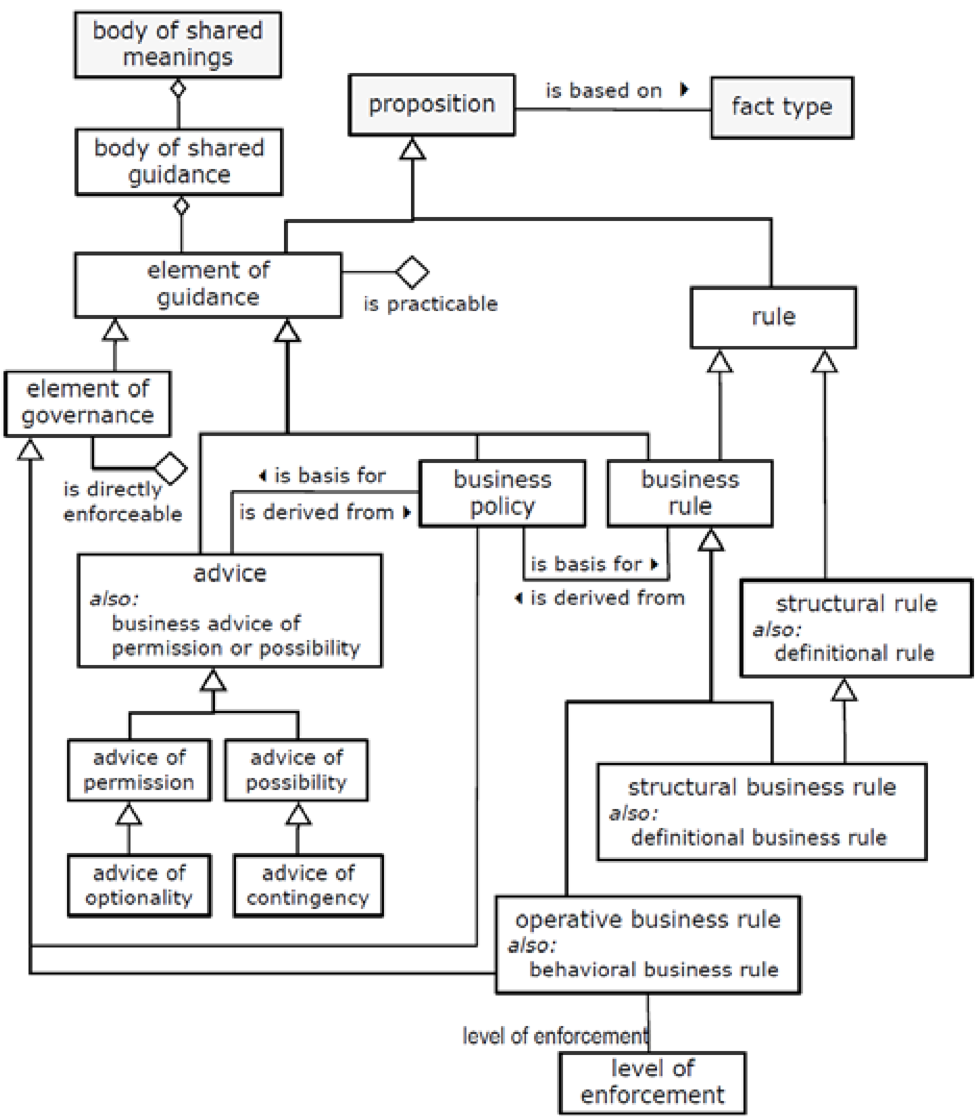


Figure 12.1

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

12.1.1 Guidance

Issue 9477 Remove text

~~directive~~

~~Definition:~~ ~~means that defines or constrains some aspect of an enterprise~~

~~General Concept:~~ ~~proposition~~

~~Note:~~ ~~This sense of 'means' (as in 'ends and means', rather than 'is meant as') arises from the Business Motivation Model [BMM].~~

~~Note:~~ ~~Intended to assert business structure or to control or influence the behavior of the enterprise.~~

~~Note:~~ ~~Its formulation is under the enterprise's control by a party authorized to manage, control or regulate an enterprise, by selection from alternatives in response to a combination of assessments.~~

~~Dictionary Basis:~~ ~~an official or authoritative instruction [NODE]~~

~~directive *is actionable*~~

~~Concept Type:~~ ~~characteristic~~

~~Note:~~ ~~'Actionable' means that a person who knows about the directive could observe a relevant situation (including his or her own behavior) and decide directly whether or not the business was complying with the directive.~~

~~Dictionary Basis:~~ ~~subject to or affording ground for an action or suit at law [MWUD 'actionable']~~

~~Dictionary Basis:~~ ~~a thing done : DEED [MWUD (5a) 'action']~~

~~Note:~~ ~~The sense intended is: "It's actually something you can put to use or apply."~~

body of shared guidance

Definition: all of the elements of guidance within a body of shared meanings

Issue 9477 Add text
Issue 11300 Add Definition

body of shared meanings *includes* body of shared guidance

Definition: the body of shared guidance is the set of all elements of guidance in the body of shared meanings uniting a semantic community that takes the elements of guidance as true

Synonymous Form: body of shared guidance *is included in* body of shared meanings

body of shared guidance *includes* element of guidance

Synonymous Form: element of guidance *is included in* body of shared guidance

Issue 9477 Replace text

element of guidance

General Concept: proposition

Definition: means that guides, defines, or constrains some aspect of an enterprise

Note: This sense of 'means' (as in 'ends and means', rather than 'is meant as') arises from the Business Motivation Model [BMM].

Note: The formulation of an element of guidance is under an enterprise's control by a party authorized to manage, control or regulate the enterprise, by selection from alternatives in response to a combination of assessments.

Issue 9477 Add text
Issue 10580 Revise Definition text

---

element of guidance *is practicable*

Concept Type: characteristic

Definition: the element of guidance is sufficiently detailed and precise that a person who knows the element of guidance can apply it effectively and consistently in relevant circumstances to know what behavior is acceptable or not, or how something is ~~decided or calculated~~ understood.

Dictionary Basis: able to be done or put into practice successfully; able to be used, useful [ODE]

Note: The sense intended is: "It's actually something you can put to use or apply."

Note: The behavior, decision, or calculation can be that person's own.

Note: Whether or not some element of guidance is practicable is decided with respect to what a person with legitimate need can understand from it.

- For an operative business rule, this understanding is about the behavior of people and what form compliant behavior takes.
- For a structural rule, this understanding is about how evaluation of the criteria vested in the rule always produces some certain outcome(s) for a decision or calculation as opposed to others.

Note: A practicable business rule is also always free of any indefinite reference to people (e.g., "you," "me"), places (e.g., "here"), and time (e.g., "now"). By that means, if the person is displaced in place and/or time from the author(s) of the business rule, the person can read it and still fully understand it, without (a) assistance from any machine (e.g., to "tell" time), and (b) external clarification.

element of governance

Definition: element of guidance that is concerned with directly controlling, influencing, or regulating the actions of an enterprise and the people in it

Dictionary Basis: conduct the policy, actions, and affairs of (a state, organization, or people) with authority: control, influence, or regulate (a person, action, or course of events) [ODE, "govern"]

Issue 9477Add text
Issue 9475/9945 Delete Necessity

---

element of governance *is directly enforceable*

Definition: violations of the element of governance can be detected without the need for additional interpretation of the element of governance

Concept Type:	characteristic
Note:	Directly enforceable' means that a person who knows about the element of governance could observe relevant business activity (including his or her own behavior) and decide directly whether or not the business was complying with the element of governance.
Necessity:	Each element of governance that *is directly enforceable is practicable*.
~~Necessity:~~	~~An element of governance that *is* not *directly enforceable is* not *practicable*.~~

Issue 9477 Revise text
Issue 9955 Add text

business policy

Definition:	element of governance that *is* not *directly enforceable* whose purpose is to guide an enterprise
Note:	Compared to a Business Rule, a Business Policy tends to be: - less structured - less discrete or not atomic - less carefully expressed in terms of a standard vocabulary - not directly enforceable.
Dictionary Basis:	definite course or method of action selected (as by a government, institution, group, or individual) from among alternatives and in the light of given conditions to guide and usually determine present and future decisions [MWUD "Policy" 5a]
Necessity:	No business policy *is* a business rule.
Example:	The policy expressed as "A prisoner is considered to be on a hunger strike after missing several meals in a row."
Example:	The policy expressed as "The prison medical authority will intervene if a hunger striker's life is in danger."
Example:	The EU-Rent policy expressed as "Rental cars must not be exported."
Example:	The policy expressed as "Each customer who complains will be personally contacted by a representative of the company."

Issue 9477 Remove "element of guidance is based on fact type" and replace with this text.

proposition *is based on* fact type

Definition:	the proposition is formulated using the fact type
Example:	The EU-Rent business rule that is expressed as "It is obligatory that each rental specifies a car group." (or, in RuleSpeak, "A rental must have a car group.") is based on the EU-Rent fact type 'rental specifies car group'.

12.1.2 Rules

Issue 9477 Revise “Definition”
Issue 9475/9945 Revise Definition
Issue 9579 Remove 3 Dictionary Basis items, Add new Dictionary Basis.

rule

Definition: proposition that is a claim of ~~introduces an~~ obligation or of ~~a~~ necessity

~~Dictionary Basis:~~ ~~standard by which something is judged or valued; criterion [MWUD (2B) ‘rule’]~~

~~Dictionary Basis:~~ ~~principle or standard by which something may be judged or decided [determined] [NODE ‘criterion’]~~

~~Dictionary Basis:~~ ~~standard on which a judgment or decision may be based [MWCD (2) ‘criterion’]~~

Dictionary Basis: one of a set of explicit or understood regulations or principles governing conduct or procedure within a particular area of activity ... a law or principle that operates within a particular sphere of knowledge, describing, or prescribing what is possible or allowable. [ODE]

Issue 9477 Add “General Concept”
Issue 9708 Add text

business rule

Definition: rule that *is* under business jurisdiction

General Concept: rule, element of guidance

Note: A rule’s being “under business jurisdiction” means that it is under the jurisdiction of the semantic community that it governs or guides - that the semantic community can opt to change or discard the rule. Laws of physics may be relevant to a company (or other semantic community); legislation and regulations may be imposed on it; external standards and best practices may be adopted. These things are not business rules from the company’s perspective, since it does not have the authority to change them. The company will decide how to react to laws and regulations, and will create business rules to ensure compliance with them. Similarly, it will create business rules to ensure that standards or best practices are implemented as intended. See subclause A.2.3

business rule *is derived from* business policy

Synonymous Form: business policy *is basis for* business rule

Issue 9444 Add text/Remove text
Issue 9475/9945 Revise text

structural rule

Definition: rule that is a claim of necessity. ~~intended as a definitional criterion~~

~~Necessity:~~ ~~Each structural rule *is* a proposition that another proposition *is* a necessity.~~

Synonym: definitional rule

definitional rule

See: structural rule

Issue 9444 Add “Necessity”
Issue 9475/9945Revise text

---

structural business rule

Definition: structural rule that *is* a business rule

Necessity: Each structural business rule *is practicable*.

Synonym: definitional business rule

definitional business rule

See: structural business rule

Issue 9444 Add, Remove, and Replace “Definition”
Issue 9475/9945 Revise text

---

operative business rule

Definition: business rule that is a claim of obligation

~~Definition:~~ ~~business rule that is intended to produce an appropriate or designed effect and *is directly enforceable*~~

~~Definition:~~ ~~business rule that covers conduct, action, practice, or procedure within a particular activity or sphere~~

~~Definition:~~ ~~business rule that there is an obligation concerning conduct, action, practice or procedure within a particular activity or sphere~~

Definition: element of governance that *is directly enforceable*

Dictionary Basis: a prescribed, suggested, or self-imposed guide for conduct or action : a regulation or principle <his parents laid down the rule that he must do his homework before going out to play> <a very sound rule for any hiker is to mind his own business […] F.D.Smith & Barbara Wilcox> <made it a rule never to lose his temper> […] [MWU (1a) ‘rule’]

Dictionary Basis: a prescribed guide for conduct or action [MWCD ‘rule’]

~~Necessity:~~ ~~Each operative business rule *is* a proposition that another proposition *is* an obligation~~

Necessity: No operative business rule *is* a structural business rule.

~~Necessity:~~ ~~Each operative business rule *is directly enforceable*~~

Synonym: behavioral business rule

behavioral business rule

See: operative business rule

12.1.3 Enforcement

Issue 9477 Revise text

level of enforcement

Definition: ~~something that represents~~ a position in a graded or ordered scale of values that specifies the severity of action imposed in order to put or keep an operative business rule in force

Dictionary Basis: a position on a real or imaginary scale of amount, quantity, extent, or quality [NODE 'level']

Dictionary Basis: compel observance of or compliance with [NODE 'enforcement']

Example: An example set of levels of enforcement, based on [BMM]

Enforcement Level: strict
Definition: strictly enforced (If you violate the rule, you cannot escape the penalty.)

Enforcement Level: deferred
Definition: deferred enforcement (Strictly enforced, but enforcement may be delayed — e.g., waiting for resource with required skills.)

Enforcement Level: pre-authorized
Definition: pre-authorized override (Enforced, but exceptions allowed, with prior approval for actors with before-the-fact override authorization.)

Enforcement Level: post-justified
Definition: post-justified override (If not approved after the fact, you may be subject to sanction or other consequences.)

Enforcement Level: override
Definition: override with explanation (Comment must be provided when the violation occurs.)

Enforcement Level: guideline
Definition: guideline (suggested, but not enforced.)

operative business rule *has* level of enforcement

12.1.4 Possibilities and Permissions

Issue 9477 Revise "Definition"/Remove text, Add 2 "Necessity" items
Issue 9475/9945 Change 12.1.4 Heading, Delete admonition, affirmation, add new entries

~~**admonition**~~

~~Definition:~~ ~~element of guidance that *is practicable* and that *is* a proposition that there is not an obligation or necessity where, by custom or practice, one might be assumed~~

Note: ~~The purpose of an admonition is to preempt application of "rules" that might be assumed by some members of a semantic community, but are not actually rules admitted by the community. Often, the reason for this assumption in a business is that other, similar, businesses have such rules. Typically, the reason for an explicit admonition is that people in the business have mistakenly applied the non-existent "rule" in the past.~~

Example: ~~(In a bank) There is no rule that a person must be over some given age in order to open a savings account: "There is no minimum age for opening a savings account."~~

Example: ~~(In EU-Rent) There is no rule that a rented car can be dropped off only at the return branch specified in the rental agreement: "At the end of a rental, a rental car can be dropped off at any EU-Rent branch."~~

~~There is a related rule that if the drop-off branch is not the specified return branch, the rental will incur a penalty charge, but the importance of this admonition is that EU-Rent wants its cars back, even if they are in the wrong places. It does not want a branch to refuse to accept a car on the grounds that it is not the specified return branch.~~

Necessity: ~~Each admonition *is* a proposition that another proposition *is* not an obligation or a necessity.~~

Necessity: ~~No rule *is* an admonition.~~

Necessity: ~~No business policy *is* an admonition~~

Necessity: ~~No business rule *is* an admonition~~

~~Issue 9477 Revise "Definition"/Remove text/Add 2 "Necessity" items~~

~~affirmation~~

Definition: ~~element of guidance that *is practicable* and that *is* a proposition that there is a permissibility or possibility where, by custom or practice, one might not be assumed~~

Example: ~~(In a bank) "It is possible that an account balance is negative."~~

Example: ~~(In EU-Rent) "A rental car may be dropped off at any EU-Rent branch, even one that is not its scheduled drop-off branch."~~

Necessity: ~~Each affirmation *is* a proposition that another proposition *is* a possibility or a permissibility.~~

Necessity: ~~No rule *is* an affirmation.~~

Necessity: ~~No admonition *is* an affirmation.~~

Necessity: ~~No business policy *is* an affirmation~~.

Necessity: ~~No business rule *is* an afffirmation~~.

advice

Definition: element of guidance that *is practicable* and that is a claim of permission or of possibility

Necessity: No business policy *is* an advice.

Necessity: No business rule *is* an advice.

Synonym: business advice of permission or possibility

Issue 10525 Add new entry

---

advice *is derived from* business policy

Synonymous Form: business policy *is basis for* advice

Issue 9475/9945Add text

---

advice of possibility

Definition: advice that is a claim of possibility

Note: Every necessity implies a possibility. So if a necessity is introduced by a structural rule, there is no practical reason to introduce the implied possibility. In such cases, best practice generally favors keeping the number of elements of guidance to be managed to a minimum.

Example: (In a bank) The element of guidance that "It is possible that an account balance is negative."

Necessity: No advice of possibility *is* an advice of permission.

advice of contingency

Definition: advice of possibility that is a claim of contingency

Note: The purpose of an advice of contingency is to preempt application of "rules" that might be assumed by some members of a semantic community, but are not actually definitional rules admitted by the community. Often, the reason for this assumption in a business is that other, similar businesses have such rules. Typically, the reason for providing such explicit advice is that people in the business have mistakenly applied the non-existent "rule" in the past.

Note: In alethic logic, a proposition that is possible but not necessary is termed 'contingent'. If people in a business were to treat it as a necessity, they would miscategorize things in the real world. This typically leads to refusal of activity (that should be permitted) because unnecessary preconditions are not met, e.g., refusing to accept a rental booking because the person wishing to rent is under 21.

Example: (In EU-Rent) Advising that it is not necessary for a qualified driver to be over 21. This might be expressed in various ways, for example as: "It is neither necessary nor impossible that the age of a qualified driver is at least 21," or "It is possible (but not necessary) that a qualified driver be under 21."

Example: (In EU-Rent) Advising that it is not necessary for a bad experience that occurs during a rental to be notified before the end of the rental. This might be expressed in various ways, for example as: "It is neither necessary nor impossible that the notification date/time of a bad experience during a rental is the actual return date/time of the rental or earlier." It is possible (but not necessary) that the notification of a bad experience during a rental occurs after the car has been returned."

advice of permission

Definition: advice that is a claim of permission

Note: Every obligation implies a permission. So if an obligation is introduced by a behavioral rule, there is no practical reason to introduce the implied permission. In such cases, best practice generally favors keeping the number of elements of guidance to be managed to a minimum.

Example: (In a bank) There is no rule that a person must be over some given age in order to open a savings account: "There is no minimum age for opening a savings account." This is understood as an advice of permission because 'minimum age' is defined as "age that must be reached in order to take part in a given activity" and no restriction has been placed on it. In other words, the behavior 'opening a bank account' is not to be disallowed based on age.

Example:	There is no rule that orders placed by FAX will not be accepted: “Placing an order by FAX is acceptable.” In other words, placing an order by FAX is not prohibited.

advice of optionality

Definition:	advice of permission that is a claim of optionality
Note:	The purpose of an advice of optionality is to preempt application of “rules” that might be assumed by some members of a semantic community, but are not actually behavioral rules imposed by the community. Often, the reason for this assumption in a business is that other, similar businesses have such rules. Typically, the reason for such explicit advice is that people in the business have mistakenly applied the non-existent “rule” in the past.
Note:	In deontic logic, a proposition that is permissible but not obligatory is termed ‘optional’. If people in a business were to treat it as an obligation, they would demand compliance that is not required by the business, e.g., to be shown picture id, or that the car be driven to the specified return branch (as the following examples illustrate).
Example:	(In EU-Rent) Advising that it is not obligatory that a renter show picture identification at the time of a rental pick-up. This might be expressed in various ways, for example as: “It is neither obligatory nor prohibited that at rental pick-up time the renter shows picture identification.” or “It is not obligatory (but permitted) that a renter shows picture id in order to pick up his car.”
Example:	(In EU-Rent) Advising that it is not obligatory (or prohibited) that a rented car be dropped off only at the return branch specified in the rental agreement. This might be expressed, for example, as “At the end of a rental, it is not obligatory (but permitted) that a rental car be dropped off at the rental agreement-specified EU-Rent return branch.”

12.2 Statements of Guidance

Issue 9475/9945 Add text
Issue 9955 Remove text

The surface syntax people use to express guidance is language-specific. It is also dependent on the particular rule language (e.g., SBVR Structured English, RuleSpeak, ORM, etc.). Clause 12.2 does not standardize any particular rule language. Instead, it provides a normative vocabulary for the kinds of guidance statements that business people assert. These kinds of guidance statements are general with respect to any particular language.

The categories presented in this subclause ~~of the vocabulary~~ are intended for business people. Business people see and hear surface syntax. Therefore, the categories defined in 12.2 are based on form or style of expression. For example, if a business person says “It is obligatory that not p,” the form or style of the expression remains an obligation statement. That interpretation reflects the ‘common sense’ of the statement.

This emphasis on form or style of expression distinguishes this subclause from Clause 10, which provides deeper logical analysis. For example, if a business person says “It is obligatory that not p,” logical analysis following Clause 10 takes the meaning of the expression to be a prohibition (which might not be “common sense”). The key to distinguishing the perspective of 12.2 from the logical analysis of Clause 10 is emphasized by the unfailing use of “statement” in the names of the concepts in 12.2. When “statement” appears, it is always the case that the concept so named refers to the style and form of surface expression, rather than underlying meaning based on logical analysis.

Issue 9475/9945 Replace figure

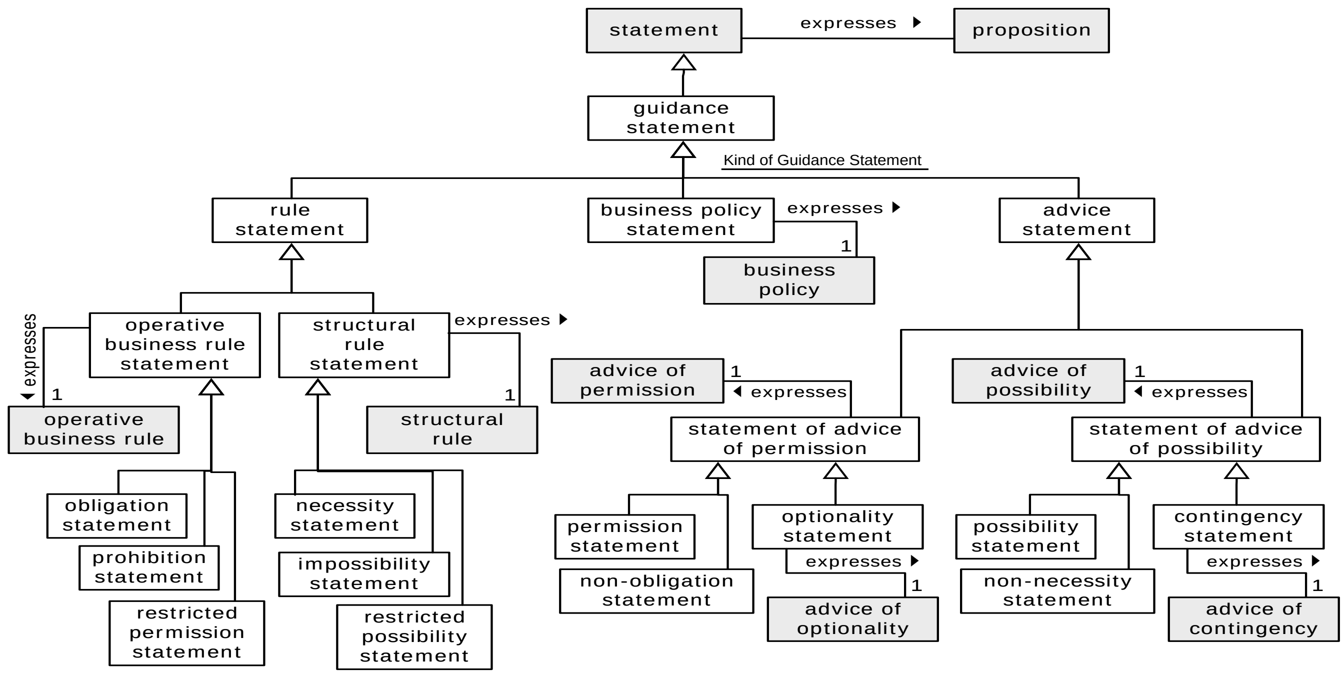


Figure 12.2

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

12.2.1 Categories of Business Statement

Issue 9475/9945 Delete text

~~**business policy statement**~~

~~Definition:~~ ~~statement that *expresses* a business policy~~

guidance statement

Definition: statement that *expresses* an element of guidance

Definition: statement that provides advice or information aimed at resolving a problem or difficulty, especially as given by someone in authority

Dictionary Basis: a statement that provides advice or information aimed at resolving a problem or difficulty, especially as given by someone in authority [NODE 'guidance']

~~Necessity:~~ ~~No guidance statement *is* a business policy statement.~~

Issue 9475/9945 Add text / change text / delete text

Kind of Guidance Statement

Definition: the categorization scheme of the concept 'guidance statement' that *classifies* a guidance statement *based on* the surface syntax of the guidance statement

business policy statement

Definition: guidance statement that *expresses* a business policy

Necessity: The concept 'business policy statement' *is included in* Kind of Guidance Statement.

rule statement

Definition: guidance statement that *expresses* an operative business rule or a structural rule ~~something to be a necessity or obligation~~

Necessity: The concept 'rule statement' *is included in* Kind of Guidance Statement.

~~**business rule statement**~~

~~General Concept:~~ ~~rule statement~~

~~Definition:~~ ~~operative business rule statement or structural business rule statement~~

structural rule statement

Definition: rule statement that *expresses* a structural rule ~~*of* necessary characteristics~~

~~Necessity:~~ ~~Each structural rule statement *expresses* exactly one structural rule.~~

Note: One structural rule can be expressed as various equivalent kinds of statements by introducing or removing negation. The following are examples of the same rule, expressed in three forms.

Example: [as a necessity statement] "It is necessary that the pick-up branch of a one-way rental is not the return branch of that rental."

Example: [as an impossibility statement] "It is impossible that the pick-up branch of a one-way rental is the return branch of that rental."

Example: [as a restricted possibility statement] "It is possible that the pick-up branch of a rental is the return branch of the rental only if the rental is not a one-way rental."

~~**structural business rule statement**~~

~~Definition:~~ ~~structural rule statement that *is* a business rule statement~~

~~Necessity:~~ ~~Each structural business rule statement *expresses* exactly one structural business rule.~~

operative business rule statement

Definition: ~~business~~ rule statement that *expresses* an operative business rule

~~Definition:~~ ~~business rule statement that is a definite or clear expression of an operative business rule in speech or writing~~

~~Dictionary Basis:~~ ~~a definite or clear expression of something in speech or writing [NODE 'statement']~~

~~Necessity:~~ ~~Each operative business rule statement *expresses* exactly one operative business rule.~~

Necessity: No operative business rule statement *is* a structural ~~business~~ rule statement.

Note: One operative business rule can be expressed as various equivalent kinds of statements by introducing or removing negation. The following are examples of the same rule, expressed in three forms.

Example: [as an obligation statement] "It is obligatory that a rental that is open has no driver that is a barred driver."

Example: [as a prohibition statement] "It is prohibited that a rental be open if a driver of the rental is a barred driver."

Example: [as a restricted permission statement] "It is permitted that a rental be open only if no driver of the rental is a barred driver."

~~admonition statement~~

~~Definition:~~ ~~guidance statement that expresses an admonition~~

~~Necessity:~~ ~~Each admonition statement *expresses* exactly one admonition.~~

~~Example:~~ ~~"It is permitted that the drop-off branch of a rental is not the return branch of the rental."~~

~~Example:~~ ~~"The drop-off branch of a rental need not be the return branch of the rental."~~

~~affirmation statement~~

~~Definition:~~ ~~guidance statement that expresses an affirmation~~

~~Necessity:~~ ~~Each affirmation statement *expresses* exactly one affirmation.~~

~~Necessity:~~ ~~No admonition statement *is* an affirmation statement.~~

~~Example:~~ ~~"It is possible that the notification date/time of a bad experience that occurs during a rental is after the actual return date/time of the rental."~~

~~Example:~~ ~~"The notification date/time of a bad experience that occurs during a rental is sometimes after the actual return date/time of the rental."~~

advice statement

Definition: guidance statement that *expresses* an advice of permission or an advice of possibility

Necessity: The concept 'advice statement' *is included in* Kind of Guidance Statement.

statement of advice of permission

Definition: advice statement that *expresses* an advice of permission

Note: One advice of permission can be expressed as various equivalent kinds of statements by introducing or removing negation. The following are examples of the same advice, expressed in alternative forms.

Example: [as a permission statement] "It is permitted that the drop-off branch of a rental is not the return branch of the rental."

Example: [as a non-obligation statement] "It is not obligatory that the drop-off branch of a rental be the return branch of the rental."

Example: [as a non-obligation statement] "The drop-off branch of a rental need not be the return branch of the rental."

statement of advice of possibility

Definition: advice statement that *expresses* an advice of possibility

Example: "The notification date/time of a bad experience that occurs during a rental can be after the actual return date/time of the rental."

Necessity: No statement of advice of possibility *is* a statement of advice of permission.

Note: One advice of possibility can be expressed as various equivalent kinds of statements by introducing or removing negation. The following are examples of the same advice, expressed in two forms.

Example: [as a possibility statement] "It is possible that the notification date/time of a bad experience that occurs during a rental is after the actual return date/time of the rental."

Example: [as a non-necessity statement] "It is not necessary that the notification date/time of a bad experience that occurs during a rental be on or before the actual return date/time of the rental."

12.2.2 Business Statements

Issue 9475/9945 Add text / change text / delete text

12.2.2.1 Business Statements of Operative Business Rules

obligation statement

~~Definition:~~ ~~operative business rule statement that is expressed in a syntactical form for expressing an operative business rule positively in terms of obligation rather than negatively in terms of prohibition~~

~~Definition:~~ ~~operative business rule statement that is expressed in an obligatory form, regardless of language, such as 'It is obligatory that …' or using the word 'must' (but not 'must not'), or listed as a 'requirement~~

Definition: operative business rule statement that is expressed positively in terms of obligation rather than negatively in terms of prohibition

Necessity: No obligation statement *is* a ~~prohibitive~~ prohibition statement.

Necessity: No obligation statement *is* a restricted ~~permissive~~ permission statement.

~~Note:~~ ~~The same rule could also be expressed in another form, such as a prohibitive form, by introducing or removing negation.~~

~~Example:~~ ~~"If the drop-off location of a rental is not the EU-Rent site of the return branch of the rental, then it is obligatory that the rental incurs a location penalty charge."~~

~~Example:~~ ~~"A rental must incur a location penalty charge if the drop-off location of the rental is not the EU-Rent site of the return branch of the rental."~~

Example: "It is obligatory that a rental incurs a location penalty charge if the drop-off location of the rental is not the EU-Rent site of the return branch of the rental."

Example: "A rental must incur a location penalty charge if the drop-off location of the rental is not the EU-Rent site of the return branch of the rental."

prohibition statement

~~**prohibition prohibitive statement**~~

~~Definition:~~ ~~operative business rule statement that is expressed in a syntactical form for expressing an operative business rule negatively in terms of prohibition rather than positively in terms of obligation~~

~~Definition:~~ ~~operative business rule statement that is expressed in a prohibitive form, regardless of language, such as 'It is prohibited that …' or using the words 'must not'~~

Definition: operative business rule statement that is expressed negatively in terms of prohibition rather than positively in terms of obligation

Necessity: No ~~prohibitive~~ prohibition statement *is* a restricted ~~permissive~~ permission statement.

Note: ~~The same rule could also be expressed in another form, such as an obligatory form, by introducing or removing negation.~~

~~Example:~~ ~~"It is prohibited that a rental is open if a driver of the rental is a barred driver."~~

~~Example:~~ ~~"A rental must not be open if a driver of the rental is a barred driver."~~

Example: "It is prohibited that the duration of a rental be more than 90 rental days."

Example: "The duration of a rental must not be more than 90 rental days."

restricted permission statement

~~restricted permissive permission statement~~

~~Definition:~~ ~~operative business rule statement that is expressed as permission being granted only when a given condition is met in a syntactical form for expressing an operative business rule in terms of permission, providing the condition is met~~

~~Definition:~~ ~~operative business rule statement that is expressed in a permissive form, but with a condition that must be satisfied, regardless of language, such as "It is permitted that … only if …" or using the word "may" but subjected to "only if" or "only when"~~

Definition: operative business rule statement that is expressed as permission being granted only when a given condition is met

~~Note:~~ ~~The same rule could also be expressed in another form, such as an obligatory form, by introducing or removing negation.~~

Example: "It is permitted that a rental is open only if an estimated rental charge is provisionally charged to the credit card of the renter of the rental."

Example: "A rental may be open only if an estimated rental charge is provisionally charged to the credit card of the renter of the rental."

Note: A restricted permission statement should not be confused with a statement of advice of permission. The latter should never contain 'only', which is always interpreted as eliminating or diminishing a degree of freedom (i.e., indicating the presence of a rule). This inclusion of 'only' is the key characteristic of restricted permission statements.

Note: Every restricted permission statement can be rephrased as a conditional prohibition statement. The pattern "it is permitted that *p* only if *q*" can be stated equivalently as "it is prohibited that *p* if not *q*" or "it is not permitted that *p* if not *q*" (refer to Clause 10). For example, the following three statements mean the same thing:

1. "It is permitted that a rental is open only if an estimated rental charge is provisionally charged to the credit card of the renter of the rental."
2. "It is prohibited that a rental is open if an estimated rental charge is not provisionally charged to the credit card of the renter of the rental."
3. "It is not permitted that a rental is open if an estimated rental charge is not provisionally charged to the credit card of the renter of the rental."

12.2.2.2 Business Statements of Structural Rules

necessity statement

~~necessity business rule statement~~

~~Definition:~~ ~~structural business rule statement that is expressed negatively in terms of impossibility rather than positively in a syntactical form for expressing a business rule statement positively in terms of necessity rather than negatively in terms of impossibility~~

~~Definition:~~	~~business rule statement that is expressed in a form indicating a necessity, regardless of language, such as 'It is necessary that …'~~
Definition:	structural rule statement that is expressed positively in terms of necessity rather than negatively in terms of impossibility
~~Necessity:~~	~~No necessity business rule impossibility statement *is* a restricted possibility impossibility business rule statement.~~
~~Necessity:~~	~~No necessity business rule statement *is* a restricted possibility business rule statement.~~
Necessity:	No necessity statement *is* an impossibility statement
Necessity:	No necessity statement *is* a restricted possibility statement
Example:	"It is necessary that each rental has exactly one requested car group."
Example:	"Each rental always has exactly one requested car group."

impossibility statement

~~impossibility business rule statement~~

~~Definition:~~	~~structural business rule statement that is expressed in a syntactical form for expressing a business rule statement negatively in terms of impossibility rather than positively in terms of necessity~~
~~Definition:~~	~~business rule statement that is expressed in a form indicating impossibility, regardless of language, such as 'It is impossible that …'~~
Definition:	structural rule statement that is expressed negatively in terms of impossibility rather than positively in terms of necessity
Necessity:	No impossibility ~~business rule~~ statement *is* a restricted possibility ~~business rule~~ statement.
~~Example:~~	~~"It is impossible that the pick-up branch of a one-way rental is the return branch of that rental."~~
~~Example:~~	~~"The pick-up branch of a one-way rental is never the return branch of that rental."~~
Example:	"It is impossible that the same rental car is owned by more than one branch."
Example:	"The same rental car is never owned by more than one branch."

restricted possibility statement

~~restricted possibility business rule statement~~

~~Definition:~~	~~structural business rule statement that is expressed in a syntactical form for expressing a business rule statement in terms of possibility, providing as possibility being acknowledged only when a given condition is met~~
~~Definition:~~	~~structural business rule statement that is expressed in a form indicating possibility, but with a condition that must be satisfied, regardless of language, such as 'It is possible that … only if …'~~
Definition:	structural rule statement that is expressed as possibility being acknowledged only when a given condition is met
Example:	"It is possible that a rental is an open rental only if the rental car of the rental has been picked up."
Example:	"A rental can be an open rental only if the rental car of the rental has been picked up."
Note:	A restricted possibility statement should not be confused with a statement of advice of possibility. The latter should never contain 'only', which is always interpreted as eliminating

or diminishing a degree of freedom (i.e., indicating the presence of a rule). This inclusion of 'only' is the key characteristic of restricted possibility statements.

Note: Every restricted possibility statement can be rephrased as a conditional impossibility statement. The pattern "it is possible that *p* only if *q*" can be stated equivalently as "it is impossible that *p* if not *q*" or "it is not possible that *p* if not *q*" (refer to Clause 10). For example, the following three statements mean the same thing:

1. "It is possible that a rental is an open rental only if the rental car of the rental has been picked up."
2. "It is impossible that a rental is an open rental if the rental car of the rental has not been picked up."
3. "It is not possible that a rental is an open rental if the rental car of the rental has not been picked up."

12.2.2.3 Business Statements of Permission

permission statement

Definition: statement of advice of permission that is expressed positively in terms of permission rather than negatively in terms of non-obligation

Necessity: No permission statement *is* a non-obligation statement.

Example: "It is permitted that the drop-off branch of a rental is not the return branch of the rental."

non-obligation statement

Definition: statement of advice of permission that is expressed negatively in terms of non-obligation rather than positively in terms of permission

Example: "It is not obligatory that the drop-off branch of a rental be the return branch of the rental."

Example: "The drop-off branch of a rental need not be the return branch of the rental."

optionality statement

Definition: statement of advice of permission that *expresses* an advice of optionality

Note: An optionality statement may take various forms, each expressing the meaning of the same advice of optionality, as illustrated by the following examples.

Example: "It is neither prohibited nor obligatory that the renter shows photo identification at the pick-up time of a rental."

Example: "It is permitted but not obligatory that the renter shows picture identification at the pick-up time of the rental."

12.2.2.4 Business Statements of Possibility

possibility statement

Definition: statement of advice of possibility that is expressed positively in terms of possibility rather than negatively in terms of non-necessity

Necessity: No possibility statement *is* a non-necessity statement.

Example: "It is possible that the notification date/time of a bad experience that occurs during a rental is after the actual return date/time of the rental."

Example: "The notification date/time of a bad experience that occurs during a rental can be after the actual return date/time of the rental."

non-necessity statement

Definition: statement of advice of possibility that is expressed negatively in terms of non-necessity rather than positively in terms of possibility

Example: "It is not necessary that the notification date/time of a bad experience that occurs during a rental be on or before the actual return date/time of the rental."

contingency statement

Definition: statement of advice of possibility that *expresses* an advice of contingency

Note: A contingency statement may take various forms, each expressing the meaning of the same advice of contingency, as illustrated by the following examples.

Example: "It is possible but not necessary that a renter's age is less than 21 years."

Example: "It is neither impossible nor necessary that a renter's age is less than 21 years."

12.3 Fundamental Principles for Elements of Guidance

12.3.1 The Severability Principle

Principle: The meaning of an element of guidance may be expressed separately from any other element of guidance; nonetheless, a body of shared guidance that includes the element of guidance will be evaluated as if all the elements of guidance had been expressed jointly and all had to hold true.

In everyday business, elements of guidance are individual elements of meaning that exist separately. Often, they are also expressed separately – e.g., by individual sentences. In a body of shared guidance of any size, such separate expression of dissimilar or disjoint elements of guidance is a practical necessity for readability and manageability.

In SBVR, a body of shared guidance is nonetheless logically considered as a whole. In other words, each element of guidance is always applied in all situations where that element of guidance is relevant – even if expressed separately. This is true even if the element of guidance is expressed without direct reference to related elements of guidance that are relevant for the same situation.

This fundamental understanding is called the *Severability Principle*.[1]

The MWUD definition of "severable" is:

> capable of being severed … ; especially : capable of being divided into legally independent rights or obligations used of a statute or contract of which the part to be performed consists of distinct items to which the consideration may be apportioned so that the invalidity or failure of performance as to one item does not necessarily affect the others

This captures the sense of what SBVR means by 'severable'. If one element of guidance is invalidated or violated somehow, the rest still apply.

It should be noted that expressing elements of guidance separately and without reference to related elements of guidance may increase the chance of conflicts, but does not create it per se. Even a single element of guidance can have internal conflicts. Conflicts must be resolved by proper specification, including cases where exceptions are intended, as discussed in 12.4.

---

1. This SBVR principle is the business counterpart to what in propositional logic is often called the *universal 'and'*. This assumption requires that all separate Propositions be true (for a body of shared guidance). Therefore, an implicit '*and*' must be considered to exist between all such Propositions.

It should also be noted that the *Severability Principle* does not apply across separate bodies of shared guidance. Therefore conflicts and exceptions, as discussed in 12.4, can only exist within a single body of shared guidance. They cannot exist across two or more bodies of shared guidance.

12.3.2 The Accommodation Principle

Principle: An element of guidance whose meaning conflicts with some other element(s) of guidance must be taken that way; if no conflict is intended, the element(s) of guidance must be expressed in such a way as to avoid the conflict.

Exceptions to elements of guidance must be accommodated explicitly; that is, cases where exceptions to elements of guidance are intended must be worded in such a way to avoid any conflict in the meanings.

In SBVR, statements can mean only what the actual words presented in the statements indicate they mean. Therefore, to indicate that an exception is intended always requires additional or alternative specification (i.e., *accommodation*). Otherwise the meanings of the statements would simply (and necessarily) be taken to be in conflict.

12.3.3 The Wholeness Principle

Principle: An element of guidance means only exactly what it says, so it must say everything it means.

Each element of guidance must be self-contained; that is, no need to appeal to any other element(s) of guidance should ever arise in understanding the full meaning of a given element of guidance.

The full impact of an element of guidance for a body of shared guidance, of course, cannot be understood in isolation. For example, an element of guidance might be in conflict with another element of guidance, or act as an authorization in the body of shared guidance. The *Wholeness Principle* simply means that if a body of shared guidance is deemed free of conflicts, then with respect to guidance, the full *meaning* of each element of guidance does not require examination of any other element of guidance. In other words, each element of guidance can be taken at face value for whatever it says.

12.4 Accommodations, Exceptions and Authorizations

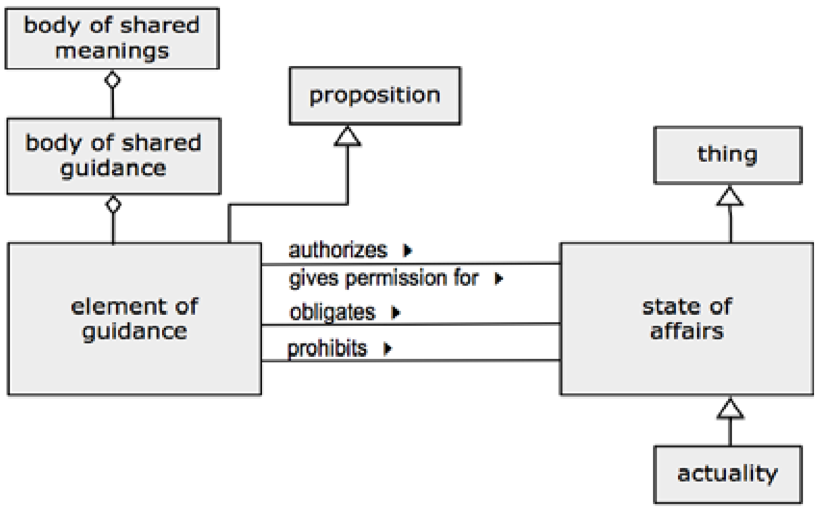


Figure 12.3

This diagram shows the SBVR Metamodel and SBVR vocabulary by two different interpretations. See Clause 13 and Annex H.

12.4.1 Relating Elements of Guidance to States of Affairs

element of guidance *authorizes* state of affairs

Definition: the element of guidance entails that the state of affairs may *be* an actuality

Synonymous Form: element of guidance *gives permission for* state of affairs

element of guidance *obligates* state of affairs

Definition: the element of guidance entails that the state of affairs must *be* an actuality

element of guidance *prohibits* state of affairs

Definition: the element of guidance entails that the state of affairs must not *be* an actuality

12.4.2 Authorizations

SBVR makes a 'light world'[2] assumption about rules. In a *light world*, anything that is not expressly prohibited is assumed permitted, and anything not expressly declared as impossible is assumed possible. Business rule practice indicates that this choice is the appropriate one for the large majority of business problems.

2. Ronald G. Ross, "The Light World vs. the Dark World ~ Business Rules for Authorization," Business Rules Journal, Vol. 5, No. 8 (August 2004), URL: http://www.BRCommunity.com/a2004/b201.html

Occasionally, practitioners may discover ‘dark areas in a light world’ – areas in which the opposite assumption is appropriate. In such a *dark area*, anything not expressly permitted is assumed prohibited, or anything not expressly declared as possible is assumed impossible. Dark areas of the former kind – the more important and common of the two cases – might involve use of, and/or access to, resources that are deemed especially sensitive, dangerous, scarce, and/or valuable. For that reason, it makes sense to grant permission for use and/or access explicitly. Such permissions are often called ‘authorizations’.

In everyday business language, an *authorization* is generally understood to mean a sanction or a warrant [MWUD].

> [MWUD “sanction” noun]: 6a. explicit permission or recognition by one in authority that gives validity to the act of another person or body
>
> [MWUD “warrant” noun]: 2a. a commission or document giving authority to do something : an act, instrument, or obligation by which one person authorizes another to do something which he has not otherwise a right to do and thus secures him from loss or damage

For SBVR, it is important to note that an authorization is *explicit* (from “sanction”), and that without it, there is *not otherwise a right to do something* (from “warrant”).

12.4.3 Exceptions

Authorizations fall under the more general topic of *exception*. In everyday business language, to ‘make an exception’ is generally understood to mean [MWUD “exception” 1] “the act of excepting or excluding: exclusion or restriction (as of a class, statement, or rule) by taking out something that would otherwise be included.” An ‘exception’ is what is omitted from consideration.

In SBVR, the *Severability Principle* permits elements of guidance to be given separately (individually), raising the possibility that one element of guidance might actually be intended as an exception with respect to another. The general element of guidance and its exceptions are always in the same body of shared guidance.

SBVR’s approach to exceptions, which includes authorizations, is based on the fundamental principles for elements of guidance given in Subclause 12.3. The following describes how exceptions and authorizations may be specified in SBVR.

12.4.4 Approaches to Capturing Accommodations, Exceptions and Authorizations

Approach 1 – General Elements of Guidance that Accommodate More Specific Cases

This approach uses the fact types specified above (in 12.4.1) to allow for more specific cases to be specified for some more general element of guidance. This discussion will use the ‘element of guidance *authorizes* state of affairs’ fact type, but it should be noted that the other two fact types would be applied similarly, as appropriate to the business situation.

A state of affairs being ‘authorized’ means that some specific element of guidance in a body of shared guidance entails that the state of affairs may validly occur, i.e., is not an error or conflict with the more general rule. Support for exceptions (and authorizations) in this approach is accomplished as follows.

- An operative business rule is specified to declare that some given area of business activity is prohibited except where there is some explicit advice of permission given (i.e., a ‘dark’ area is declared).
- Explicit advice(s) of permission, qualified as appropriate, are specified to declare selective exceptions/authorizations. Without such permissions, there would otherwise be no right to do something.

In general, a *logical OR* is always assumed between the more specific cases given separately from the more general element of guidance. The body of shared guidance can contain any number of ‘exceptions’ to general cases without introducing conflicts as long as the general case element of guidance allows for exceptions.

The two Examples illustrate different subjects for authorization. The first authorizes an action (use of a vehicle on an ice road) under given conditions, whereas the second authorizes people to carry out an action (making a payment).

EXAMPLE

Two guidance statements, expressing a general rule and a more specific case for EU-Rent:

Vehicle Usage Rule

A vehicle may *use* an ice road only if the use *is authorized by* a Vehicle Usage Advice.

Arctic Circle Exemption

Any ice road that *is north of the Arctic Circle* may *be used by* any vehicle.

The Arctic Circle Exemption *is* a Vehicle Usage Advice.

These elements of guidance work together like this:

The first element (an operative business rule) sets up the *dark area*, prohibiting any use that is not explicitly authorized. It does this by use of the fact type ‘element of guidance *authorizes* state of affairs’.

The second element is one of perhaps many Vehicle Usage Advices. The concept ‘Vehicle Usage Advice’ is a category of advices within EU-Rent’s body of shared guidance.

Note that this Example assumes the standard SBVR constructs have been used, e.g., ‘vehicle’ and ‘ice road’ are assumed to be defined terms; as well as the fact type (vehicle *uses* ice road) being defined and objectified as ‘use’. For simplicity, ‘being north of the Arctic Circle’ is taken to be a characteristic of an ice road, but other, more elaborate solutions could have been worked out.

EXAMPLE

Three guidance statements, expressing a general case and two more specific cases, with facts that classify the specific cases and connect them to the general case:

Guidance Statements:

Payments Business Rule

A person may *make* a payment *only if a* Payment Authorization *authorizes* that the person *make* the payment.

Senior Manager Exemption

Any senior manager may *make* any payment.

Jane Smith may *make* any payment.

Facts:

The Senior Manager Exemption *is* a Payment Authorization.

"Jane Smith may *make* any payment" *is* a Payment Authorization.

The first element (an operative business rule) sets up the *dark area*, prohibiting any payment that is not explicitly authorized. The fact type used is 'element of guidance *authorizes* state of affairs'.

The second element is a blanket advice of permission that allows any person who is a senior manager to make a payment. The third element stipulates that a specific person (Jane Smith) may make payments.

This Example assumes the defined fact type 'person *makes* payment'. It also assumes that the terms used are defined (e.g., person, payment) and that Jane Smith is a known person (and no assumption beyond that is made about her). The two facts classify the second and third elements as 'Payment Authorizations', a category of advices of permission in the body of shared guidance, and thus relate them to the general case, in which 'Payment Authorization' plays a role.

Regarding any person and payment, the *exception condition* of the rule statement is that the person be explicitly permitted to make the payment, either directly (as in the case of Jane Smith) or indirectly (as in the case of any senior manager). The advice of permission statements express, for certain persons and any payment, that a person is permitted to make the payment. It can be determined, for every instance of the fact type 'person *makes* payment', that the condition is satisfied. As long as a person satisfies either *exception condition* of the rule, that person is permitted to make any payment – i.e., that he or she has 'authorization'.

Approach 2 – Using a Business Concept

Another acceptable approach, illustrated below by a reworking of the second Example given for Approach 1, is that the business has some concept(s) to help express authorizations.

EXAMPLE

Consider the following rule and supporting statements that use the concept ‘authorized payer’, which has been defined as “person that may *make* any payment.”

Rule Statement: Only an authorized payer may *make* a payment.

Specification of Authorized Payers:

- Each senior manager *is* an authorized payer.
- Jane Smith *is* an authorized payer.

Given the definition of ‘authorized payer’, these two statements meet the same business requirement as the advice statements in the second Example given for Approach 1 – that senior managers and Jane Smith may make any payment. Regardless of the definition of ‘authorized payer’, these two statements clearly satisfy the condition of the rule statement by identifying instances of ‘authorized payer’, which is the concept considered by the condition in the rule.

Approach 3 – Formulating Elements of Guidance to Avoid Exceptions

A third approach is to simply specify a set of elements of guidance whose conditions are mutually-exclusive.

EXAMPLE

Two rules, expressed as individual statements with mutually-exclusive conditions:

1. The state sales tax must *be charged on* each order *shipped within the state*
2. The state sales tax must not *be charged on* an order *shipped out-of-state*.

Note that the second rule above would not be considered to be “an exception” to the first. Rather, its expression includes “out-of-state” to differentiate it from “orders shipped “within the state.” This accommodation avoids a collision between the meanings of the rules that would otherwise arise.

Issue 10575 Add new Clause

12.5 Relating Structural Rules to Concepts

Structural rules often, but not always, propose necessary characteristics of concepts. Here are three cases:

1. A structural rule uses universal quantification (e.g., “each” or “all”) to propose a necessary characteristic of a concept. The structural rule proposes that something is always true about all instances of the concept.
2. A structural rule proposes a necessary characteristic of an individual concept - no universal quantification is used because it is implicit in referring to the one and only instance of the individual concept.
3. Cases other than 1 and 2 above: a structural rule does not propose a necessary characteristic of a concept, but it proposes something to be necessarily true. See Rule 4 in the examples below.

A fact that a concept has a necessary characteristic is a structural rule that the characteristic is always true about each instance of the concept. How is it a structural rule? It is a proposition that the necessary characteristic is always true of each instance of the concept. Conversely, a structural rule proposes that a characteristic is a necessary characteristic of a concept if and only if the structural rule proposes that the characteristic is always true about each instance of the concept. The structural rule does not imply that the concept incorporates the characteristic, because necessary characteristics can be either incorporated or implied.

There is a logical connection between concepts and structural rules. A starting point of the logical connection is these two necessary truths about concepts:

1. For each concept, each characteristic it incorporates is attributed to each instance of the concept.
2. For each individual concept, the instance of the individual concept exists.

From this starting point, considering concepts together, there are any number of propositions can be proved to be true by logical implication. A structural rule is logically connected to concepts when it proposes that one of these propositions is necessarily true. Structural rule statements often facilitate a deeper understanding of concepts, but a structural rule never changes a concept. Rather, it proposes what logically follows from an understanding of concepts, and in some cases, from business decisions that define specific thresholds.

In cases where definitions of concepts taken together do not logically imply something proposed in a structural rule statement, there is an inadequacy or mistake in either the relevant definitions or in the rule statement. The case of inadequate definitions is common and is acceptable in some communities. It occurs when a community shares a tacit understanding of many of its concepts. Words either have no explicit definitions or have definitions that use words that have no explicit definitions. Structural rule statements in this context can be correct, even if they logically follow from a tacit understanding of what characteristics are incorporated by concepts.

Practices of developing concept systems range from creating highly precise, rigorously complete definitions for all concepts to creating no or few definitions, or largely descriptive or informal ones, but many structural rules. Where highly precise, rigorously complete definitions are given there is less need for structural rules because such rules would appear redundant. Where definitions are missing or unclear, or largely descriptive or informal, structural rules are important to sharing a common understanding of concepts.

Advices of possibility relate to concepts following the same pattern by which structural rules relate to concepts.

Where there is a definition, a concept is just what the definition says, no more and no less. Something called a “definition” as used in common speech is not necessarily a definition as defined by SBVR. It might be just a general description. It is only a definition if it defines the concept, differentiating it from others. As a matter of practice, a simple test for adequacy and correctness of definitions is to restate a rule by substituting a definition of a concept into a rule statement in place of the concept's designation. Does the restatement express the same meaning as the original statement? If not, the so-called definition is inadequate or incorrect. Consider the example below:

sports car
Definition: kind of car

Rule 1: A rental of a sports car must include collision coverage.

A restatement of Rule 1, “A rental of a kind of car must include collision coverage,” expresses a different meaning, so the definition is inadequate. Here is an adequate definition:

sports car
Definition: small, fast automobile equipped for racing

When the adequate definition is substituted into a restatement of the rule, the same rule is expressed. Consider some examples of structural rules related to 'sports car'.

Rule 2: Each sports car is always small.

Rule 2 expresses a characteristic attributed to all sports cars by the definition of 'sports car'. It is an incorporated characteristic of 'sports car'.

Rule 3: Each Corvette is always a sports car.

Rule 3 does not change the meaning of 'sports car'. Rather, it expresses an understanding that every Corvette is a small, fast automobile equipped for racing. This understanding is found in the meaning of Corvette. Agreement on this understanding might come from analysis of a definition of 'Corvette', or it might be established by a business decision about meaning based on tacit knowledge. Structural rules expressing such business decisions are often important guides to business knowledge.

EU-Rent Speedway
Definition: the test track owned by EU-Rent where any small car is testable

Rule 4: A test track always exists.

Rule 4 follows logically from the individual concept 'EU-Rent Speedway'. An individual concept always has one instance. So there is always an EU-Rent Speedway, and therefore, a test track.

Rule 5: The EU-Rent Speedway is always in Germany.

Rule 5 does not appear to follow logically from an understanding of definitions. It might well be true that the EU-Rent Speedway is in Germany, but Rule 5 proposes that it is always true - true in all possible worlds. Structural rules are about what is true in all possible worlds, so a statement of a fact, not a rule, is more appropriate here:

Fact 6: The EU-Rent Speedway is in Germany.

Rule 7: Every sports car is always testable at the EU-Rent Speedway.

Finally, Rule 7 proposes a necessary characteristic of the concept 'sports car'. This characteristic is an implied characteristic because it is not an incorporated characteristic of 'sports car'. It follows logically from the combination of characteristics of 'sports car' and 'EU-Rent Speedway'.

Issue 9930 Replace entire clause

13 SBVR’s Use of MOF and XMI

The SBVR Metamodel is a MOF-based metamodel that supports a MOF representation of the concepts represented by the SBVR vocabularies. The UML figures in clauses 8, 9, 11, and 12 show the SBVR vocabulary and the SBVR Metamodel at the same time. This is because the vocabulary used by people and the MOF-based metamodel reveal the same concept system. Conceptual integration across vocabularies and languages involves one set of concepts (one model) expressed using different vocabularies or different languages.

SBVR’s use of MOF and how the SBVR Metamodel handles certain semantic modeling challenges using MOF 2.0 are described below. The SBVR Metamodel is available as an XML document (see “SBVR Metamodel” on page 277) It is drawn from the text of clauses 8, 9, 11 and 12. UML Figures in those clauses illustrate the Metamodel using an interpretation explained in 13.1 below. This interpretation should not be confused with the 'Business Object Model' interpretation of the same figures explained in Annex H, which is based on a different profile. An example model that instantiates the SBVR Metamodel is then shown and explained. Finally, the MOF-based SBVR model of SBVR is explained.

Models of business concepts, business vocabularies and business guidance can be communicated in terms of SBVR using XML documents that conform to an XMI-based XML schema created from the SBVR Metamodel (see “SBVR Metamodel” on page 277).

13.1 SBVR's Use of MOF

The following terms used in this clause are not words defined by SBVR. Their meanings come from MOF 2.0.

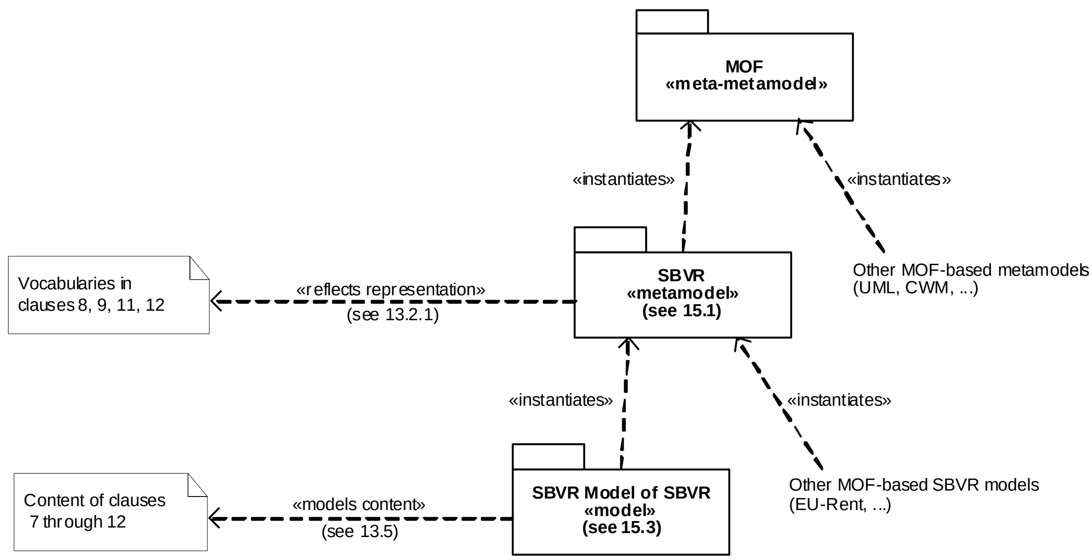
metamodel	package	association	association end	class	attribute	data type
model		link		element		data value

How each of these is used with respect to SBVR is explained below.

The UML figures in clauses 8 through 11 use normal UML notation to show the SBVR Metamodel except for custom notations described below.

13.1.1 Metamodels

A model is a representation of facts. A model instantiates a metamodel which describes the structure and language by which facts are represented in models. A metamodel is itself a model which instantiates the MOF model (the meta-metamodel). The diagram below illustrates how SBVR fits into the MOF metamodeling architecture.



The SBVR Metamodel instantiates the MOF model. It describes MOF-based SBVR models, which represent facts built on SBVR concepts represented in these vocabularies:

Meaning And Representation Vocabulary

Logical Formulation of Semantics Vocabulary

Vocabulary for Describing Business Vocabularies

Vocabulary for Describing Business Rules

The combination of these vocabularies is the SBVR Vocabulary.

The SBVR Metamodel does not include definitions, rules, notes, examples or semantic formulations. Rather, it mirrors the SBVR namespaces for those vocabularies. It provides a MOF means of expression (classes and associations) where the SBVR vocabulary namespaces identify an English language means of expression (designations and fact type forms). Both use the same signifiers. A result of this alignment of the SBVR Metamodel with the SBVR vocabulary is that knowledge of the vocabulary implies knowledge of the Metamodel and vice versa. The SBVR Metamodel is serialized as an XML document (see “SBVR Metamodel” on page 277).

13.1.2 MOF-based SBVR Models

MOF-based SBVR models represent facts that are about or within a body of shared meanings. For example, facts about EU-Rent's concepts, rules, their representations and their semantic formulations can be represented in a MOF-based SBVR model. A thing represented in a model is identified by facts about the thing that satisfy a reference scheme. An example MOF-based SBVR model is shown in 13.4 below. MOF-based SBVR models are often incomplete representations of a body of shared meanings. The size of a model depends on what facts are being represented, which can be as little as a single fact.

One particular MOF-based SBVR model is the MOF-based SBVR model of SBVR, which is a model of SBVR in terms of itself. It is described in 13.5 below.

Issue 9930 Revise text format

A MOF-based SBVR Model instantiates the SBVR Metamodel. It represents a fact model, which combines a conceptual schema and a set of facts. The conceptual schema is described by the SBVR model of SBVR. The facts are expressed in terms of the concepts in the conceptual schema and are limited to what is possible according to the conceptual schema.

13.2 MOF Model Elements for SBVR

Issue 9930 Revise text format

The SBVR Vocabulary is mapped to MOF elements that make up the SBVR Metamodel. It should not be construed from this one-way mapping that a MOF class is the same thing as an SBVR concept or that there is any semantic equivalence between MOF and SBVR.

SBVR model content is represented in MOF-based SBVR models according to the SBVR Metamodel. MOF-based SBVR models instantiate the SBVR Metamodel, not the UML Metamodel. Another transform would be needed to represent SBVR model content using UML.

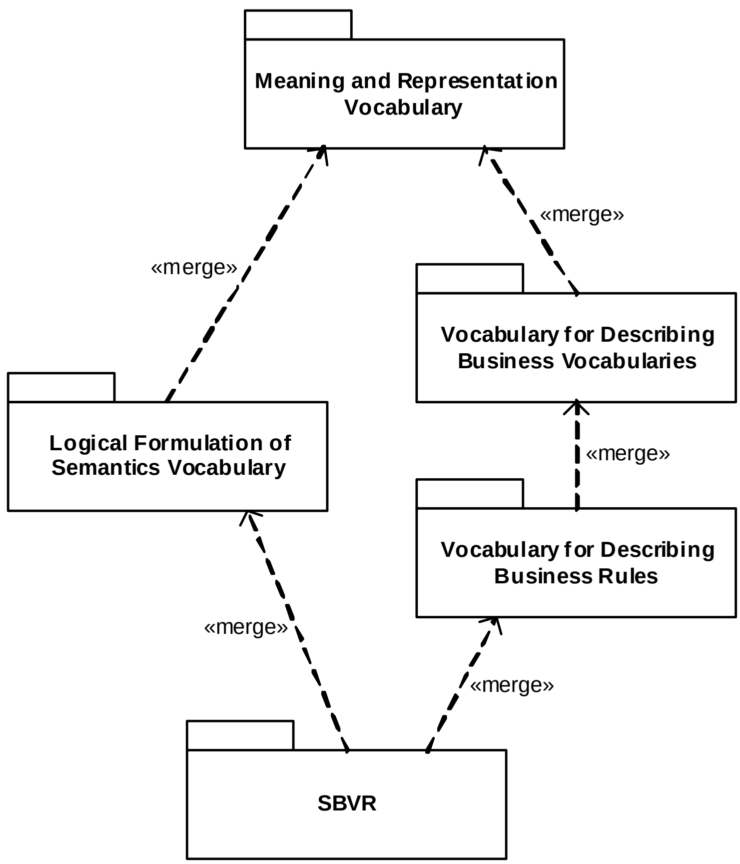
Both the mapping of the SBVR Vocabulary to MOF and the representation of SBVR model content using MOF are described below, divided using the following headings.

Heading	Purpose
MOF Elements of the SBVR Metamodel	Prescriptive description of the mapping of the SBVR Vocabulary into a MOF-based metamodel
Elements of MOF-based SBVR Models	Prescriptive description of how facts are represented within a MOF-based SBVR model
Rationale	Design rationale explaininng aspects of SBVR or MOF that led to the MOF representations described here

13.2.1 MOF Packages for SBVR Vocabulary Namespaces

MOF Elements of the SBVR Metamodel

The SBVR Vocabulary is mapped to the SBVR Metamodel, which is made up of multiple packages shown in the diagram below. Each package is a MOF-based reflection of one of SBVR's vocabulary namespaces.



The merge relationships between the packages exactly reflects the include relationships between the corresponding SBVR vocabularies.

Issue 9950 Add text

Elements of MOF-based SBVR Models

The packages that make up the SBVR Metamodel contain classes and associations. The elements of MOF-based SBVR Models are elements of those classes and associations.

Rationale

Each of the packages merged into the SBVR package can serve as a metamodel in its own right as a subset of the overall SBVR Metamodel. These packages correspond with compliance points described in Clause 2.

SBVR Metamodel packages can be imported or merged into other MOF-based metamodels. For example, a metamodel of organizational structure can import SBVR's 'Meaning and Representation Vocabulary' package as a starting point for modeling organization types and organizational roles. Similarly, a metamodel of business process can import SBVR's 'Vocabulary for Describing Business Rules' package in order to relate processes to rules and can import SBVR's 'Logical Formulation of Semantics Vocabulary' package for modeling semantic formulations of rules that govern processes. Such rules can use concepts from the metamodel of business process (e.g., 'process') if those concepts are also modeled using elements of classes in the SBVR Metamodel packages (e.g., an element of the class 'noun concept' for the concept 'process'). Also, other metamodels can import individual model elements from SBVR in cases where a portion of SBVR smaller than a package is wanted. Importing from SBVR is appropriate *only when using SBVR concepts as defined by SBVR.*

13.2.2 MOF Classes for SBVR Noun Concepts

MOF Elements of the SBVR Metamodel

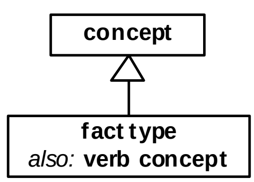
Each designation in a vocabulary namespace for a noun concept that is not a role is mirrored in the SBVR Metamodel as a class. The signifier of the designation is the name of the class. The signifier of each synonym of the designation is an alias for the class.

The metamodel includes generalizations between classes reflecting generalizations between the represented noun concepts. Each SBVR concept besides 'thing' specializes 'thing', so the classes have the class '**thing**' as a superclass either directly or indirectly.

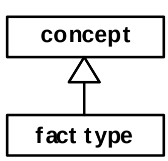
Example Vocabulary:

fact type
General Concept: concept
Synonym: verb concept

Figure:



SBVR Metamodel:



{***element import*** fact type ***as*** verb concept}

Elements of MOF-based SBVR Models

Where a class represents a noun concept, an element (in a MOF-based SBVR model) that instantiates the class represents a fact that an instance of the noun concept exists. References to the element within the MOF-based SBVR model indicate references to the instance of the noun concept. Note that it is possible that two elements in a MOF-based SBVR model represent the same actual thing (13.3.1 explains situations where this is likely and tells how to relate the two elements within the MOF-based SBVR model). Also, a lack of an element in a MOF-based SBVR model implies nothing - it does not imply that something does not exist.

Rationale

Use of aliasing, though not common in MOF-based metamodels, keeps a strong alignment of the SBVR Metamodel with the SBVR vocabulary.

Some UML figures in clauses 8 through 12 show partitioning or disjoint categories using UML notation, but those features are not included in MOF 2.0, so partitioning and disjointness are not reflected in the SBVR Metamodel. Also, MOF 2.0 does not support association classes. Each case of an association class in a figure corresponds with a fact type and a noun concept, and each of the two is represented separately in the SBVR Metamodel.

13.2.3 MOF Boolean Attributes for SBVR Characteristics

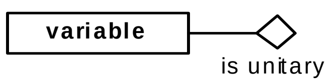
MOF Elements of the SBVR Metamodel

A characteristic is represented in MOF as an optional Boolean attribute as shown below.

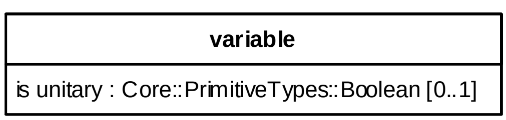
Example Vocabulary:

variable *is unitary*

Figure:



SBVR Metamodel:



Elements of MOF-based SBVR Models

For an element in a MOF-based SBVR model, the meaning of the value TRUE is that the characteristic is attributed to the thing represented by the element. A meaning of FALSE is that the thing represented by the element does not have the characteristic. A meaning of the attribute being null is the same as the attribute being unspecified for the element.

Rationale

The attribute is optional in support of the Open World Assumption, explained in 13.3.2 below.

13.2.4 MOF Associations for SBVR Binary Fact Types

MOF Elements of the SBVR Metamodel

Each binary fact type is represented in MOF terms as an association. Association names match fact type forms. If a fact type has only one fact type form, the association's name is the expression of that fact type form, but with subscripts raised to normal text. The names of the association's ends are the placeholder expressions from the fact type form. The ends are owned by the association so that individual links can be serialized using XMI.

In cases of more than one fact type form (synonymous forms), one is chosen to name the association that does not imply a designation in an attributive namespace. Then there is an alias for the association for each other fact type form that has matching placeholder expressions (which implies matching association end names).

In figures in the normative clauses, a label on an association line that includes a reading direction arro► ’) is meant to be read starting with the name of the class on the first end and ending with the name of the class on the other end, except where a name for an end is already in the label. The association names match this reading exactly. Including the names of an association's ends in the association's name makes the association's name unique within a package, as required by MOF.

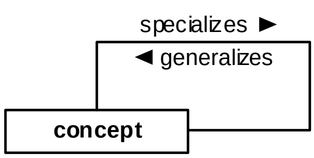
In cases where an association's ends both connect to the same class, subscripts are used on placeholders to distinguish them. In the association name and its ends' names the subscripts are raised to normal text and serve to distinguish the ends.

Example Vocabulary:

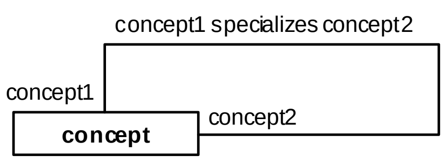
concept$_1$ *specializes* concept$_2$

Synonymous Form: concept$_2$ *generalizes* concept$_1$

Figure:



SBVR Metamodel:



{*element import* concept1 specializes concept2 ***as*** concept2 generalizes concept1**}**

Some structural rules impose multiplicity constraints for binary fact types. These are shown in the Figures in Clauses 8 through 12 and are included in the SBVR Metamodel.

Elements of MOF-based SBVR Models

Where an association represents a binary fact type, a link of the association within a MOF-based SBVR model represents a fact of that binary fact type. The absence of a link implies nothing. There are no defaults.

Rationale

Partitive fact types are shown in figures as UML shared aggregation, which is not supported by MOF 2.0. All association ends in the SBVR metamodel are noncomposite.

13.2.5 MOF Attributes for SBVR Roles of Fact Types

MOF Elements of the SBVR Metamodel

A role of a binary fact type that has a designation in an attributive namespace is understood in MOF terms as an attribute owned by the subject class. Such designations appear in figures as names on association ends. In the example below, ‘element’ is in an attributive namespace for the concept ‘set,’ so it is mirrored in the SBVR Metamodel as an attribute.

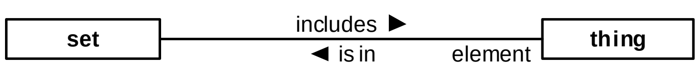
Example Vocabulary:

thing *is in* set

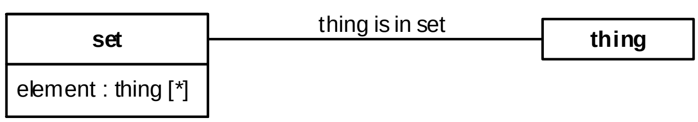
Synonymous Form: set *includes* thing

Synonymous Form: set *has* element

Figure:



SBVR Metamodel:



{*element import* thing is in set ***as*** set includes thing**}**

In each case where an attribute and an association end represent the same role, the SBVR Metamodel includes a tag that tags both the attribute and the association end. The tag connects them to show their correlation. The tag’s name is “org.omg.sbvr.sameRole”, its value is "" (the empty string) and its elements are the attribute and the association end.

Where structural rules impose multiplicity constraints, they are shown in figures and are included in the SBVR Metamodel for association ends and for attributes.

Elements of MOF-based SBVR Models

Where a role of a binary fact type is understood in MOF terms as an attribute, specification of the attribute for an element in a MOF-based SBVR model represents the entire extension of that fact type for the element. There are no defaults. If the attribute is unspecified for an element, it is simply unspecified - it is not presumed by default to have no value. If anything is specified, all values of the attribute are specified. Specification that the attribute is null means there is no instance of the fact type for the element.

Rationale

The attributes described here in 13.2.5 are in addition to the associations that represent the binary fact types - the reason for the distinction is explained in 13.3.2 below.

To preserve ‘set’ semantics, any two values of the same attribute of the same element in a MOF-based SBVR Model represent two different things. Where an attribute has two or more values, it can be concluded that each of the values represents a thing that is distinct from the others.

13.2.6 MOF Classes for SBVR Ternary Fact Types

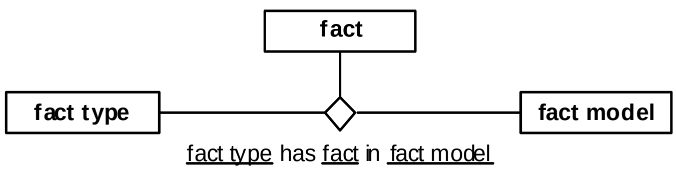
MOF Elements of the SBVR Metamodel

MOF 2.0 does not support ternary associations. Therefore, a ternary fact type is represented in MOF terms as a class with one single-valued, required attribute for each role of the fact type. The class's name takes the same form as the name of an association for a binary fact type. If there are multiple fact type forms for a ternary fact type, aliases are used.

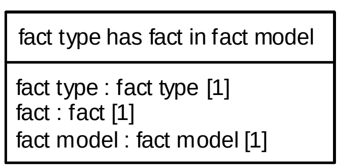
Example Vocabulary:

fact type ***has*** **fact** ***in*** **fact model**

Figure:



SBVR Metamodel:



Elements of MOF-based SBVR Models

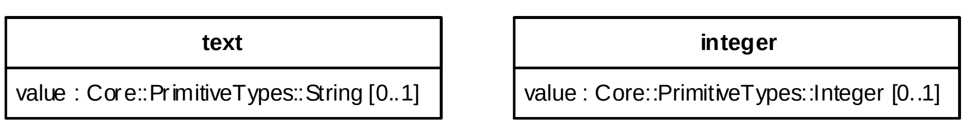
In a MOF-based SBVR model, an element of such a class represents a fact of the ternary fact type.

13.2.7 Data Values

MOF Elements of the SBVR Metamodel

The classes 'text' and 'integer', representing 'text' and 'integer', have data attributes shown below.

SBVR Metamodel:



Elements of MOF-based SBVR Models

If one of these attributes is specified in a MOF-based SBVR model, the represented text or integer is the specified value. Specification of null is equivalent to not specifying anything. There are no defaults.

Rationale

The attributes are optional because SBVR allows that texts and integers, like other kinds of things, can be described by facts without necessarily being identified. Also, the data types 'String' and 'Integer' in MOF have size limitations, so the attributes cannot be used for all cases. To refer to a string or integer that is beyond the MOF limitations, a model can identify the string or integer using facts about it that satisfy a reference scheme. For example, the number 999999999999 can be identified as having a designation in the ISO 6093 Number Namespace with the signifier "999999999999."

13.2.8 XMI Names

MOF Elements of the SBVR Metamodel

A named element is tagged with an 'org.omg.xmi.xmiName' tag if its XMI name differs from its MOF name. XMI names are determined from MOF names by upcasing each character that follows a blank and then removing the blank. The names, which come from the SBVR vocabularies, do not contain any characters that are invalid in XML identifiers.

13.3 Using MOF to Represent Semantics

The SBVR Metamodel is a direct reflection of the SBVR vocabulary, which represents SBVR meanings, but this direct representation of SBVR meanings requires two semantic modeling capabilities not directly provided by MOF 2.0. The two following clauses explain how the two capabilities, multiclassification and the Open World Assumption, are supported by the SBVR Metamodel.

13.3.1 Multiclassification

MOF 2.0 requires that each element is described by one class (its "metaClass"). Sometimes a thing cannot be represented by an element of a single class. This happens when a thing is an instance of multiple concepts, neither one specializing the other. To represent this case, multiple elements are used, one per concept. A link of the association '**thing1 is thing2**' (representing the fact type 'thing_1 *is* thing_2') is used to indicate that the multiple elements represent the same thing.

As an example, consider the noun concepts 'closed logical formulation' and 'obligation formulation'. Neither specializes the other. Where an obligation formulation is a closed formulation that formulates a proposition, a model uses one element of type '**closed logical formulation**' and a separate element of type '**obligation formulation**' along with a '**thing1 is thing2**' link that says the two elements represent the same thing.

13.3.2 Open World Assumption

The open world assumption is that representation of facts in a model does not imply that those are the only facts of a particular type nor that they are the only facts of a particular type about a subject thing - there are no implications to be taken from what is not represented in a model. For example, consider facts about a set S. The two facts, "1 is in S" and "2 is in S," do not convey the same meaning as "S = {1, 2}" because the two facts do not imply anything about whether other things are in S.

In general, models represent facts with an open world assumption. But some reference schemes use roles of binary fact types extensionally, so models represent a complete extension with respect to a subject thing being identified.

MOF supports the open world assumption about instantiation of classifiers (classes and associations). MOF's attributes support representation of an entire extension of an attribute with respect to a given subject. In order to enable a clear distinction in a model between individual facts and complete extensions with respect to a subject, association links are used to represent individual facts of a binary fact type while attributes are used when identifying a complete extension of a binary fact type with respect to a particular subject. This means that a fact can in one model be represented by a link, and in another by a value of an attribute of an element. The fact is represented using an attribute only when the complete extension of the fact type is being represented for the subject. Examples of both cases appear in the example below. SBVR has a designation in an attributive namespace for every role that is extensionally used by a reference scheme such that the SBVR Metamodel has the required attributes to satisfy all of SBVR's reference schemes.

Issue 9955 Replace text
Issue 11283 Replace URL

13.4 Example MOF-based SBVR Model

Consider the following example, which ~~is~~ includes a small portion of a vocabulary and a rule statement.

company

officer

company appoints officer

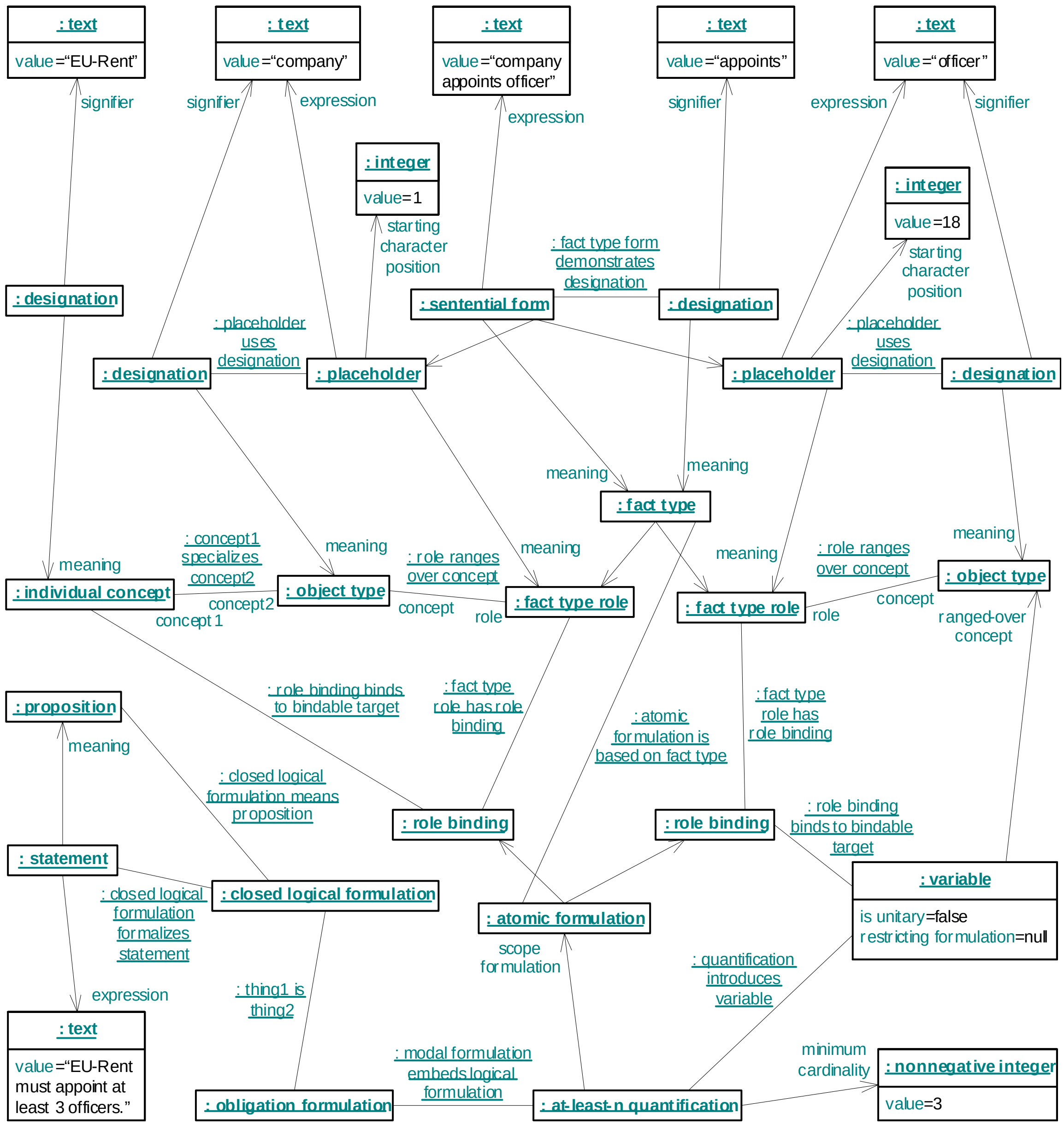
EU-Rent

General Concept:company

EU-Rent must appoint at least 3 officers.

The following figure is a UML instance diagram showing a MOF-based SBVR model of the example. For simplicity, only facts expressible in terms of the Meaning And Representation Vocabulary and the Logical Formulation of Semantics Vocabulary are shown. Some end names are elided where they are obvious from the class names or for '**thing1 is thing2**' (where it makes no difference). For elements of the vocabulary, the three layers of expression, representation, and meaning are apparent in the diagram. The rule, shown at the bottom, connects to the ~~vocabulary's~~ meanings of the elements of the vocabulary though its logical formulation.

Issue 9948 Replace figure



The example MOF-based SBVR model is expressed below in XML based on the SBVR XML Schema. The xmi:id values are arbitrary and have no special meaning, but they build on the related signifiers to help readability. The XML tags, which include the namespace prefix 'sbvr', are the XMI names for model elements of the SBVR Metamodel.

Issue 9948 Replace text, revise text

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmi:XMI xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
    xmlns:sbvr="http://www.omg.org/spec/SBVR/1.0/SBVR.xml">
```

For ‘company’:

```
<sbvr:designation xmi:id="company" signifier="company-t" meaning="company-c"/>
<~~sbvr:nounConcept~~ sbvr:objectType xmi:id="company-c"/>
<sbvr:text xmi:id="company-t" value="company"/>
```

For ‘officer’:

```
<sbvr:designation xmi:id="officer" signifier="officer-t" meaning="officer-c"/>
<~~sbvr:nounConcept~~ sbvr:objectType xmi:id="officer-c"/>
<sbvr:text xmi:id="officer-t" value="officer"/>
```

For ‘company appoints officer’:

```
<sbvr:sententialForm xmi:id="companyAppointsOfficer" expression="cao-t" meaning="cao-c" placeholder="cao-p1 cao-p2"/>
<sbvr:factType xmi:id="cao-c" role="cao-r1 cao-r2"/>
<sbvr:factTypeFormDemonstratesDesignation factTypeForm="companyAppointsOfficer" designation="appoints"/>
<sbvr:designation xmi:id="appoints" signifier="appoints-t" meaning="cao-c"/>
<sbvr:text xmi:id="cao-t" value="company appoints officer"/>
<sbvr:text xmi:id="appoints-t" value="appoints"/>

<sbvr:placeholder xmi:id="cao-p1" expression="company-t" startingCharacterPosition="i1" meaning="cao-r1"/>
<sbvr:placeholderUsesDesignation placeholder="cao-p1" designation="company"/>
<sbvr:concept1SpecializesConcept2 concept1="cao-r1" concept2="company-c"/>
<sbvr:factTypeRole xmi:id="cao-r1"/>
<sbvr:positiveInteger xmi:id="i1" value="1"/>

<sbvr:placeholder xmi:id="cao-p2" expression="officer-t" startingCharacterPosition="i18" meaning="cao-r2"/>
<sbvr:placeholderUsesDesignation placeholder="cao-p2" designation="officer"/>
<sbvr:concept1SpecializesConcept2 concept1="cao-r2" concept2="officer-c"/>
<sbvr:factTypeRole xmi:id="cao-r2"/>
<sbvr:positiveInteger xmi:id="i18" value="18"/>
```

Issue 9948 Remove XML, revise text

For ‘EU-Rent’ with “General Concept: company”:

```
<sbvr:designation xmi:id="EU-Rent" signifier="EU-Rent-t" meaning="EU-Rent-c"/>
<sbvr:individualConcept xmi:id="EU-Rent-c"/>
<sbvr:text xmi:id="EU-Rent-t" value="EU-Rent"/>
<sbvr:concept1SpecializesConcept2 concept1="EU-Rent-c" concept2="company-c"/>
```

For “EU-Rent must appoint at least 3 officers”:

```
<sbvr:statement xmi:id="stmt" expression="stmt-t" meaning="stmt-p"/>
<sbvr:text xmi:id="stmt-t" value="EU-Rent must appoint at least 3 officers."/>
<sbvr:proposition xmi:id="stmt-p"/>
<sbvr:closedLogicalFormulationFormalizesStatement closedLogicalFormulation="ob2" statement="stmt"/>
<sbvr:closedLogicalFormulationMeansProposition closedLogicalFormulation="ob2" proposition="stmt-p"/>
<sbvr:obligationFormulation xmi:id="ob"/>
<sbvr:closedLogicalFormulation xmi:id="ob2"/>
```

```
<sbvr:thing1IsThing2 thing1="ob" thing2="ob2"/>
<sbvr:modalFormulationEmbedsLogicalFormulation modalFormulation="ob" logicalFormulation="am3"/>
<sbvr:at-least-nQuantification xmi:id="am3" scopeFormulation="atom" minimumCardinality="i3"/>
<sbvr:quantificationIntroducesVariable quantification="am3" variable="v"/>
<sbvr:variable xmi:id="v" ranged-overConcept="officer-c" restrictingFormulation="" isUnitary="false"/>
<sbvr:atomicFormulation xmi:id="atom" roleBinding="bind1 bind2"/>
<sbvr:atomicFormulationIsBasedOnFactType atomicFormulation="atom" factType="cao-c"/>
<sbvr:roleBinding xmi:id="bind1"/>
<sbvr:roleBindingBindsToBindableTarget roleBinding="bind1" bindableTarget="EU-Rent-c"/>
<sbvr:factTypeRoleHasRoleBinding factTypeRole="cao-r1" roleBinding="bind1"/>
<sbvr:roleBinding xmi:id="bind2"/>
<sbvr:roleBindingBindsToBindableTarget roleBinding="bind2" bindableTarget="v"/>
<sbvr:factTypeRoleHasRoleBinding factTypeRole="cao-r2" roleBinding="bind2"/>
<sbvr:positiveInteger xmi:id="i3" value="3"/>

</xmi:XMI>
```

The example shows some of the points explained previously about MOF-based SBVR models.

- Fact Model - the entire XML content represents a fact model, which is a combination of a conceptual schema and a set of facts. The conceptual schema of the fact model is identified in the heading where it says, xmlns:sbvr="http://www.omg.org/spec/SBVR/1.0/SBVR.xml." The URL identifies a document that serializes the MOF-based SBVR model of SBVR, which describes the concepts and rules that make up the conceptual schema (see 13.4 and 15.3). The elements of the XML content represent the set of facts of the fact model.

- ~~Multiclassification - There are two occurrences of 'thing1IsThing2', each of which is used to connect pairs of elements that represent the same thing. For example, the concept 'EU-Rent' is both an individual concept and a noun concept, but neither of these kinds of concepts specia lizes the other. Therefore, the concept 'EU-Rent' is represented by 'EU-Rent-c', an element of type 'individualConcept', and by 'EU-Rent-c2', an element of type 'nounConcept'. A 'thing1IsThing2' link then indicates that 'EU-Rent-c' and 'EU-Rent-c2' represent the same thing.~~

- Multiclassification - There is an occurrence of 'thing1IsThing2' which is used to connect a pair of elements that represent the same thing. There is an element of type 'obligationFormulation' (xmi:id="ob") and another element of type 'closedLogicalFormulation' (xmi:id="ob2"). Neither type specializes the other so there is one element of each type and a 'thing1IsThing2' link indicates that the two elements represent the same thing.

- Open World Assumption - Links, rather than attributes, are always used where there is an open world assumption, such as for the fact that the individual concept 'EU-Rent' specializes the concept 'company' - there is no indication that these concepts are not involved in other specializations.

- Attributes giving Complete Extensions for a Subject - Each specification of an attribute occurs where the entire extension of the attribute is being specified for a subject thing, such as for identifying the two placeholders of the fact type form 'company appoints officer' or the two roles of the fact type. The one 'variable' in the example is serialized with "restrictingFormulation=""" representing that it has no restricting formulation. In a number of cases, attributes are unspecified because the entire extension of the attribute for an element is not being specified. For example, the attribute 'representation' is unspecified for the elements representing meanings (e.g., 'company-c' and 'officer-c' - there can be any number of representations of a meaning, and the example model does not specify them all. However, each representation has exactly one meaning, so the 'meaning' attribute is specified for each representation to identify its one meaning.

13.5 The MOF-based SBVR Model of SBVR

The MOF-based SBVR model of SBVR represents facts concerning all of the formally captioned contents of clauses 7 through 12. In general, this includes all of the information given in the SBVR specification about its concepts that can be represented in terms of the SBVR Metamodel. This includes:

- noun concepts and their designations
- fact types and their fact type forms
- specializations/generalizations
- concept types
- definitions and, where formal, their semantic formulations
- necessity statements and, where formal, their semantic formulations
- vocabularies, language, namespaces and their URIs
- notes, examples, sources, descriptions

The MOF-based SBVR model of SBVR is like the example in 13.3 above except that it is about SBVR's vocabulary and meanings, not EU-Rent's. The complete MOF-based SBVR model of SBVR is serialized as XML documents listed in 15.3. It can be used and extended by other MOF-based SBVR models that build on SBVR's concepts.

Issue 9950 Add text, revise text

13.6 XMI for the SBVR Model of SBVR

XML patterns are shown below for the various parts of vocabulary descriptions and vocabulary entries used in clauses 7 through 12. These patterns are used to create the XML documents that serialize the MOF-Based SBVR model of SBVR. Each pattern is shown for a corresponding SBVR Structured English entry – see Annex C for entry descriptions.

The XML patterns provide a normative definition of which SBVR concepts are represented by each use of SBVR Structured English in the vocabulary descriptions and entries contained in clauses 7 through 12.

The general principles used for the patterns are these: First, the facts of what is presented using SBVR Structured English are represented using XML. Second, for the objects referenced by those facts, further facts are represented to satisfy reference schemes for those objects wherever sufficient detail is given. The principles are applicable to SBVR-based communication in general. The XML files identified in Clause 13.3, which are created based on these principles following the patterns below, are examples of XML serializations of MOF-based SBVR models.

The xmi:id values used in the patterns below are replaced by different values in the actual XML documents because the multitude of repetitions of the patterns need their own unique xmi:id values. But the xmi:id values shown below consistently and correctly show relationships within the patterns. Most xmi:id values are referenced only locally within the XML elements for the same Structured English entry, but some are referenced beyond that scope and are shown in bold blue (e.g., "**vocabulary**") so that references to them are easily followed. The different types of vocabulary entries (term, name and fact type form) are mutually exclusive. They each introduce an xmi:id value "**meaning**" which is referenced in other patterns.

Made-up names (e.g., "Xyz Vocabulary"), terms (e.g., "example term") and fact type forms (e.g., "example *is seen*") are used to show the patterns and to show how signifiers and other expressions appear in XML. Certain assumptions are made by the patterns based on the way the vocabularies in clauses 7 through 12 are interrelated. The patterns assume that a vocabulary being described has a name in the Vocabulary Registration Vocabulary (of clause 7). The patterns assume that where a term or name is used with a formal interpretation in Structured English, that term or name is found by way of the vocabulary namespace derived from the vocabulary being described. These assumptions are correct regarding clauses 7 through 12, but they cannot necessarily be assumed about all vocabulary descriptions.

Each pattern has a part that remains unchanged for the kind of entry or caption shown (except for differences in xmi:id values as described above) and a part that varies based on the content of the entry. The part that varies is shown in ***bold italics***. It can be a text or integer value, a quoted xmi:id of an object introduced elsewhere, or an XML tag.

The final XML documents created from the vocabulary clauses can differ slightly from what is exactly produced from the templates, but the represented meaning does not differ. In cases where two objects are created and then connected by a 'thing1IsThing2' link, the objects can be combined into one if they are of the same class or if one class specializes the other. In cases where the patterns would create two identical XML elements, only one is actually created. For example, all uses of an element for the integer 1 can use the same element.

13.6.1 XML Patterns for Vocabularies

Xyz Vocabulary

```
<sbvr:vocabulary xmi:id="vocabulary"/>
<sbvr:thingHasName thing="vocabulary" name="XyzVocabulary"/>
<sbvr:name xmi:id="XyzVocabulary" signifier="v-s" meaning="vocabulary-concept"/>
<sbvr:individualConcept xmi:id="vocabulary-concept" instance="vocabulary"/>
<sbvr:text xmi:id="v-s" value="Xyz Vocabulary"/>
<sbvr:designationIsInNamespace designation="XyzVocabulary" namespace="vocabularyRegistrationNamespace"/>
<sbvr:vocabularyNamespace xmi:id="vocabularyNamespace"/>
<sbvr:vocabularyNamespaceIsDerivedFromVocabulary vocabularyNamespace="vocabularyNamespace" vocabulary="vocabulary"/>
```

The pattern above assumes the Vocabulary Registration Vocabulary has a vocabulary namespace like this:

```
<sbvr:vocabularyNamespace xmi:id="vocabularyRegistrationNamespace"/>
```

Included Vocabulary: Abc Vocabulary

```
<sbvr:vocabulary1IncorporatesVocabulary2 vocabulary1="vocabulary" vocabulary2="Abc"/>
<sbvr:namespace1IncorporatesNamespace2 namespace1="vocabularyNamespace" namespace2="Abc-ns"/>
```

The pattern above assumes there is a vocabulary named Abc Vocabulary like this:

```
<sbvr:vocabulary xmi:id="Abc"/>
<sbvr:vocabularyNamespace xmi:id="Abc-ns"/>
```

Language: English

```
<sbvr:language xmi:id="language"/>
<sbvr:vocabularyNamespaceIsForLanguage vocabularyNamespace="vocabularyNamespace" language="language"/>
<sbvr:thingHasName thing="language" name="English"/>
<sbvr:name xmi:id="English" signifier="l-s" meaning="l-c"/>
<sbvr:individualConcept xmi:id="l-c" instance="language"/>
<sbvr:text xmi:id="l-s" value="English"/>
<sbvr:designationIsInNamespace designation="English" namespace="ISO639-2English"/>
<sbvr:vocabularyNamespace xmi:id="ISO639-2English"/>
<sbvr:namespaceHasURI namespace="ISO639-2English" URI="lm-u"/>
<sbvr:URI xmi:id="lm-u"
    value="http://www.loc.gov/standards/iso639-2/php/English_list.php"/>
```

Namespace URI: http://some.uri

```
<sbvr:namespaceHasURI namespace="vocabularyNamespace" URI="vn-uri"/>
<sbvr:URI xmi:id="vn-uri" value="http://some.uri"/>
```

Issue 9950 Revise text

Speech Community: English Mechanics

```
<sbvr:speechCommunityOwnsVocabulary speechCommunity="em" vocabulary="vocabulary"/>
<sbvr:conceptHasInstance concept="em-concept" instance="em"/>
<sbvr:speechCommunity xmi:id="em"/>
```

It is assumed for this entry that there is a name ‘English Mechanics’ for an individual concept like this:

```
<sbvr:name xmi:id="em-name" signifier="em-s" meaning="em-concept"/>
<sbvr:individualConcept xmi:id="em-concept"/>
<sbvr:text xmi:id="em-s" value="English Mechanics"/>
```

The captions “Description:”, “Note:” and “Source:” are handled for a vocabulary in the same way as for terms within a vocabulary, as shown below, except that the related meaning is given as meaning="vocabulary-concept".

Issue 9950 Revise text

13.6.2 XML Patterns for Object Types

example term

```
<sbvr:term xmi:id="exampleTerm" signifier="et-s" meaning="meaning"/>
<sbvr:objectType xmi:id="meaning"/>
<sbvr:text xmi:id="et-s" value="example term"/>
<sbvr:setIncludesThing set="vocabulary" thing="exampleTerm"/>
<sbvr:designationIsInNamespace designation="exampleTerm" namespace="vocabularyNamespace"/>
```

If there is no “See:” caption, then the following is included:

```
<sbvr:preferredDesignation xmi:id “exampleTermPreferred”/>
<sbvr:thing1IsThing2 thing1="exampleTermPreferred" thing2="exampleTerm"/>
```

Concept Type: role

```
<sbvr:role xmi:id="meaningAsRole"/>
<sbvr:thing1IsThing2 thing1="meaningAsRole" thing2="meaning"/>
```

The pattern above is used if the concept type is an SBVR concept. The pattern below is used if the concept type is not an SBVR concept.

Concept Type: example type

```
<sbvr:conceptHasInstance concept="exampleType-c" instance="meaning"/>
```

There is assumed to be a term ‘example type’ for an object type like this:

```
<sbvr:term xmi:id="exampleType" signifier="exampleType-s" meaning="exampleType-c"/>
<sbvr:objectType xmi:id="exampleType-c"/>
<sbvr:text xmi:id="exampleType-s" value="example type"/>
```

Definition: example that *is seen*

<sbvr:definition xmi:id="def-formal" expression="def-formal-e" meaning="meaning"/>
<sbvr:text xmi:id="def-formal-e" value="*example that is seen*"/>
<sbvr:concept1SpecializesConcept2 concept1="meaning" concept2="*example-concept*" />
<sbvr:closedProjectionFormalizesDefinition closedProjection="*def-formal-projection*" definition="def-formal"/>
<sbvr:closedProjectionDefinesNounConcept closedProjection="*def-formal-projection*" nounConcept="meaning"/>

The closed projection of the definition (not shown) has xmi:id="def-formal-projection". It is assumed for this entry and several others that there is a term ‘example’ for an object type like this:

<sbvr:term xmi:id="example" signifier="example-s" meaning="example-concept"/>
<sbvr:objectType xmi:id="example-concept"/>
<sbvr:text xmi:id="example-s" value="example"/>

Definition: example that shows something

<sbvr:definition xmi:id="def-semiformal" expression="def-semiformal-e" meaning="meaning"/>
<sbvr:text xmi:id="def-semiformal-e" value="*example that shows something*"/>
<sbvr:concept1SpecializesConcept2 concept1="meaning" concept2="*example-concept*" />

Definition: whatever demonstrates

<sbvr:definition xmi:id="def-informal" expression="def-informal-e" meaning="meaning"/>
<sbvr:text xmi:id="def-informal-e" value="*whatever demonstrates*"/>

Definition: A description of something

<sbvr:descriptionPortraysMeaning description="desc" meaning="meaning"/>
<sbvr:description xmi:id="desc" expression="desc-e"/>
<sbvr:text xmi:id="desc-e" value="*A description of something.*"/>

Dictionary Basis: example

None

Example: An example of an example

<sbvr:descriptiveExampleIllustratesMeaning descriptiveExample="de" meaning="meaning"/>
<sbvr:descriptiveExample xmi:id="de" expression="de-e"/>
<sbvr:text xmi:id="de-e" value="*An example of an example*"/>

General Concept: example

<sbvr:concept1SpecializesConcept2 concept1="meaning" concept2="*example-concept*" />

Necessity: Each example *is seen*.

<sbvr:statement xmi:id="nec-stmt" expression="nec-e" meaning="nec"/>
<sbvr:text xmi:id="nec-e" value="*Each example is seen.*"/>
<sbvr:proposition xmi:id="nec"/>
<sbvr:propositionIsNecessarilyTrue proposition="nec"/>
<sbvr:closedLogicalFormulationFormalizesStatement closedLogicalFormulation="*nec-formulation*" statement="nec-stmt"/>
<sbvr:closedLogicalFormulationMeansProposition closedLogicalFormulation="*nec-formulation*" proposition="nec"/>

A closed logical formulation of the statement (not shown) has xmi:id="nec-formulation".

Note: This note says little.

<sbvr:noteCommentsOnMeaning note="note" meaning="meaning"/>
<sbvr:note xmi:id="note" expression="note-e"/>
<sbvr:text xmi:id="note-e" value="*This note says little.*"/>

Possibility: Some example *is seen*.

<sbvr:statement xmi:id="pos-stmt" expression="pos-e" meaning="pos"/>
<sbvr:text xmi:id="pos-e" value="*Some example is seen.*"/>
<sbvr:proposition xmi:id="pos"/>
<sbvr:propositionIsPossiblyTrue proposition="pos"/>
<sbvr:closedLogicalFormulationFormalizesStatement closedLogicalFormulation="*pos-formulation*" statement="pos-stmt"/>
<sbvr:closedLogicalFormulationMeansProposition closedLogicalFormulation="*pos-formulation*" proposition="pos"/>

A closed logical formulation of the statement (not shown) has xmi:id="pos-formulation".

Reference Scheme: An id *of* the example term and the set of authors *of* the example term

<sbvr:referenceScheme xmi:id="refScheme" simplyUsedRole="*ethi-r2*" extensionallyUsedRole="*etha-r2*"/>

It is assumed for this entry that there is a binary fact type ‘example term *has* id’ whose ‘id’ role has xmi:id="ethi-r2".

It is assumed for this entry that there is a binary fact type ‘example term *has* author’ whose ‘author’ role has xmi:id="etha-r2".

Issue 9950 Revise text

See: example object type designation

Same as “Synonym: example object type designation”.

Source: ISO 1087-1 [‘example’]

<sbvr:referenceSupportsMeaning reference="ref" meaning="meaning"/>
<sbvr:reference xmi:id="ref" expression="*ISO 1087-1 [‘example’]*"/>

Subject Field: Philosophy

<sbvr:representationIsInSubjectField representation="exampleTerm" subjectField="philosophy"/>
<sbvr:conceptHasInstance concept="*philo-concept*" instance="philosophy"/>
<sbvr:subjectField xmi:id="philosophy"/>

It is assumed for this entry that there is a name ‘Philosophy’ for an individual concept like this:
<sbvr:name xmi:id="philo-name" signifier="philo-s" meaning="philo-concept"/>
<sbvr:individualConcept xmi:id=" philo-concept"/>
<sbvr:text xmi:id="philo-s" value="Philosophy"/>

Synonym: example object type designation

<sbvr:term xmi:id="exampleObjectTypeDesignation" signifier="eotd-s" meaning="meaning"/>
<sbvr:text xmi:id="eotd-s" value="*example object type designation*"/>
<sbvr:setIncludesThing set="vocabulary" thing="exampleObjectTypeDesignation"/>
<sbvr:designationIsInNamespace designation="exampleObjectTypeDesignation" namespace="vocabularyNamespace"/>

13.6.3 XML Patterns for Individual Concepts

Example Name

<sbvr:name xmi:id="exampleName" signifier="en-s" meaning="meaning"/>
<sbvr:individualConcept xmi:id="meaning"/>
<sbvr:text xmi:id="en-s" value="*Example Name*"/>

```
<sbvr:setIncludesThing set="vocabulary" thing="exampleName"/>
<sbvr:designationIsInNamespace designation="exampleName" namespace="vocabularyNamespace"/>
```

If there is no “See:” caption, then the following is included:

```
<sbvr:preferredDesignation xmi:id “exampleNamePreferred”/>
<sbvr:thing1IsThing2 thing1="exampleNamePreferred" thing2="exampleName"/>
```

Definition: the example that *is seen*

```
<sbvr:definiteDescription xmi:id="defDesc-formal" expression="defDesc-formal-e" meaning="meaning"/>
<sbvr:text xmi:id="defDesc-formal-e" value="the example that is seen"/>
<sbvr:concept1SpecializesConcept2 concept1="meaning" concept2="example-concept" />
<sbvr:closedProjectionFormalizesDefinition closedProjection="defDesc-formal-projection" definition="defDesc-formal"/>
<sbvr:closedProjectionDefinesNounConcept closedProjection="defDesc-formal-projection" nounConcept="meaning"/>
```

The closed projection of the definition (not shown) has xmi:id="defDesc-formal-projection". Note that informal and semiformal definitions of individual concepts follow the same pattern as shown for object types above with the exception that they are rendered as sbvr:definiteDescription.

The captions “Concept Type:”, “Description:”, “Dictionary Basis:”, “Example:”, “General Concept:”, “Necessity:”, “Note:”, “Possibility:”, “See:”, “Source:”, “Subject Field:” and “Synonym:” are handled for a name in the same way as for terms as shown above.

13.6.4 XML Patterns for Fact Types

example *is seen*

```
<sbvr:sententialForm xmi:id="exampleIsSeen" expression="eis-e" meaning="meaning" placeholder="eis-p"/>
<sbvr:factSymbol xmi:id="example.isSeen" signifier="isSeen-s" meaning="meaning"/>
<sbvr:characteristic xmi:id="meaning" role="eis-r"/>
<sbvr:factTypeFormDemonstratesDesignation factTypeForm="exampleIsSeen" designation="isSeen"/>
<sbvr:text xmi:id="eis-e" value="example is seen"/>
<sbvr:text xmi:id="isSeen-s" value="is seen"/>
<sbvr:placeholder xmi:id="eis-p" expression="example-s" startingCharacterPosition="i1" meaning="eis-r"/>
<sbvr:placeholderUsesDesignation placeholder="eis-p" designation="example"/>
<sbvr:positiveInteger xmi:id="i1" value="1"/>
<sbvr:factTypeRole xmi:id="eis-r"/>
<sbvr:roleRangesOverObjectType role="eis-r" objectType="example-concept"/>
<sbvr:setIncludesThing set="vocabulary" thing="exampleIsSeen"/>
<sbvr:setIncludesThing set="vocabulary" thing="example.isSeen"/>
<sbvr:factTypeFormIsInNamespace factTypeForm="exampleIsSeen" namespace="vocabularyNamespace"/>
<sbvr:attributiveNamespaceIsWithinVocabularyNamespace attributiveNamespace="example-ans"
    vocabularyNamespace="vocabularyNamespace"/>
<sbvr:attributiveNamespace xmi:id="example-ans" subjectConcept="example-concept"/>
<sbvr:designationIsInNamespace designation="example.isSeen" namespace="example-ans"/>
```

$example_1$ *follows* $example_2$

```
<sbvr:sententialForm xmi:id="example1FollowsExample2" expression="efe-e" meaning="meaning" placeholder="efe-p1 efe-p2"/>
<sbvr:factSymbol xmi:id="efe-follows" signifier="follows-s" meaning="meaning"/>
<sbvr:binaryFactType xmi:id="meaning" role="efe-r1 efe-r2"/>
<sbvr:factTypeFormDemonstratesDesignation factTypeForm="example1FollowsExample2" designation="efe-follows"/>
<sbvr:text xmi:id="efe-e" value="example1 follows example2"/>
```

```
<sbvr:text xmi:id="follows-s" value="follows"/>
<sbvr:text xmi:id="example1-s" value="example1"/>
<sbvr:text xmi:id="example2-s" value="example2"/>
<sbvr:placeholder xmi:id="efe-p1" expression="example1-s" startingCharacterPosition="i1" meaning="efe-r1"/>
<sbvr:placeholder xmi:id="efe-p2" expression="example2-s" startingCharacterPosition="i18" meaning="efe-r2"/>
<sbvr:placeholderUsesDesignation placeholder="efe-p1" designation="example"/>
<sbvr:placeholderUsesDesignation placeholder="efe-p2" designation="example"/>
<sbvr:positiveInteger xmi:id="i1" value="1"/>
<sbvr:positiveInteger xmi:id="i18" value="18"/>
<sbvr:factTypeRole xmi:id="efe-r1"/>
<sbvr:factTypeRole xmi:id="efe-r2"/>
<sbvr:roleRangesOverObjectType role="efe-r1" objectType="example-concept"/>
<sbvr:roleRangesOverObjectType role="efe-r2" objectType="example-concept"/>
<sbvr:setIncludesThing set="vocabulary" thing=" example1FollowsExample2"/>
<sbvr:setIncludesThing set="vocabulary" thing=" efe-follows"/>
<sbvr:factTypeFormIsInNamespace factTypeForm="example1FollowsExample2" namespace="vocabularyNamespace"/>
```

Definition: the example$_1$ *comes after* the example$_2$ *in* a sequence

```
<sbvr:definition xmi:id="efe-def-formal" signifier="efe-def-formal-e" meaning="meaning"/>
<sbvr:text xmi:id="efe-def-formal-e" value="the example1 comes after the example2 in a sequence"/>
<sbvr:closedProjectionFormalizesDefinition closedProjection="efe-projection" definition="efe-def-formal"/>
<sbvr:closedProjectionDefinesFactType closedProjection="efe-projection" factType="meaning"/>
<sbvr:variableMapsToFactTypeRole variable="efe-var1" factTypeRole="efe-r1"/>
<sbvr:variableMapsToFactTypeRole variable="efe-var2" factTypeRole="efe-r2"/>
```

The definition formally defines ‘example$_1$ *follows* example$_2$’ and has a closed projection (not shown) with xmi:id="efe-projection" projectionVariable="efe-var1 efe-var2".

Definition: the first example is after the second

```
<sbvr:definition xmi:id="efe-def-informal" signifier="efe-def-informal-e" meaning="meaning"/>
<sbvr:text xmi:id="efe-def-informal-e" value="the first example is after the second"/>
```

Issue 9950 Revise text

See: example$_1$ *has* prior example

Same as “Synonymous Form: example$_1$ *has* prior example”.

Synonymous Form: example$_1$ *has* prior example

```
<sbvr:sententialForm xmi:id="example1HasPriorExample" expression="ehpe-e" meaning="meaning" placeholder="ehpe-p1
    ehpe-p2"/>
<sbvr:factSymbol xmi:id="ehpe-has" signifier="has-s" meaning="meaning"/>
<sbvr:factTypeFormDemonstratesDesignation factTypeForm="example1HasPriorExample" designation="ehpe-has"/>
<sbvr:term xmi:id="example.priorExample" signifier="priorExample-s" meaning="efe-r2"/>
<sbvr:factTypeRoldDesignation xmi:id="example.priorExample-ftr"/>
<sbvr:thing1IsThing2 thing1="example.priorExample" thing2="example.priorExample-ftr"/>
<sbvr:text xmi:id="ehpe-e" value="example1 has prior example"/>
<sbvr:text xmi:id="has-s" value="has"/>
<sbvr:text xmi:id="priorExample-s" value="prior example"/>
<sbvr:placeholder xmi:id="ephe-p1" expression="example1-s" startingCharacterPosition="i1" meaning="efe-r1"/>
```

```
<sbvr:placeholder xmi:id="ephe-p2" expression="priorExample-s" startingCharacterPosition="i14" meaning="efe-r2"/>
<sbvr:placeholderUsesDesignation placeholder="ephe-p1" designation="example"/>
<sbvr:positiveInteger xmi:id="i1" value="1"/>
<sbvr:positiveInteger xmi:id="i14" value="14"/>
<sbvr:setIncludesThing set="vocabulary" thing="example1HasPriorExample"/>
<sbvr:factTypeFormIsInNamespace factTypeForm="example1HasPriorExample" namespace="vocabularyNamespace"/>
<sbvr:attributiveNamespaceIsWithinVocabularyNamespace attributiveNamespace="example-ans"
    vocabularyNamespace="vocabularyNamespace"/>
<sbvr:attributiveNamespace xmi:id="example-ans" subjectConcept="example-concept"/>
<sbvr:designationIsInNamespace designation="example.priorExample" namespace="example-ans"/>
```

If there is a term ‘prior example’ for an object type like this:

```
<sbvr:term xmi:id="priorExample" signifier="priorExample-s" meaning="priorExample-c"/>
```

then the following is included:

```
<sbvr:placeholderUsesDesignation placeholder="ephe-p2" designation="priorExample"/>
<sbvr:roleRangesOverObjectType role="efe-r2" objectType="priorExample-c"/>
```

The captions “Concept Type:”, “Description:”, “Dictionary Basis:”, “Example:”, “General Concept:”, “Necessity:”, “Note:”, “Possibility:” and “Source:” are handled for a fact type form in the same way as for terms as shown above.

13.6.5 XML Patterns for Rule Sets

Xyz Rules

```
<sbvr:set xmi:id="ruleSet"/>
<sbvr:thingHasName thing="ruleSet" name="XyzRules"/>
<sbvr:name xmi:id="XyzRules" signifier="XyzRules-s" meaning="ruleSet-concept"/>
<sbvr:individualConcept xmi:id="ruleSet-concept" instance="ruleSet"/>
<sbvr:text xmi:id="XyzRules-s" value="Xyz Rules"/>
<sbvr:setIncludesThing set="vocabulary" thing="XyzRules"/>
<sbvr:designationIsInNamespace designation=" XyzRules " namespace="vocabularyNamespace"/>
```

Vocabulary: Abc Vocabulary

None.

The captions “Description:”, “Note:” and “Source:” are handled for a rule set in the same way as for terms within a vocabulary, as shown above, except that the related meaning is given as meaning="ruleSet-concept".

13.6.6 XML Patterns for Guidance Statements

Each example must *be seen*.

```
<sbvr:guidanceStatement xmi:id="stmt-formal" expression="stmt-formal-e" meaning="meaning"/>
<sbvr:elementOfGuidance xmi:id="meaning"/>
<sbvr:text xmi:id="stmt-formal-e" value="Each example must be seen."/>
<sbvr:closedLogicalFormulationFormalizesStatement closedLogicalFormulation="stmt-formal-formulation"
    statement="stmt-formal"/>
<sbvr:closedLogicalFormulationMeansProposition closedLogicalFormulation="stmt-formal-formulation" proposition="meaning"/>
<sbvr:setIncludesThing set="ruleSet" meaning="meaning"/>
```

The closed logical formulation of the statement (not shown) has xmi:id="stmt-formal-formulation".

Guidance Type: operative business rule

In this case where the guidance type is an SBVR concept, the line above that says,
"<sbvr:elementOfGuidance xmi:id="meaning"/>", is replaced with this:
<sbvr:*operativeBusinessRule* xmi:id="meaning"/>

Guidance Type: exemplary rule

<sbvr:conceptHasInstance concept="*exemplaryRule-c*" instance="meaning"/>

This pattern is used if the concept type is not an SBVR concept. There is assumed to be a term 'exemplary rule' for an object type like this:
<sbvr:term xmi:id="exemplaryRule" signifier="exemplaryRule-s" meaning="exemplaryRule-c"/>
<sbvr:objectType xmi:id="exemplaryRule-c"/>
<sbvr:text xmi:id="exemplaryRule-s" value="exemplary rule"/>

Issue 9950 Revise text

Enforcement Level: strict

<sbvr:operativeBusinessRuleHasLevelOfEnforcement
operativeBusinessRule="meaning~~AsOperativeBusinessRule~~"
levelOfEnforcement="strict-instance"/>
<sbvr:conceptHasInstance concept="*strict-concept*" instance="strict-instance"/>
<sbvr:levelOfEnforcement xmi:id="strict-instance"/>

It is assumed that the name 'strict' represents an individual concept like this:
<sbvr:name xmi:id="strict" signifier="strict-s" meaning="strict-concept"/>
<sbvr:individualConcept xmi:id="strict-concept"/>
<sbvr:text xmi:id="strict-s" value="strict"/>

Name: Rule 25

<sbvr:thingHasName thing="meaning" name="Rule25"/>
<sbvr:name xmi:id="Rule25" signifier="Rule25-s" meaning="rule25Meaning"/>
<sbvr:individualConcept xmi:id="rule25Meaning" instance="meaning"/>
<sbvr:text xmi:id="Rule25-s" value="*Rule 25*"/>
<sbvr:setIncludesThing set="vocabulary" thing="Rule25"/>
<sbvr:designationIsInNamespace designation="Rule25" namespace="vocabularyNamespace"/>

Synonymous Statement: It is obligatory that each rule *be seen*.

<sbvr:guidanceStatement xmi:id="synstmt-formal" expression="synstmt-formal-e" meaning="meaning"/>
<sbvr:text xmi:id="synstmt-formal-e" value="*It is obligatory that each rule be seen.*"/>
<sbvr:closedLogicalFormulationFormalizesStatement closedLogicalFormulation="*synstmt-formal-formulation*"
statement="synstmt-formal"/>
<sbvr:closedLogicalFormulationMeansProposition closedLogicalFormulation="*synstmt-formal-formulation*" proposition="meaning"/>

The captions "Description:", "Example:", "Note:" and "Source:" are handled for a guidance statement in the same way as for terms as shown above.

~~13 Vocabulary-driven Interchange using MOF and XMI~~

This section presents the Vocabulary-to-MOF/XMI Vocabulary and the Vocabulary-to-MOF/XMI Mapping Rule Set. The vocabulary is used by the rule set, which contains rules for transforming any vocabulary defined in terms of the Logical Formulation of Semantics Vocabulary into a MOF model and XMI-based XML schema that supports repository services and data interchange of facts that can be formulated as atomic or instantiation formulations.

The UML diagrams shown in the vocabulary sections of this document might at first appear to make up a MOF model of SBVR. However, these diagrams represent vocabulary, not MOF classes. The diagrams show orthogonal dimensions of specialization in which a single thing can be an instance of multiple concepts at the same time without there being a single most specific concept. This sort of specialization is normal in language but is not supported by MOF which requires an object to be an instance of exactly one most-specific class. The diagrams show characteristics as if they were unary associations, but unary associations are not supported by MOF, nor are ternary associations which also occur in the diagrams. The MOF model of SBVR is created by a transformation from the SBVR vocabulary following the rules in this section.

~~13.1 Vocabulary-to-MOF/XMI Vocabulary~~

~~Issue 10528 Revise text~~
~~Issue 10569 Remove footnotes and revise MOF/XMI text~~

This vocabulary includes the Logical Formulation of Semantics Vocabulary Meaning and Representation Vocabulary as its starting point. It further incorporates many concepts from UML 2, MOF 2, and XMI 2.1 specifications.

UML 2 concepts are limited to the subset of UML 2 included in EMOF as described by the final adopted specification of the *Meta Object Facility (MOF) 2.0 Core*.

XMI concepts are used to tailor XML schema production. See "Tailoring Schema Production" in OMG's *MOF 2.0/XMI Mapping Specification, v2.1* for further descriptions of how MOF tags are used to represent XMI directives.

Reference schemes are given for several of the types. These reference schemes satisfy the needs of the SBVR-to-MOF/XMI mapping and are consistent with constraints in the UML 2 specification.

~~Vocabulary-to-MOF/XMI Vocabulary~~

~~Language:~~	~~English~~
~~Included Vocabulary:~~	~~Meaning and Representation Vocabulary~~
~~Included Vocabulary:~~	~~Essential SBVR Vocabulary~~

~~**expression$_1$ *combines* expression$_2$ *with* expression$_3$**~~

~~Definition:~~	~~the expression$_1$ is the result of concatenating the expression$_2$ to the front of the expression$_3$~~

~~13.1.1 Mapping~~

~~representation *maps to* UML element~~

~~Definition: the representation is mapped based on the Vocabulary to MOF/XMI Mapping Rule Set to the UML element~~

~~Synonymous Form: UML element *comes from* representation~~

~~vocabulary namespace *maps to* package~~

~~Definition: the vocabulary namespace is mapped based on the Vocabulary to MOF/XMI Mapping Rule Set to the UML package that contains UML classes providing a MOF/XMI implementation of the vocabulary namespace~~

~~Synonymous Form: package *comes from* vocabulary namespace~~

~~Issue 9929 Replace text~~
~~Issue 9958 Replace form of expression with fact type form~~

~~text *is for* placeholder~~

~~Definition: the text represents the placeholder of a fact type form form of expression based on the designation corresponding to the placeholder with the conditional addition of a numeral in the case where the same designation also occurs for another placeholder within the fact type form form of expression~~

~~Note: A text for a placeholder is normalized to have no leading or trailing spaces and no other white space than single spaces.~~

~~Example: The text "concept" is for the first placeholder in the fact type form form of expression 'concept has definition' and the text "definition" is for the second placeholder.~~

~~Example: The text "thing1" is for the first placeholder in the fact type form form of expression 'thing is thing' and the text "thing2" is for the second placeholder.~~

~~Note: Note that subscripts appearing in a fact type form form of expression (e.g., 'thing$_1$ is thing$_2$') are not actually part of the fact type form form of expression or its placeholders, and it is only by coincidence that they might match numerals added to the text for placeholders.~~

~~XMI name *is derived from* text~~

~~Definition: the XMI name is derived from the text as described below:~~

~~An XMI name derived from a text takes each character directly from the sequence except for the following:~~

~~1. A space character is replaced by a hyphen ('-').~~

~~2. A hyphen is replaced by two consecutive hyphens ('--').~~

~~3. An underscore character is replaced by two consecutive underscores ('__').~~

~~4. A character specified to be invalid by the *Extensible Markup Language (XML) 1.0* specification (Second Edition, see www.w3.org/TR/REC-xml), either in general or as a first character if in the first character position of the text, is replaced with an underscore followed by the hexadecimal encoding (using lower case) of the character with leading zeros suppressed followed by another underscore (e.g., '_d7ff_').~~

~~Note: Note that a text is normalized to have no leading or trailing spaces and no other white space than single spaces.~~

~~instantiation prefix~~

~~Definition:~~	~~text that is used with a designation to form a class name for objects representing that a thing is an instance of the designated concept~~
~~Concept Type:~~	~~role~~

~~instantiation prefix *is used for* language~~

~~Definition:~~	~~the instantiation prefix is used when generating class names for designations of the language~~

~~Issue 10528 Add text~~

~~expression$_1$ *combines* expression$_2$ *with* expression$_3$~~

~~Definition:~~	~~the expression$_1$ is the result of concatenating the expression$_3$ to the end of the expression$_2$~~

~~13.1.2 Relevant Concepts from UML 2~~

~~UML element~~

~~Source:~~	~~UML 2 Infrastructure [Core::Abstractions::Elements::Element]~~

~~name~~

~~Source:~~	~~UML 2 Infrastructure [Core::Basic::NamedElement::name]~~
~~General Concept:~~	~~string~~
~~Concept Type:~~	~~role~~

~~named element~~

~~Source:~~	~~UML 2 Infrastructure [Core::Basic::NamedElement]~~
~~General Concept:~~	~~UML element~~

~~named element *has* name~~

~~Source:~~	~~based on UML 2 Infrastructure [Core::Basic::NamedElement::name]~~

~~type~~

~~Source:~~	~~UML 2 Infrastructure [Core::Basic::Type]~~
~~General Concept:~~	~~named element~~
~~Reference Scheme:~~	~~the name *of* the type and a package that *owns* the type~~

~~typed element~~

~~Source:~~	~~UML 2 Infrastructure [Core::Basic::TypedElement]~~
~~General Concept:~~	~~named element~~

~~***typed element*** ***has*** ***type***~~

~~Source: based on UML 2 Infrastructure [Core::Basic::TypedElement::type]~~

~~***class***~~

~~Source: UML 2 Infrastructure [Core::Basic::Class]~~

~~General Concept: type~~

~~***property***~~

~~Source: UML 2 Infrastructure [Core::Basic::Property]~~

~~General Concept: typed element~~

~~Issue 10528 Add text~~

~~***property has lower bound***~~

~~***property has upper bound***~~

~~***lower bound***~~

~~Source: UML 2 Infrastructure [Core::Abstractions::Multiplicities:: MultiplicityElement::lower]~~

~~General Concept: integer~~

~~Concept Type: role~~

~~***upper bound***~~

~~Source: UML 2 Infrastructure [Core::Abstractions::Multiplicities:: MultiplicityElement::upper]~~

~~General Concept: unlimited natural number~~

~~Concept Type: role~~

~~***unlimited natural number***~~

~~Source: UML 2 Infrastructure [Core::PrimitiveTypes::UnlimitedNaturalNumber]~~

~~Infinity~~

~~Source: based on UML 2 Infrastructure ['*' designating an instance of Core::PrimitiveTypes::UnlimitedNaturalNumber]~~

~~General Concept: unlimited natural number~~

~~***owned attribute***~~

~~Source: UML 2 Infrastructure [Core::Basic::Class::ownedAttribute]~~

~~General Concept: property~~

~~Concept Type: role~~

~~Reference Scheme: the name *of* the owned attribute and a class that *has* the owned attribute~~

~~class *has* owned attribute~~

~~Source:~~ ~~based on UML 2 Infrastructure [Core::Basic::Class::ownedAttribute]~~

~~superclass~~

~~Source:~~ ~~UML 2 Infrastructure [Core::Basic::Class::superClass]~~

~~General Concept:~~ ~~class~~

~~Concept Type:~~ ~~role~~

~~class *has* superclass~~

~~Source:~~ ~~based on UML 2 Infrastructure [Core::Basic::Class::superClass]~~

~~class *is abstract*~~

~~Source:~~ ~~based on UML 2 Infrastructure [Core::Basic::isAbstract]~~

~~data type~~

~~Source:~~ ~~UML 2 Infrastructure [Core::Basic::DataType]~~

~~General Concept:~~ ~~type~~

~~primitive type~~

~~Source:~~ ~~UML 2 Infrastructure [Core::DataTypes::PrimitiveType]~~

~~General Concept:~~ ~~data type~~

~~Integer Type~~

~~Source:~~ ~~UML 2 Infrastructure ['Core::PrimitiveTypes::Integer']~~

~~General Concept:~~ ~~primitive type~~

~~string~~

~~Source:~~ ~~UML 2 Infrastructure [Core::PrimitiveTypes::String]~~

~~General Concept:~~ ~~text~~

~~String Type~~

~~Source:~~ ~~UML 2 Infrastructure ['Core::PrimitiveTypes::String']~~

~~General Concept:~~ ~~data type~~

~~package~~

~~Source:~~ ~~UML 2 Infrastructure [Core::Basic::Package]~~

~~General Concept:~~ ~~named element, namespace~~

~~owned type~~

~~Source:~~ ~~UML 2 Infrastructure [Core::Basic::Package::ownedType]~~

~~General Concept:~~ ~~type~~

~~Concept Type:~~ ~~role~~

~~**package *has* owned type**~~

~~Source: based on UML 2 Infrastructure [Core::Basic::Package::ownedType]~~

~~13.1.3 XMI for MOF 2~~

~~Issue 10528 Revise text~~

~~**named element *has* XMI name**~~

~~Definition: the named element is tagged with the XMI name~~

~~**package *has* XMI namespace prefix**~~

~~Definition: the package is tagged with the XMI namespace prefix~~

~~**package *has* XMI namespace URI**~~

~~Definition: the package is tagged with the XMI namespace URI~~

~~**XMI name**~~

~~Source: XMI 2.1 Tags ['xmiName']~~

~~Definition: string that is the value of an 'org.omg.xmi.xmiName' tag~~

~~Concept Type: role~~

~~**XMI namespace prefix**~~

~~Source: XMI 2.1 Tags ['nsPrefix']~~

~~Definition: string that is the value of an 'org.omg.xmi.nsPrefix' tag~~

~~Concept Type: role~~

~~**XMI namespace URI**~~

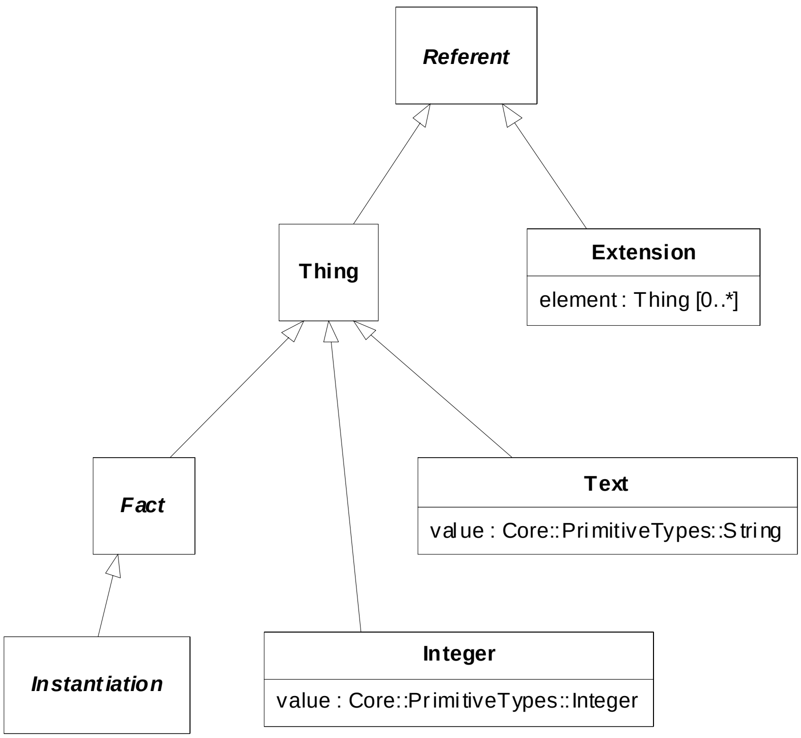
~~Source: XMI 2.1 Tags ['nsURI']~~

~~Definition: string that is the value of an 'org.omg.xmi.nsURI' tag~~

~~Concept Type: role~~

~~13.2 Essential SBVR~~

The Essential SBVR Package is a UML/MOF package containing classes used by target packages of the Vocabulary-to-MOF/XMI Mapping Rule Set. Its contents are all defined in the Essential SBVR Vocabulary below.



~~**Figure 13.1 The Essential SBVR Package**~~

The Referent Class is used to represent what a fact can refer to. It has two subclasses. One, the Thing Class, is used to represent individual things. The other, the Extension Class, is used to represent a set of things, the entire set for which a fact is true. Each instance of the Extension Class has zero or more members which are instances of the Thing Class.

The Thing Class is used to represent things in general that are the subjects or objects of facts. The expressed type or types of a thing are known from facts about it. The Thing Class has subclasses for different kinds of representations of things: the Fact Class, the Text Class and the Integer Class.

A subclass of the Fact Class is created according to the Vocabulary-to-MOF/XMI Mapping Rule Set for each sentential form in a vocabulary. Each instance of one of these subclasses represents a fact of the fact type that has the sentential form.

A subclass of the Instantiation Class is created according to the Vocabulary-to-MOF/XMI Mapping Rule Set for each designation in a vocabulary. Each instance of one of these subclasses represents a fact that a thing is an instance of the concept denoted by the designation.

The Text Class and the Integer Class provide convenient ways to use text and integers in representing facts.

The Essential SBVR Vocabulary is used to refer to model elements of the Essential SBVR Package.

~~Essential SBVR Vocabulary~~

~~Language:~~ ~~English~~

~~Issue 9457 Remove some "Necessity" statements~~

~~Essential SBVR Package~~

~~Definition:~~ ~~the package that contains classes used in general by MOF/XMI implementations of vocabularies generated according to the Vocabulary-to-MOF/XMI Mapping Rule Set~~

~~Necessity:~~ ~~'Essential SBVR' *is* the name *of* the Essential SBVR Package.~~
~~'ESBVR' *is* the XMI namespace prefix *of* the Essential SBVR Package.~~

~~Referent Class~~

~~Definition:~~ ~~the class that *is owned by* the Essential SBVR Package and that *has* the name 'Referent'~~

~~Necessity:~~ ~~The Referent Class *is abstract*.~~

~~Thing Class~~

~~Definition:~~ ~~the class that *is owned by* the Essential SBVR Package and that *has* the name 'Thing'~~

~~Necessity:~~ ~~The Thing Class *is* not *abstract*.~~
~~The Referent Class *is* a superclass *of* the Thing Class.~~

~~Fact Class~~

~~Definition:~~ ~~the class that *is owned by* the Essential SBVR Package and that *has* the name 'Fact'—~~

~~Necessity:~~ ~~The Fact Class *is abstract*.~~
~~The Thing Class *is* a superclass *of* the Fact Class.~~
~~'fact' *is* the XMI name *of* the Fact Class.~~

~~Instantiation Class~~

~~Definition:~~ ~~the class that *is owned by* the Essential SBVR Package and that *has* the name 'Instantiation'~~

~~Necessity:~~ ~~The Instantiation Class *is abstract*.~~
~~The Fact Class *is* a superclass *of* the Instantiation Class.~~
~~'instantiation' *is* the XMI name *of* the Instantiation Class.~~

~~Integer Class~~

~~Definition:~~ ~~the class that *is owned by* the Essential SBVR Package and that *has* the name 'Integer'~~

~~Necessity:~~ ~~The Thing Class *is* a superclass *of* the Integer Class.~~
~~The Integer Class *is* not *abstract*.~~
~~'value' *is* the name *of* an owned attribute *of* the Integer Class and the Integer Type *is* the type *of* that owned attribute.~~
~~'integer' *is* the XMI name *of* the Integer Class.~~

~~Text Class~~

~~Definition:~~ ~~the class that is owned by the Essential SBVR Package and that has the name 'Text'~~

~~Necessity:~~ ~~The Thing Class is a superclass of the Text Class.~~

~~The Text Class is not abstract.~~

~~'value' is the name of an owned attribute of the Text Class and the String Type is the type of that owned attribute.~~

~~'text' is the XMI name of the Text Class.~~

~~Extension Class~~

~~Definition:~~ ~~the class that is owned by the Essential SBVR Package and that has the name 'Extension'~~

~~Necessity:~~ ~~The Referent Class is a superclass of the Extension Class.~~

~~The Extension Class is not abstract.~~

~~'element' is the name of an owned attribute of the Extension Class, the Thing Class is the type of the owned attribute, 0 is the lower bound of the owned attribute and Infinity is the upper bound of the owned attribute.~~

~~'extension' is the XMI name of the Extension Class.~~

~~13.3 Vocabulary-to-MOF/XMI Mapping Rule Set~~

The Vocabulary-to-MOF/XMI Mapping Rule Set guides the transformation of a vocabulary namespace defined in terms of the Logical Formulation of Semantics Vocabulary into a MOF-compliant model with tags for XML schema production based on the XMI 2.1 specification.

In general, the rules should be strictly enforced, but deviation is permissible for rules that map symbols to UML and XMI names as long as all parties using the resulting MOF model or XML schema are aware of the deviations.

The rules aim at producing a UML package. The source is a vocabulary namespace. The rules are expressed using the Vocabulary-to-MOF/XMI Vocabulary and the Essential SBVR Vocabulary.

~~Vocabulary-to-MOF/XMI Mapping Rule Set~~

~~Language:~~ ~~English~~

~~Vocabulary:~~ ~~Vocabulary-to-MOF/XMI Vocabulary~~

~~13.3.1 Namespace Mapping Rules~~

~~Issue 9474 Add note~~

~~1. A vocabulary namespace must map to a package.~~

~~Possibility:~~ ~~It is possible that a vocabulary namespace maps to more than one package.~~

Note: ~~While a vocabulary namespace maps to a package, its name maps to a class, just like any name, following the designation mapping rules below.~~

~~13.3.2 Designation Mapping Rules~~

~~Issue 10528 Revise text~~

1. ~~Each designation that *is in* a vocabulary namespace that *maps to* a package must *map to* exactly one owned type *of* the package if the signifier *of* the designation *is* a text.~~
2. ~~Each owned type that *comes from* a designation must *be* a class.~~
3. ~~The Instantiation Class must *be* a superclass *of* each class that *comes from* a designation.~~
4. ~~Each class that *comes from* a designation must not *be abstract*.~~
5. ~~The name *of* each class that *comes from* a designation that *is in* a vocabulary namespace that *is for* a language must *be* the text that *combines* the instantiation prefix that *is used for* the language *with* the signifier *of* the designation.~~
6. ~~The instantiation prefix that *is used for* English must *be* "is ."~~
7. ~~The XMI name *of* each class that *comes from* a designation must *be derived from* the name *of* the class.~~
8. ~~Each class that *comes from* a designation *has* exactly one owned attribute.~~
9. ~~The name *of* the owned attribute *of* each class that *comes from* a designation must *be* the signifier *of* the designation.~~
10. ~~The type *of* the owned attribute *of* each class that *comes from* a designation must *be* the Referent Class.~~
11. ~~The lower bound and the upper bound *of* the owned attribute *of* each class that *comes from* a designation must *be* 1.~~
12. ~~The XMI name *of* the owned attribute *of* each class that *comes from* a designation must *be derived from* the name *of* the owned attribute.~~

~~13.3.3 Sentential Form Mapping Rules~~

~~Issue 10528 Revise text~~

1. ~~Each sentential form that *is in* a vocabulary namespace that *maps to* a package must *map to* exactly one owned type *of* the package if the expression *of* the sentential form *is* a text.~~
2. ~~Each owned type that *comes from* a sentential form must *be* a class.~~
3. ~~The Fact Class must *be* a superclass *of* each class that *comes from* a sentential form.~~
4. ~~Each class that *comes from* a sentential form must not *be abstract*.~~
5. ~~The name *of* each class that *comes from* a sentential form must *be* the expression *of* sentential form.~~

6. ~~The XMI name of each class that comes from a sentential form must be derived from the text that is the expression of the sentential form.~~

~~13.3.4 Placeholder Mapping Rules~~

~~Issue 10528 Revise text~~

1. ~~Each placeholder of a sentential form that maps to a class must map to exactly one owned attribute of the class.~~
2. ~~The Referent Class must be the type of each property that comes from a placeholder.~~
3. ~~The name of each property that comes from a placeholder must be the text that is for the placeholder.~~
4. ~~The XMI name of a property that comes from a placeholder must be derived from the text that is for the placeholder.~~
5. ~~The lower bound and the upper bound of each property that comes from a placeholder must be~~ 1.

~~13.3.5 Notes and Limitations~~

~~Issue 9958 Replace form of expression with fact type form~~

1. ~~If there are specifications of XMI names in addition to or overriding what is provided by the mapping rules, then a writer and reader of XML documents must both share those specifications.~~
2. ~~If different reference schemes are used, then the XML document contents should satisfy the reference schemes of both (or at least that of the reader). But this is a matter of selecting content and is independent of the XML format of the content.~~
3. ~~Only designations and fact type forms forms of expression expressed in Unicode are used to derive MOF and XMI names according to mapping rules. Other kinds of expression require additional specification beyond what is covered by the rules.~~

14 Index of Vocabulary Entries (Informative)

A

B

C

D

E

F

G

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

15 Supporting Documents

Issue 9468 Replace text
Issue 9930 Add text, remove text

Several XML documents are derived from this document, particularly for the following vocabularies specified in clauses 7 through 13. Each of these has a namespace URI specified in clause 7.

- Vocabulary Registration Vocabulary
- Meaning And Representation Vocabulary
- Logical Formulation of Semantics Vocabulary
- Formal Logic and Mathematics Vocabulary
- Vocabulary for Describing Business Vocabularies
- Vocabulary for Describing Business Rules
- SBVR Vocabulary
- ~~Vocabulary to MOF/XMI Vocabulary~~
- ~~Essential SBVR Vocabulary~~

The content of each of the documents listed in this clause is normative.

Issue 9930 Replace text and indent text
Issue 11283 Replace text

~~15.1 MOF 2.0~~

~~A MOF-based metamodel is created from each of the vocabularies listed above following the rules of the Vocabulary to MOF/XMI Mapping Rule Set and is available as an XML document based on CMOF 2.0 XMI. The URL of each document is constructed by adding "-cmof.xml" to the corresponding namespace URI. For example, for the SBVR metamodel mapped from the SBVR Vocabulary the URL is http://schema.omg.org/specs/SBVR/1.0/SBVR-cmof.xml.~~

~~Each of the metamodels builds on a MOF package named Essential SBVR which is available as an XML document based on CMOF 2.0 XMI. The document's URL is http://schema.omg.org/specs/SBVR/1.0/EssentialSBVR-cmof.xml.~~

15.1 SBVR Metamodel

Each MOF-based metamodel package shown in 13.2.1 is serialized, with all merging of packages performed, as an XML document. The URL of each document is constructed by adding ~~"-mof"~~ "-model" in front of the ".xml" in the corresponding namespace URI. Each document's URL is listed here:

http://www.omg.org/spec/SBVR/1.0/MeaningAndRepresentation-model.xml
http://www.omg.org/spec/SBVR/1.0/LogicalFormulationOfSemantics-model.xml
http://www.omg.org/spec/SBVR/1.0/DescribingBusinessVocabularies-model.xml

http://www.omg.org/spec/SBVR/1.0/DescribingBusinessRules-model.xml
http://www.omg.org/spec/SBVR/1.0/SBVR-model.xml

~~15.2 XML Schema~~

~~An XML Schema is created based on the XMI 2.1 specification from each of the MOF-based metamodels. The URL of each document is constructed by adding ".xsd" to the corresponding namespace URI. For example, for the SBVR metamodel mapped from the SBVR Vocabulary the URL is http://schema.omg.org/specs/SBVR/1.0/SBVR.xsd.~~

~~The URL of the Essential SBVR XML Schema is http://schema.omg.org/specs/SBVR/1.0/EssentialSBVR.xsd.~~

15.2 SBVR Metamodel XML Schema

An XML Schema is created based on the XMI 2.1 specification from each of the MOF-based metamodel packages listed in 15.1. The URL of each document is constructed by putting ".xsd" in place of ".xml" in the corresponding namespace URI. Each schema's URL is listed here:

http://www.omg.org/spec/SBVR/1.0/MeaningAndRepresentation.xsd
http://www.omg.org/spec/SBVR/1.0/LogicalFormulationOfSemantics.xsd
http://www.omg.org/spec/SBVR/1.0/DescribingBusinessVocabularies.xsd
http://www.omg.org/spec/SBVR/1.0/DescribingBusinessRules.xsd
http://www.omg.org/spec/SBVR/1.0/SBVR.xsd

~~15.3 SBVR Content~~

~~For each of Clauses 7 through 13, all vocabulary entries and rules are described in terms of the SBVR metamodel. An XMI document for each chapter or section that specifies a vocabulary has a URL constructed by adding "-sbvr.xml" to the vocabulary's namespace URI. For example, for Clause 12 the URL is http://schema.omg.org/specs/SBVR/1.0/DescribingBusinessRules-sbvr.xml.~~

15.3 MOF-based SBVR Model of SBVR

Issue 11283 Replace text

For each of clauses 7 through 12, all vocabulary entries and rules are described in terms of the SBVR Metamodel (http://www.omg.org/spec/SBVR/1.0/SBVR-model.xml) and are serialized as XML documents based on http://www.omg.org/spec/SBVR/1.0/SBVR.xsd. Each clause that specifies a vocabulary is serialized in a separate document whose URL is the same as the vocabulary's namespace URI. These documents are an XML serialization of SBVR in terms of itself. Each document's URL is listed here:

http://www.omg.org/spec/SBVR/1.0/VocabularyRegistration.xml
http://www.omg.org/spec/SBVR/1.0/MeaningAndRepresentation.xml
http://www.omg.org/spec/SBVR/1.0/LogicalFormulationOfSemantics.xml
http://www.omg.org/spec/SBVR/1.0/FormalLogicAndMathematics.xml
http://www.omg.org/spec/SBVR/1.0/DescribingBusinessVocabularies.xml
http://www.omg.org/spec/SBVR/1.0/DescribingBusinessRules.xml
http://www.omg.org/spec/SBVR/1.0/SBVR.xml

In each of the XML documents, an xmi:id used for a designation in a vocabulary namespace is constructed from the signifier of the designation by upcasing each character that follows a blank and then removing the blanks. Similarly, an xmi:id for a fact type form is constructed from the expression of the fact type form by removing subscripts, upcasing each character that follows a blank and then removing the blanks. This allows any of these designations and fact type forms described by one of the documents to be referenced using a URI which appends a "#" and an xmi:id to the document's URL. For example, a URI for 'noun concept' is

http://www.omg.org/spec/SBVR/1.0/MeaningAndRepresentation.xml#nounConcept.

Part III - Annexes

This part contains the annexes, including:

A - Overview of the Approach

B - The Business Rules Approach

C - SBVR Structured English

D - SBVR Structured English Patterns

E - EU-Rent Example

F - The RuleSpeak® Business Rule Notation

G - Concept Diagram Graphic Notation

H - Use of UML Notation in a Business Context to Represent SBVR-style Vocabularies

I - The ORM Notation for Verbalizing Facts and Business Rules

J - ORM Examples Related to the Logical Foundations for SBVR

K- Mappings and Relationships to Other Initiatives

L - A Conceptual Overview of SBVR and the NIAM2007 Procedure to Specify a Conceptual Schema

M- Additional References

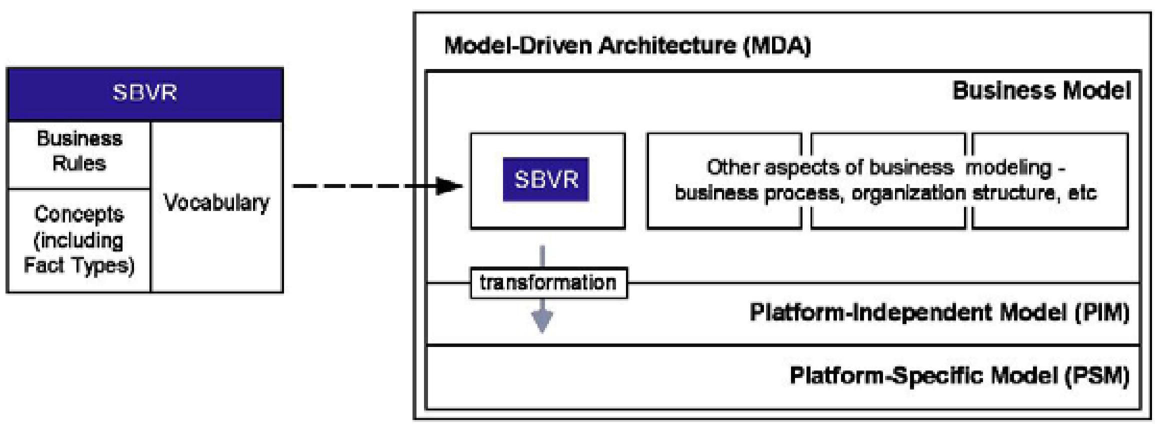
Annex A
(informative)

Overview of the Approach

Issue 9941 Replace symbol with designation
Issue 9955 Replace text in footnote

A.1 Positioning of SBVR in Model-Driven Architecture

SBVR is positioned to be entirely within the business model layer of the OMG's Model Driven Architecture (MDA)[1].



This positioning has two implications.

- SBVR is targeted at business rules and business vocabularies, including those relevant for usage in conjunction with those rules. Other aspects of business models also have to be developed, including business process and organization structure, but these are to be addressed by the OMG in other initiatives.
- Business models, including the models that SBVR supports, describe businesses and not the IT systems that support them.

In MDA, IT systems are specified using Platform Independent Models (PIMs) and Platform-Specific Models (PSMs). Guidance will be needed for transformation of business models to PIMs. Such guidance is outside the scope of SBVR. It is

1. SBVR enables the specific capture of terminology and meaning for any level of the MDA, so SBVR could be used for PIM and PSM vocabularies and rules. However, this specification is focused on SBVR as a vehicle for describing businesses rather than their information systems. In the kinds of SBVR model assumed here, the concept called "customer" would be a role of a real-world person or organization. In a PIM, it would be a UML class whose objects represent real-world customers; the business rule "a rental car must not be handed over to a customer who appears to be intoxicated" would probably not appear in a PIM.

anticipated that the OMG will ensure that the metamodels for different aspects of business modeling form a coherent whole, and will call for development of guidance on the transformation from business model to PIM as appropriate.

A.2 The Key Notions of the SBVR Approach

A.2.1 What is Semantics?

'Semantics' is "the meaning or relationship of meanings of a sign or set of signs" [MWCD]. In SBVR the signs can be of any form: words, phrases, codes, numbers, icons, sounds, etc. SBVR includes two specialized vocabularies:

- the SBVR "Vocabulary for Describing Business Vocabularies," which deals with all kinds of terms and meanings (other than meanings of Business Rules);
- the SBVR "Vocabulary for Describing Business Rules," which deals with the specification of the meaning of business rules, and builds on the "Vocabulary for Describing Business Vocabularies."

The two have been separated so that the "Vocabulary for Describing Business Vocabularies" could be used independently - for example, as a basis for vocabularies for business processes or organizational roles.

The next two subclauses deal with the semantics of business vocabularies and the semantics of business rules.

Issue 9955 Replace text

A.2.2 What is a Business Vocabulary?

A business vocabulary contains all the specialized terms, ~~and definitions~~ names, and fact type forms of concepts that a given organization or community uses in their talking and writing in the course of doing business.

The SBVR "Vocabulary for Describing Business Vocabularies" is based on the ISO terminology standards:

- ISO 1087-1 (2000) "Terminology work — Vocabulary — Theory and application" [ISO1087-1]
- ISO 704 (2000) "Terminology work — Principles and methods" [ISO704]
- ISO 860 (1996) "Terminology work – Harmonization of concepts and terms" [ISO860]

These standards have been used for many decades for multilingual correlation of vocabularies in support of language translation work. SBVR is the result of the integration of these ISO standards, formal logics, linguistics, and practical experience from foremost practitioners in the field of business vocabulary for business rules. They have over ten years experience in the development and application of the applied techniques included in the SBVR approach.

There are additional ISO standards for representing basic concepts such as country names and codes (ISO/IEC 3166), dates and times (ISO/IEC 8601), currency codes (ISO/IEC 4217), addresses (ISO/IEC 11180), which are likely to be adopted into vocabularies using SBVR as a matter of practice, but have not been included in this specification.

An SBVR-based model describing a business vocabulary strengthens the semantics of ordinary business glossaries of terms and their definitions in several ways. It provides:

1. A powerful multi-dimensional, hierarchical categorization capability to organize concepts from general to specific such as those used by library/information scientists to index documents. This is often referred to as taxonomies or categorization schemes. The ability to define categories is also included.
2. The capabilities associated with Thesauri including synonyms, abbreviations, 'see also,' multiple vocabularies for

one set of meanings for different languages, etc. The function of the ISO 2788:1986 Monolingual and ISO 5964:1985 Multi-Lingual Thesaurus standards is included in SBVR-based business vocabularies.

3. The ability to specify definitions (both intensional and extensional) formally and unambiguously in terms of other definitions in the business vocabulary as a result of its formal logics and linguistic underpinning.

4. The ability to define connections between concepts that are of interest to the organization. These connections provide the business-level semantic structure required to find information about such relationships in text documents and relational databases, as well as providing the ability to specify business rules formally and unambiguously. The function in the ISO/IEC 13250:2000 “Topic Maps” standard is ~~included in SBVR-based business vocabularies~~ supported by SBVR-based models.

5. A semantically rich set of templates to facilitate capturing the full semantics of each concept and connection between concepts of interest to the business community owning the business vocabulary.

6. A basis for identification and/or definition of individual entities, events and states, the relationships among them, and their relationship to time for text document and data mining.

7. The basis for tools that can support powerful visualization and ‘navigation’ of business vocabularies based on business meaning.

8. Business community ownership and management of their independent business vocabularies and business rules.

9. The basis to integrate separately created business vocabularies, using the ‘characteristic analysis’ capability from ISO 1087-1 and ISO 860. When separate business vocabularies are integrated and the business rules based on them are modified to reflect the vocabulary integration, the business rules will also be integrated.

10. The ability to minimize the number of definitions an organization needs to create by providing powerful, pragmatic features for vocabulary adoption on a well-managed basis. The SBVR approach encourages (a) incorporation of ready-made ‘outside’ vocabularies and (b) communication between people in different communities.

11. A comprehensively integrated capability to support the specification of the meaning of all kinds of business rules.

A.2.3 What is a Business Rule?

Issue 9708 Change text

The SBVR follows a common-sense definition of ‘business rule’:

Business Rule: *rule that is under business jurisdiction*

‘Under business jurisdiction’ is taken to mean that a business ~~enact, revise, and discontinue business rules as it sees fit~~. (or any other semantic community) can, as sees fit, enact, revise, and discontinue the business rules that govern and guide it. If a rule is not under business jurisdiction in that sense, then it is not a business rule. For example, the ‘law’ of gravity is obviously not a business rule. Neither are the ‘rules’ of mathematics.

The more fundamental question in defining ‘business rule’ is the meaning of ‘rule.’ Careful consideration was given to a variety of real-world interpretations of ‘rule,’ including numerous authoritative dictionaries and previously-published works on business rules. Foremost consideration was given to how people think naturally about ‘rule’ in everyday life, not only within business activities, but also outside of them. For example, several rule books for professional sports were reviewed.

Clearly, ‘rule’ carries the sense of ‘guide for conduct or action’ both in everyday life and in business. In one way or another, this sense of ‘rule’ can be found in most, if not all, authoritative dictionaries.

Examining the question more closely, it is obvious that if rules are to serve as guides for conduct or action, they must also provide the actual criteria for judging and guiding that conduct or action. In other words, for the context of business rules (and probably in most other contexts), rules serve as *criteria* for making decisions. The SBVR's interpretation of 'rule' therefore encompasses the sense of 'criteria' as given by authoritative dictionaries.

This point is fundamentally important for professionals creating business models. In business process engineering, for example, the most prevalent understanding of 'business rule' is as criteria for decision points ('branch points') in business process models. Often such decision points are relatively simple (for example, "do we treat a customer as gold level, silver level, or bronze level?"). In other cases, such decision points may be highly complex (for example, "should an insurance claim be paid, denied, or considered as possibly fraudulent?"). For these more complex cases in particular, special inferencing techniques are quite likely to be helpful (for example, tools supporting 'production *rules*').

A.2.3.1 Rules and Formal Logic

Issue 9721 Revise text

An additional and no less important driver in the SBVR's treatment of 'rule' is consistency with formal logics. Notable experts in this area recommended that the best treatment for the SBVR's interpretation of rules would involve *obligation* and *necessity* claims.

Consequently, in SBVR, a Rule is "an element of guidance that introduces an obligation or a necessity." The two fundamental categories of Rule are:

- **Structural Rule** ~~(necessities)~~: These are rules about how the business chooses to organize (i.e., 'structure') the things it deals with. Structural Rules supplement definitions (for example, from EU-Rent):

  Necessity: A Customer has at least one of the following:

  - a Rental Reservation.
  - an in-progress Rental.
  - a Rental completed in the past 5 years.

- **Operative Rule** ~~(obligations)~~: These are rules that govern the conduct of business activity. In contrast to Structural Rules, Operative Rules are ones that can be *directly* violated by people involved in the affairs of the business (for example, from EU-Rent):

  ~~Obligation:~~ A Customer who appears intoxicated or drugged must not be given possession of a Rental Car.

A.2.3.2 Rules, Fact Types and Concepts expressed by Terms

Informally, a fact type is an association[2] between two or more concepts; for example "Rental Car is located at Branch."

In SBVR, rules are always constructed by applying necessity or obligation to fact types. For example, the rule "A Rental must not have more than three Additional Drivers" is based on the fact type "Rental has Additional Driver."

By this means, SBVR realizes a core principle of the Business Rules Approach at the business level, which is that "Business rules build on fact types, and fact types build on concepts as expressed by terms." This notion is well-documented in published material by foremost industry experts over the past 10 years.

The Business Rules Approach is summarized in Annex B.

2. "Association" is used here in its everyday, business sense - not the narrower, technical sense that would apply to a UML class model.

One important consequence of the SBVR's approach in this regard is that concepts (including fact types) are *distinct* from rules, which are in a separate Compliance Point. This design permits SBVR's support for concepts (including fact types) to be optionally used on its own for building business vocabularies.

Issue 9477 Replace text
Issue 9475/9945 Replace text

A.2.3.3 What 'Practicable' Means

All business rules (and ~~affirmations and admonitions~~ advices as well) need to be practicable. Whether or not some element of guidance is practicable is decided with respect to what a person with legitimate need can understand from it.

- For an operative business rule, this understanding is about the behavior of people and what form compliant behavior takes. Because an operative business rule is practicable, a person who knows about it can decide directly whether it is being followed when that person observes relevant behavior.
- For a structural rule, this understanding is about how evaluation of the criteria vested in the rule always produces some certain outcome(s) for a decision or calculation as opposed to others. If a structural rule is practicable, a person who knows about it can also decide directly whether it is being followed when that person observes some relevant outcome from a decision or calculation.

A practicable business rule is also always free of any indefinite reference to people (e.g., "you," "me"), places (e.g., "here"), and time (e.g., "now"). By that means, if the person is displaced in place and/or time from the author(s) of the business rule, the person can read it and still fully understand it, without (a) assistance from any machine (e.g., to "tell" time), and (b) external clarification.

All these criteria assume that the person understands the business concepts that underlie the business rule. A practicable business rule always imparts ready-to-apply knowledge of the kinds above 'on top' of such concepts.

An important best practice for business rules, following naturally from this, is that the underlying business vocabulary/ies must be well developed and well managed. Specifically, each business concept should:

- Be individually well defined.
- Fit logically into the overall structure of concepts.
- Be made available to the person in appropriate manner.

In addition, each business rule should be directly expressible in the given business vocabulary/ies. These best practices point toward the essential role of business vocabularies in supporting business rules – indeed, the bulk of SBVR is devoted to that area.

A.2.3.4 Business Rules that Cannot Be Automated

Just because business rules are practicable, this does *not* imply they are always automatable. Many business rules, especially operative business rules, are *not* automatable in IT systems. For instance, consider the obligation example given above, "A Customer who appears intoxicated or drugged must not be given possession of a Rental Car."

This distinction is not important within SBVR, which focuses on rules only from the business perspective, regardless of whether the rules could be automated. However, it is obviously important in defining a transformation from business model to PIM. In particular, non-automatable business rules need to be implemented as user activity, supported by procedure manuals or rulebooks.

A.2.3.5 What 'Directly Enforceable' Means

All operative business rules need to be directly enforceable. To be enforceable, an operative business rule has to be defined in such a way that violations can be detected. The enforcement regime can then detect a violation and take appropriate action (e.g., correct the violation, notify other parties, and/or impose penalties on the violators).

Elements of governance directly govern what people do in the business, and they need to be enforceable. Being **directly** enforceable is what distinguishes business policies from operative business rules. The importance of this is that when the people specifying a business encounter (or need to define) elements of governance in the real world, they need to think about two things.

First, is the element of governance directly enforceable – i.e., is it possible to observe what people are doing, and recognize whether they are complying or not, without needing further amplification or explanation of the element of governance? If it is not, then the element of governance is a business policy and those who are defining the business haven't yet finished. They also need to develop operative business rules, derived from the business policy, that are directly enforceable.

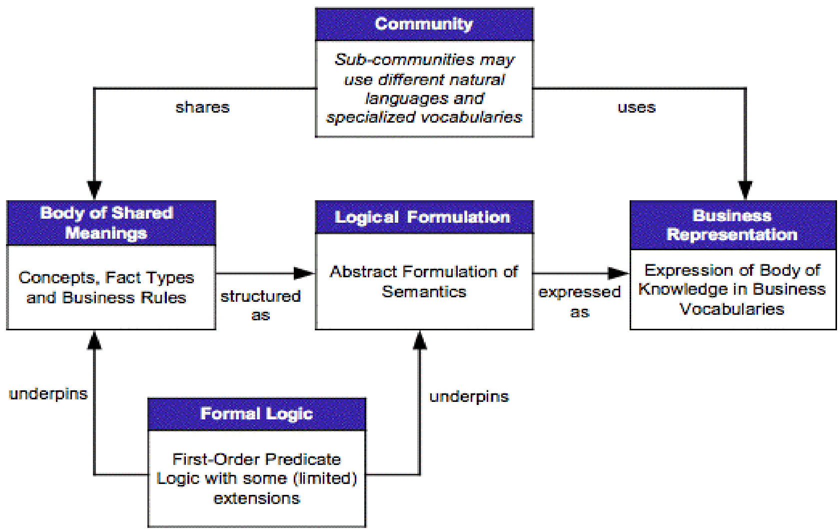
For example, the EU-Rent element of governance 'rental cars must not be exported' is not sufficiently precise to be enforced. It is a business policy and needs operative business rules through which it can be enforced. For example:

- Each rental car must be registered in the country of the local area to which it is assigned after purchase.
- The country of registration of a rental car must not be changed.
- If a car is at a location outside its country of registration, it may be assigned only to a rental with return location in its country of registration.
- If a rental car is at a location outside its country of registration for more than five days, it must be returned to its country of registration.

Second, if an element of governance is directly enforceable, it ought to be derived from a business policy. If it is not, the business designers ought to be aware that this is so (and might choose to question whether the rule is appropriate).

A.3 Informal Overview of SBVR

SBVR can be viewed as having five major aspects, as illustrated below:



A.3.1 Community

The basis for business vocabulary is community. At the business level, communities of primary importance are enterprises for which business rules are being established and expressed. However, other communities - the industry in which an enterprise operates, partner enterprises, standards groups, regulatory authorities, etc. - also need to be recognized.

An important aspect of community is that sub-communities within an enterprise may need its body of shared meanings (starting with fundamental concepts) to be expressed in different vocabularies, ranging from specialized jargon to different natural languages. In SBVR, such sub-communities are called "speech communities."

A.3.2 Body of Shared Meanings

Issue 9958 Replace form of expression with fact type form
Issue 9948 Revise text

A community has a body of shared meanings, comprising concepts (which include fact types) and business rules. What is shared is the meaning, not the fact type form ~~form of expression~~. Clearly, for shared meanings to be exchanged, discussed, and validated, they must be expressed. But SBVR separates the business meaning from any particular fact type form ~~form of expression~~. The structure of the body of shared meanings (i.e., which concepts instances play which roles in facts, which facts form the basis of which rules, etc.) is defined by associating abstract concepts, fact types, and business rules, not by associating statements in any given language.

A.3.3 Logical Formulation

Logical formulation provides a formal, abstract, language-independent syntax for capturing the semantics of a body of shared meanings. It supports multiple forms of representation, such as: noun and verb fact type forms ~~forms of expression~~, reading of associations in both directions.

Logical formulation supports two essential features of SBVR. First is the mapping of a body of shared meanings to vocabularies used by communities. Second is the mapping to XMI that enables interchange of concepts, facts, and business rules between tools that support SBVR.

Issue 9952 Revise text

A.3.4 Business Representation

The concepts and business rules in a body of shared meanings need to be represented in vocabularies acceptable to, and usable by, speech communities that share their meaning. These vocabularies may be in different natural languages, in artificial languages such as the UML, or in specialized subsets of natural languages, as used by, for example, engineers or lawyers.

SBVR supports mapping of business meaning to concrete language by ~~associating~~ representing elements of the body of shared meanings with signifiers. Examples of these representations are terms such as “customer,” “car,” “branch” for noun concepts, and ~~fact symbols~~ designations (often verb phrases) such as “rents,” “is located at” for fact types. ~~Logical formulations provide the structure, and signifiers are placed in logical formulations to provide the expression.~~ Designations are used in statements and definitions whose logical formulations are structures of business meaning.

SBVR supports adoption from external sources, such as standards bodies and industry groups. For example, SBVR itself adopts some of its basic definitions from ISO standards for terminology and vocabulary (ISO 1087-1 and ISO 704).

A.3.5 Formal Logic

SBVR has a sound theoretical foundation of formal logic, underpinning both logical formulation and the structures of bodies of shared meanings. The base is first-order predicate logic (with some restricted extensions into higher-order logics), with some limited extensions into modal logic – notably some deontic forms, for expressing obligation and prohibition, and alethic forms for expressing necessities.

A.4 SBVR Beneficiaries

A different perspective of SBVR is provided by considering the different groups of people who will benefit from it.

A.4.1 Business Analysts and Modelers

Business analysts and modelers work in enterprises such as EU-Rent. Their business view is the enterprise business view, or perhaps a view of part of the business.

Their view of Community is generally the enterprise in which they work, and its Speech Communities. Within this, they are most concerned with building on the enterprise’s Body of Shared Meanings and Vocabulary in which to express it. They have to negotiate with the Integrators/Administrators (see next subclause) for inclusion of new concepts and business rules and new signifiers in the Vocabularies.

Business analysts and modelers need to specify business policies and rules precisely, but to do so they do not need any in-depth knowledge of SBVR’s Logical Formulation or Formal Logic. They will see the effects of these parts of SBVR in facilities provided by tools that support their enterprise’s business vocabularies and rules, e.g., templates, options, constraints, consistency checks.

A.4.2 Business Vocabulary+Rules Integrators/Administrators

Business Vocabulary+Rules integrators/administrators generally work within enterprises. Their business view is maintaining a consistent enterprise-wide Body of Shared Meanings, plus Vocabularies for Speech Communities within the enterprise.

They are responsible for integrating and quality-assuring content provided by business analysts and modelers. An important part of this is deciding what to adopt from external vocabularies. They will also be responsible for maintaining the Business Vocabulary+Rules over time. This is outside the scope of SBVR; Business Rule Management is a separate issue to be addressed by the OMG as appropriate.

Integrators/administrators will generally be more aware than business analysts and modelers of Logical Formulation. However they do not need to understand it formally: they will see its effects in administration tools.

A.4.3 Tool Builders

Issue 9941 Replace terms, fact type symbols with designations, fact type forms
Issue 9955 Replace text

Two kinds of tool will be needed to support SBVR:

- For interchange of business vocabularies and rules between different platforms.
- For developing and maintaining business vocabularies and rules for a community.

Interchange standards (and tools that use them) are of great importance to the OMG. Compliance with MOF and XMI was mandated by the OMG, and its achievement is a major part of SBVR. Developers of interchange tools will have four major concerns:

- The types of construct in a Body of Shared Meanings – Concepts, Fact Types, Facts and Business Rules - and the types of relationship between them.
- The association of elements of the Body of Shared Meanings with elements of Vocabulary – ~~terms,~~ fact symbols, fact type forms, definitions, references to external sources.
- Logical Formulation.
- Mapping to MOF/XMI.

Issue 9930 Remove text

The developers will not be concerned with the content of Business Vocabulary+Rules for enterprises. And although tool architects and designers will need to understand the Formal Logic theory underpinning of SBVR, the developers will not (although it should be reassuring that it is there). ~~For further discussion see Annex L~~.

Business analysts and modelers and integrators/administrators will need tools for developing and maintaining enterprise Business Vocabulary+Rules.

Development of such tools is not the direct concern of the OMG; they will be developed by vendors to meet market demand. However, it is important that they are developed – it would be futile to have good interchange standards and tools if nobody was developing worthwhile content for interchange.

Ensuring that the SBVR model will provide a sound basis for development and maintenance tools has been a judgment call by the BRT. Tools will need to support Body of Shared Meaning, Business Expression and Logical Formulation, plus multiple Communities and vocabulary adoption between them. Tool developers will also have to work with methodologists to ensure support of processes for development and integration of Business Vocabulary+Rules.

A.4.4 Logicians, Semanticists, and Linguists

Logicians, semanticists, and linguists provide the logical, mathematical, and linguistic capabilities that make it possible to transform business vocabularies and rules from the business perspective to PIM and PSM information systems designs, to structure a variety of natural language statements into SBVR constructs, and to verbalize SBVR entries into any number of natural language statements.

They design the algorithms to ensure integrity in Business Vocabulary+Rules interchange documents, and in the translation between interchange documents and internal tool designs. They also help ensure the formal logic, mathematatic, and linguistic integrity of the internal designs of Business Vocabulary+Rules tools.

Issue 9955 Replace text

A.4.5 Summary of Audiences (Business Beneficiaries) by Activity and Business Context

Business Context *(excluding recordkeeping & information system activities)*

- Creating Business Content in a 'Business Vocabulary+Rules' (e.g., EU-Rent)

  Audience: Business People in General

- Integrating & Quality Assuring Business Content in a 'Business Vocabulary+Rules' (e.g., EU-Rent)

  Audience: 'Business Vocabulary+Rules' Integrator/Administrators

'Business Vocabulary+Rules' Technology and Tool Context

- Providing the Semantic and Logical Foundation for all 'Business Vocabulary+Rules'

  Audience: Linguists, Semanticists, and Logicians

- Designing a 'Business Vocabulary+Rules' Tool for Business People to Document Business Content (e.g., EU-Rent)

  Audience: Designers of vocabulary and rules software tools for business people

- Designing Tool capability to interchange Business Content in a 'Business Vocabulary+Rules' (e.g., EU-Rent) among Business Communities within and between Organizations

  Audience: Infrastructure Designers for Business Vocabulary and Rules Tools

Information System (Recordkeeping) Context *(Out of Scope for SBVR)*

- Designing Information Systems that Talk and Work according to the Business Content in a 'Business Vocabulary+Rules" (e.g., EU-Rent)

  Audience: Designers of information systems that support business vocabularies or automate business rules

A.5 Technical Overview of the Approach

SBVR is designed to support interchange of business vocabularies and rules among organizations. SBVR is conceptualized optimally for business people and designed to be used for business purposes independent of information systems designs.

It is also intended to provide the business vocabulary and rules underpinned by First Order Predicate Logic for transformations by IT staff into information system designs. Note that, in most cases, such transformations will not be fully automated; there will be many options for information system design, with decisions required from system architects and PIM modelers.

A.5.1 How SBVR is Underpinned by Formal Logics

The formal semantics of SBVR is based on the following formal approaches: typed predicate logic; arithmetic; set and bag comprehension (grounded in ur-elements), with some additional basic results from modal logic. The logic is essentially classical logic, so mapping to various logic-based tools should be straightforward. Typed logic is used for convenience but is easily translatable into untyped logic.

SBVR is neutral as to whether types may be instances of other types in the same model. We provide a basic formalization in first-order logic for those who wish to exclude higher-order types. We also provide an extended formalization for those who wish to allow higher-order types. The extended formalization uses a restricted version of higher-order logic that is closely related to Henkin semantics in restricting the range of types over which quantification is permitted. In first-order logic, quantification is permitted only over individuals (objects: lexical or non-lexical). The SBVR's restricted higher-order formalization also allows quantification over at least one (one may choose either or both) of the following: object types that are instances of a declared categorization type (whether or not these instances have been explicitly declared); object types (primitive or derived) that are explicitly declared in the schema.

It is well known that any function may be rewritten as an equivalent relation, and vice versa. For simplicity, SBVR treats all functions (including mathematical operations) as relations. Relations may be of any arity (1, 2, 3, etc.).

SBVR has no dependency on artificial identifiers (such MOF ids, surrogate keys), so that all individuals are identified by definite descriptions that are ultimately grounded in lexical constants (note that this does not prevent businesses from using artificial identifiers within their specific SBVR models). Individual constants may be introduced by definition as a shorthand for definite descriptions. Unnamed structures are permitted. For example, sets may be identified by their extensions, and formulae may be identified by their structural composition. The avoidance of artificial identifiers ensures that business statements may be easily understood and communicated between businesses. This is not to discourage the use of names, which is highly recommended, but only to cater for cases where they are not supplied. This also does not prohibit the use of artificial identifiers by supporting tools, provided that such identifiers are hidden from business users of such tools.

Modal operators used include the alethic operators 'It is necessary that,' and 'It is possible that,' and the deontic operators 'It is obligatory that,' and 'It is permissible that.' Other modal operators are allowed at the surface level but are translated into these more basic operators with the help of negation (e.g., 'It is forbidden that' is captured internally as 'It is obligatory that it is not the case that'). Apart from standard modal operator transformations involving negation, no other use is made of modal logic theorems, so there is no requirement to choose one out of many specific modal logics for a given modality.

The term 'fact' is used in the sense of epistemic commitment, but the underlying logic used for logical connectives is isomorphic to standard truth-functional logic rather than epistemic logic. Ultimately all ground facts are existential or elementary. The truth functional logic is two-valued, with negated existential formulae being used to avoid the use of null values.

A.5.2 SBVR Inherent Extensibility

Issue 9930 Remove text / revise text
Issue 9941 Replace symbols with designations and fact type forms
Issue 9955 Replace text

1. The SBVR Vocabularies given by this specification are themselves vocabularies that can be included in other business vocabularies. An extended SBVR vocabulary can be created by including an SBVR vocabulary into another business vocabulary that has other ~~symbols~~ designations. An extended SBVR vocabulary can, for example, provide for expression of additional information about ~~symbols~~ designations and rules that is not covered by this specification. An extended SBVR vocabulary can add new ~~symbols (terms, names, and sentential forms)~~ designations

and fact type forms for existing concepts as well as add new concepts along with ~~symbols~~ designations that represent them.

2. The SBVR Vocabularies given by this specification are based on the English language, but can be used to define vocabularies in any language. Alternative SBVR vocabularies based on a different language can be defined by providing ~~symbols~~ designations from the different language for the concepts represented in the SBVR Vocabularies.

   ~~The Vocabulary-to-MOF/XMI Mapping Rule Set provided with this specification can be applied to any extended SBVR vocabulary in order to produce a repository model and an XML schema that can extend the repository models and XML schemas, respectively, that are provided with this specification (see Annex L). XML documents formed according to that schema can accommodate facts expressible in the extended business vocabulary.~~

3. The SBVR Vocabularies are used to express rules in this specification concerning the definition of business vocabularies and formation of business rules. The SBVR Vocabularies can be further used to express other rules or to form expressions for other purposes. Such other rules can stipulate additional requirements concerning, among other things, what constitutes valid business vocabularies and what is allowable and required in the expression of rules. This specification describes how such rules, like other rules, are formally modeled and communicated and makes no requirement concerning enforcement of such additional rules.

Use of an SBVR vocabulary outside this specification (as in 1 through 3 above) does not change the SBVR vocabulary itself, but only uses it by way of reference.

A.5.3 MOF/XMI Models for SBVR

Issue 9930 Replace text / revise text

A business vocabulary provides a means of recording and communicating facts. Following OMG's Model Driven Architecture, a business vocabulary developed as an information system independent model of business communication is used to drive the creation of a platform independent MOF model. The MOF model is, in turn, used to drive generation of Java interfaces (based on JMI) and an XML schema (based on XMI).

SBVR is mapped to MOF in two ways. First, the SBVR Vocabularies are mapped to a MOF-based model of repositories that can hold representations of facts that can be meant by any atomic formulation expressible using the business vocabulary. This first mapping does not capture the full SBVR with all of its semantics. It only maps the business vocabulary, using MOF as a mode of representation. ~~The creation of this model is guided by mapping rules given in Clause 13 Part II~~. The metamodel is described in Clause 13.

Second, the full SBVR is captured in terms of the MOF-based model created from the SBVR Vocabularies (the first mapping). This includes the definitions of concepts, terms, business rules and other facts of the SBVR Metamodel that are expressed in terms of the SBVR Vocabularies.

~~The rules that guide the first mapping are general enough to be used for any vocabulary defined in terms of SBVR. A vocabulary for any business domain can be mapped to a MOF repository model by these same rules.~~

~~The MOF repository model mapped from the SBVR Vocabularies, which is the model used to capture the SBVR itself, is also used to capture business vocabularies and business rules in general. For additional detail on the design rationale, see Annex K.~~

A.6 Special Features of SBVR

A.6.1 Coherent Business Example: EU-Rent

It is valuable to have a common, consistent base for a large body of examples to illustrate the SBVR approach and use of the SBVR Metamodel. SBVR uses EU-Rent, a (fictitious) car rental company that has been used in several other R&D projects and publications, including papers published by the Business Rules Group. EU-Rent was also used as the basis for the *Business Rules Product Derby*, held at the Business Rules Forum in (New Orleans, 2002, Nashville, 2003, and Las Vegas, 2004), and as the common case study for vendors at the European Business Rules Conference (Zurich, 2003, and Amsterdam, 2004).

EU-Rent includes a broad range of concepts, facts and rules. Most readers of this specification should find the business requirements easy to understand. They should be able to move into the detail of the examples without having to spend much time on the general business scenario.

An important feature of EU-Rent is that it is an international business, which has requirements for expression in different natural languages, and for adaptation of some polices and rules to local regulation, custom, and practice.

A.6.2 Internationalization

Internationalization is handled from two directions. First, the meanings of concepts (including fact types) and rules within a body of shared meanings are modeled separately from how they are expressed. The same meaning can be expressed in different languages, both natural and artificial (such as UML and XML).

Second, communities who define concepts and set rules can be grouped and associated. An international company could, for example, define core concepts. Each of its regional divisions would adopt the core into its local body of shared meanings, which also addressed adaptation to local regulation, custom, and practice.

The resulting content could then be mapped into different languages. For example, global policies and rules could be expressed globally in a common language such as English, but operational detail mapped to as many languages as are needed. Communities can also adopt business vocabularies, so that the Swiss division could adopt business vocabularies developed and maintained by the French, German and Italian divisions. SBVR uses “ISO 639-2 Codes for the Representation of Names of Languages” [ISO639-2] to specify the language used to express a given vocabulary (see Part II entry for ‘language’).

One issue still to be addressed in internationalization concerns adoption of business vocabularies from outside the business. Adoption of such business vocabularies, e.g., from trade associations or special interest groups, has two major advantages: it reduces the work needed to maintain the adopting company’s own vocabulary, and it eases communication with other organizations in the same business area. If such business vocabularies are adopted in different natural languages for the same meaning there is some risk of inconsistency in the mappings. The issue that needs further discussion is the trade-off between:

- Adopting an externally-defined vocabulary and supplementing it as needed
- Modifying an externally-defined vocabulary to create a new one and taking on the overhead of maintaining the modifications

The outcome is likely to be heuristics to be applied case by case, rather than a general recommendation one way or the other.

Issue 9955 Remove text

A.6.3 Independence

Rule Independence. SBVR bases the expression of all business rules on ~~structured~~ business vocabularies. By doing so, business rules can be specified independently of all processes and events.

Enforcement. SBVR carefully segregates business rule specification from any aspect of enforcement.

Methodology and Notation. Although proven compatible with both existing notations and new innovative visualization techniques, SBVR is completely neutral with respect to methodology or notation, permitting the widest possible adoption.

A.6.4 Notations for Business Vocabulary+Rules

A.6.4.1 Special Note on Notations

'Notation' is used in SBVR (as instructed by OMG) to mean any language used to represent semantics, or more precisely, abstract syntax. Notations can be verbal, graphical or any combination thereof. Other words for 'notation' are 'gramma,' 'syntax,' and 'concrete surface syntax.'

It is specifically *not* the intention of SBVR to mandate any particular notation(s) that must or should be used with the SBVR Metamodel. Indeed, this would be neither productive not desirable. Instead, SBVR encourages wide innovation, experimentation, and value-adding software development in the area of compliant notations.

A.6.4.2 SBVR Structured English

It should be remembered that SBVR Structured English (presented in Annex C) is just one of possibly many notations that can be used to express the SBVR Metamodel, and, as a notation, is nonnormative in the SBVR standard. Indeed, additional compliant notations are welcomed and encouraged.

Compliant enrichments of various parts of SBVR Structured English itself are also welcomed and encouraged.

Two styles of SBVR Structured English are documented in this specification:

1. Prefixed Rule Keyword Style
2. Embedded (mixfix) Rule Keyword Style

The Prefix Style introduces rules by prefixing a statement with keywords that convey a modality. Examples of some of the prefixes are shown in the table below.

Operative	**Structural**
It is obligatory that	It is necessary that
It is prohibited that	It is impossible that
It is permitted that	It is possible that

This style, which is explained in Annex C, is included in this specification for two primary reasons:

- It is supported by the commercial reference implementation of Unisys Corporation, an implementation that satisfies the OMG submission's compliance requirements.
- Its rule keywords correspond to the modal operators in the logical formulation portion of SBVR, so it illustrates the translation of notation to metamodel in the most direct and easy-to-understand fashion.

The Embedded Style features the use of rule keywords embedded (usually in front of verbs) within rules statements of appropriate kinds. Examples of some of the embedded keywords are shown in the table below.

Operative	**Structural**
... must ...	... always ...
... must not ...	... never ...
... may ...	... sometimes ...

This style of notation, which is introduced in Annex F and examined more closely in Annex I, is included in this specification for two primary reasons:

- It is an existing, documented notation[3] (RuleSpeak®, by Business Rule Solutions, LLC) that has been used with business people in actual practice for a number of years.
- It clearly demonstrates that alternative notations for business rules, which some business people find more natural and/or friendly, are easily accommodated under SBVR Structured English.

A.6.5 State

'State' is an important notion for business vocabularies and business rules. As far as business people are concerned, 'state' is a concept they can refer to and use in creating definitions, facts, and rules. For example, in EU-Rent a car's states would include: 'available,' 'allocated to rental,' 'on rental,' 'damaged,' and so on. The company uses these state names in defining business rules, e.g., "The car assigned to a walk-in rental must be the available car with the lowest odometer reading in the requested car group." One way to express states is using unary predicates, e.g., "car is available."

Businesses name only those states that are useful to them, and these may be only a small subset of the real-world states that real-world cars may have. For example, a car will, early in its EU-Rent life, have a state 'just delivered and checked out, ready for its first rental.' But EU-Rent can decide that this has no practical difference from 'returned from rental, cleaned and refueled' and combine the two (with others, like 'transferred in from another branch') into a named state called 'available.'

The SBVR approach to Business Vocabulary+Rules regards state as largely definitional ('available' is the concept we use for a car that is …), unlike in a system design or implementation, where state handling is often about applying rules to data ("when a car is returned from a rental, its state must be set to 'available'"). And selection of the states that are useful to name and define is a business decision.

States are associated with other kinds of concept, including concepts that represent:

- things in the business (like cars and rentals).
- things happening in the business (like rental reservation, late return from rental).
- other states ("when a car is in state 'due for service' it cannot become 'available' again until it has been serviced -- i.e., been through the pattern of events that describe servicing").

'State' may need some further development; for example, *dynamic* models of events, cycles, schedules, etc. were considered to be outside the scope of SBVR. As SBVR is, states can be represented using concepts and fact types.

3. [Ross2003], Clauses 8-12

Annex B
(informative)

The Business Rules Approach

SBVR provides a formal foundation for business rules. It also defines what they are. Much of the thinking in this area arose from the work of the Business Rules Group, which has been working exclusively in the area since the late 1980s.

Key notions of the business rules approach are presented succinctly by the BRG's *Business Rules Manifesto*. An extract from the Manifesto is presented below, to assist readers in positioning some of the central notions of SBVR. This brief extract is followed by a figure providing an overview of SBVR support.

A brief word on the BRG follows, along with citations to its work products. The full text of the Business Rules Manifesto[1] can be found in numerous languages at: http://www.businessrulesgroup.org/brmanifesto.htm.

B.1 Extract from the Business Rules Manifesto

Primary Requirements, Not Secondary. Rules are essential for, and a discrete part of, business models and technology models.

Separate From Processes, Not Contained In Them. Rules apply across processes and procedures. There should be one cohesive body of rules, enforced consistently across all relevant areas of business activity.

Deliberate Knowledge, Not A By-Product. Rules build on facts, and facts build on concepts as expressed by terms. Terms express business concepts; facts make assertions about these concepts; rules constrain and support these facts. Rules are basic to what the business knows about itself — that is, to basic business knowledge. Rules need to be nurtured, protected, and managed.

Declarative, Not Procedural. Rules should be expressed declaratively in natural-language sentences for the business audience. A rule is distinct from any enforcement defined for it. A rule and its enforcement are separate concerns.

Well-Formed Expression, Not Ad Hoc. Business rules should be expressed in such a way that they can be validated for correctness by business people. Business rules should be expressed in such a way that they can be verified against each other for consistency.

For the Sake of the Business, Not Technology. Rules are about business practice and guidance; therefore, rules are motivated by business goals and objectives and are shaped by various influences. The cost of rule enforcement must be balanced against business risks, and against business opportunities that might otherwise be lost.

Of, By, and For Business People, Not IT People. Rules should arise from knowledgeable business people.

Managing Business Logic, Not Hardware/Software Platforms. Rules, and the ability to change them effectively, are fundamental to improving business adaptability.

1. [BRM].

B.2 An Overview of SBVR Support for Key Business Rule Ideas

A core idea of business rules formally supported by SBVR is the following from the Manifesto: "Rules build on facts, and facts build on concepts as expressed by terms. Terms express business concepts; facts make assertions about these concepts; rules constrain and support these facts."

This core idea, originating in the BRG's seminal 1995 white paper [BRG2002], has been called the business rules "mantra." It is often abbreviated for convenience to simply: *"Rules are based on facts, and facts are based on terms."*

Figure B-1 provides an overview of how SBVR supports the "mantra." It requires separation of viewpoints as follows.

Business Rule "Mantra." An approximation that simplifies explanation for business people and others new to the approach.

Representation (in SBVR terminology). The SBVR notions that classify the words that people use to express their vocabulary+rules.

Meaning (in SBVR terminology). The SBVR notions that classify the underlying meaning of the words that people use in expressing their vocabulary+rules.

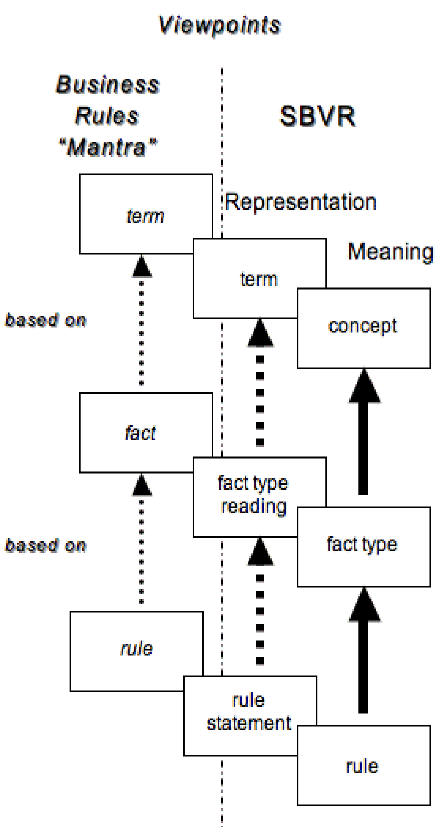


Figure B.1 - How SBVR Supports the Business Rules "Mantra"

B.3 About the Business Rules Group (BRG[2])

Background. Information Systems analysts have long been able to describe an enterprise in terms of the structure of the data the enterprise uses and the organization of the functions it performs. Unfortunately, there is often neglect of the rules (constraints and conditions) under which the enterprise operates.

Frequently these rules are not articulated until it is time to convert them into program code. While rules that are represented by the structure and functions of an enterprise have been documented to a degree, others have not been articulated well, if at all. The Business Rules Group was organized to carry out that articulation.

The BRG Charter. Originally a project within GUIDE International, the Business Rules Group has been an independent organization since the 1990s. Its membership comprises experienced practitioners in the field of systems and business analysis methodology who work in both the public and the private sectors.

The charter of the BRG is to formulate statements and supporting standards about the nature and structure of business rules, the relationship of business rules with the way an enterprise is organized, and the relationship of business rules with systems' architectures.

2. [BRJ2005]

Annex C
(informative)

SBVR Structured English

Issue 9930 Remove text
Issue 9941 Replace preferred symbol with preferred designation
Issue 9955 Replace text

The most common means of expressing definitions and business rules is through statements, not diagrams. While diagrams are helpful for seeing how concepts ar related, they are impractical as a primary means of defining vocabularies and expressing business rules.

This specification defines an English vocabulary for describing vocabularies and stating rules. There are many different ways that this vocabulary and other English vocabularies described using SBVR can be combined with common English words and structures to express definitions and statements. However expressed, the semantics of definitions and rules can be formally represented in terms of the SBVR vocabulary and, particularly, in terms of logical formulations (the SBVR conceptualization of formal logic).

This annex describes one such way of using English that maps mechanically to SBVR concepts. It is not meant to offer all of the variety of common English, but rather, it uses a small number of English structures and common words to provide a simple and straightforward mapping.

All formal definitions and rules in this document that are part of 'SBVR in terms of itself' ~~(or of the other vocabularies and rules, such as those for mapping to MOF and XMI)~~ are stated using the SBVR Structured English. These statements can then be interpreted automatically in order to create MOF and/or XMI representations.

The description of the SBVR Structured English is divided into subclauses.

- Expressions in SBVR Structured English
- Describing a Vocabulary
- Vocabulary Entries
- Specifying a Rule Set
- Guidance Entries

C.1 Expressions in SBVR Structured English

Issue 9721 Remove text

This document contains numerous statements and definitions that represent corresponding logical formulations. These statements are recognized by being fully expressed using the fonts listed below. Note that these fonts are also used for individual designations in the context of ordinary, unformalized statements in order to note that defined concepts are being used.

There are four font styles with formal meaning:

term — The ‘term’ font is used for a designation for a noun concept (other than an individual concept), one that is part of a vocabulary being used or defined (e.g., ~~modality,~~ modal formulation, fact type). This style is applied to the designation where it is defined and wherever it is used.

Terms are usually defined using lower case letters unless they include a proper noun. Terms are defined in singular form. Plural forms are implicitly available for use.

Name — The ‘name’ font is used for a designation of an individual concept — a name. Names tend to be proper nouns (e.g., California). This style is applied to a name where it is defined and wherever it is used. Note that names of numerical values in formal statements are also shown in this style (e.g., 25). See the definition of ‘name’ for more details.

Names appear using appropriate capitalization, which is usually the first letter of each word, but not necessarily.

Issue 9467 Revise/remove text
Issue 9958 Replace Forms of expressions with Fact type forms
Issue 9721 Replace text
Issue 9621 Replace text

---

verb — The ‘verb’ font is used for designations for fact types — usually a verb, preposition, or combination thereof. Such a designation is defined in the context of a fact type form ~~form of expression~~. This font is used both in the context of showing a fact type form ~~form of expression~~ (e.g.,

~~‘modal formulation *claims* modality’~~
‘reference scheme *is for* concept’

~~and~~

~~‘modality *is claimed by* modal formulation’)~~

and in the context of using it in a statement (e.g.,

~~“Each modal formulation *claims* exactly one modality.”).~~
“Each reference scheme *is for* at least one concept.”

See the definition of ‘fact type form’ ~~form of expression~~ in Part II for more details.

~~Forms of expressions~~ Fact type forms shown as vocabulary entries use singular, active forms of verbs with the exception that present particples are sometimes used for characteristics. ~~are defined using singular, active forms of verbs with the exception that present participles are sometimes used for characteristics.~~ Infinitive, subjunctive, passive, and plural forms of verbs are implicitly usable in statements and definitions. For a binary fact type, the implicit passive form of a verb uses the past participle of the verb preceded by the word “*is*” and followed by the preposition “*by*.” For example, the implicit passive form of ~~‘modal formulation *claims* modality’~~ ‘expression *represents* meaning‘ is ~~‘modality *is claimed by* modal formulation’~~ ‘meaning *is represented by* expression‘. The same pattern holds for fact types with more than two roles where a verb is used between the first two placeholders. For example, the implicit passive form of ‘thing *fills* role *in* actuality‘ is ‘role *is filled by* thing *in* actuality‘. Note that there is no inverse implication of an active form from a passive form.

keyword — The ‘keyword’ font is used for linguistic symbols used to construct statements – the words that can be combined with other designations to form statements and definitions (e.g., ‘each’ and ‘it is obligatory that’). Key words and phrases are listed below.

Quotation marks are also in the ‘keyword’ font. The text within quotes is in ordinary font if the meaning of the quotation is uninterpreted text. The text within quotes is in styled text if the meaning of the quotation is formally represented. Single quotation marks are used to quote a designation or fact type form ~~form of expression~~ that is being mentioned. If a designation is mentioned (where the designation is itself the subject of a statement) it appears within single quote marks (e.g., ~~'modality'~~ 'actuality' and 'California' used to talk about those designations). Single quotes are also used around a fact type form ~~form of expression~~ that is being mentioned (e.g., ~~'modal formulation *claims* modality'~~ 'reference scheme *is for* concept' used to talk about that fact type form ~~form of expression~~). Double quotation marks are used in other cases, such as to quote a statement.

Single quotation marks are also used to mention a concept – to refer to the concept itself rather than to the things it denotes. In this case, a quoted designation or fact type form ~~form of expression~~ is preceded by the word 'concept' or by a term for a kind of concept. For example, the statement,

"The concept 'quantification' *is* a category *of* the concept 'logical formulation'"

refers to the named concepts, not to quantifications and logical formulations. A role can be named with respect to a fact type in this same way (e.g.,

~~the role 'modality' *of* the fact type 'modal formulation *claims* modality').~~

the role 'meaning' *of* the fact type 'expression *represents* meaning'."

Periods also appear in the 'keyword' font. A period is used to terminate a statement, but not a definition. Other punctuation symbols (e.g., parentheses, comma) also apply the 'keyword' font when part of a formal expression.

~~The SBVR Structured English uses designations and forms of expressions exactly as they are defined in a vocabulary. Plural forms are not used. For example, a formal statement would say "each concept" rather than "all concepts." Both the active form and the passive form of a verb need to be defined in a vocabulary if both are used.~~

C.1.1 Key words and phrases for logical formulations

Key words and phrases are shown below for expressing each kind of logical formulation. The letters '*n*' and '*m*' represent use of a literal whole number. The letters '*p*' and '*q*' represent expressions of propositions.

C.1.1.1 Quantification

each	universal quantification
some	existential quantification
at least one	existential quantification
at least *n*	at-least-n quantification
at most one	at-most-one quantification
at most *n*	at-most-n quantification
exactly one	exactly-one quantification
exactly *n*	exactly-n quantification
at least *n* and at most *m*	numeric range quantification
more than one	at-least-n quantification with *n* = 2

C.1.1.2 Logical Operations

Issue 9475/9945 Add text

it is not the case that *p*	logical negation
p and *q*	conjunction
p or *q*	disjunction
p or *q* but not both	exclusive disjunction
if *p* then *q*	implication
q if *p*	implication
p if and only if *q*	equivalence (see exception explained under Modal Operations below)
not both *p* and *q*	nand formulation
neither *p* nor *q*	nor formulation
p whether or not *q*	whether-or-not formulation

Where a subject is repeated when using 'and' or 'or' the repeated subject can be elided. For example, the statement, "An implication has an antecedent and the implication is embedded in a modal formulation," can be abbreviated to this: "An implication has an antecedent and is embedded in a modal formulation." Similarly, a repeated subject and verb can be elided. For example, the statement, "An implication has an antecedent and the implication has a consequent," can be abbreviated to this: "An implication has an antecedent and a consequent."

The keyword 'not' is used within an expression before the verb "*is*" as a way of introducing a logical negation. Also, the key words "does not" are used before other verbs (modified to be infinitive) to introduce a logical negation.

C.1.1.3 Modal Operations

Issue 9475/9945Add text
Issue 9721 Replace text
Issue 10443 Replace text
Issue 10569 Remove text

it is obligatory that *p*	obligation ~~claim~~ formulation
it is prohibited that *p*	obligation ~~claim~~ formulation embedding a logical negation
it is necessary that *p*	necessity ~~claim~~ formulation
it is impossible that *p*	necessity ~~claim~~ formulation embedding a logical negation
it is possible that *p*	possibility ~~claim~~ formulation
it is permitted that *p*	permissibility ~~claim~~ formulation

The following key words are used within expressions having a verb ~~(often modified to be infinitive)~~ to form verb complexes that add a modal operation.

… must … obligation ~~claim~~ formulation

… must not … obligation ~~claim~~ formulation embedding a logical negation

… always … necessity ~~claim~~ formulation

… never … necessity ~~claim~~ formulation embedding a logical negation

… may … permissibility ~~claim~~ formulation

The key word phrase "only if" is used in combination with some of the key words and phrases shown above to invert a modality.

~~… may … only if p~~ ~~obligation claim over an implication~~

~~it is permitted that q only if p~~ ~~obligation claim over an implication~~

~~it is possible that q only if p~~ ~~necessity claim over an implication~~

~~The key word "only" can also be used with "may" in an expression before a preposition to invert a modality.~~

~~… may … only …~~ ~~obligation claim over an implication~~

… may … only if p is equivalent to … must not … if not p

it is permitted that q only if p is equivalent to it is obligatory that not q if not p

it is possible that q only if p is equivalent to it is necessary that not q if not p

For example, the following two statements have the same meaning.

A car may be rented only if the car is available.

A car must not be rented if the car is not available.

The key word "only" can also be used before a preposition in combination with "may" to invert a modality. The noun phrase after the preposition is then understood as a negated restriction as shown in these two equivalent statements:

A car may be rented only to a licensed driver.

A car must not be rented to a person that is not a licensed driver.

Because of the use of "only" in stating modal operations, the pattern "p if and only if q" for equivalence is not used if p involves a modal operation.

C.1.2 Other Keywords

Issue 10569 Revise text

the 1. used with a designation to make a pronominal reference to a previous use of the same designation. This is formally a binding to a variable of a quantification.

2. introduction of a name of an individual thing or of a definite description

a, an — universal or existential quantification, depending on context based on English rules

another — (used with a term that has been previously used in the same statement) existential quantification plus a condition that the referent thing is not the same thing as the referent of the previous use of the term

a given — universal quantification pushed outside of a ~~demonstrative expression~~ logical formulation where ‘a given’ is used such that it represents one thing at a time – this is used to avoid ambiguity where the ‘a’ by itself could otherwise be interpreted as an existential quantification. Within a definition, ‘a given’ introduces an auxiliary variable into the closed projection that formalizes the definition.

that

1. when preceding a designation for a noun concept, this is a binding to a variable (as with ‘the’)
2. when after a designation for a noun concept and before a designation for a fact type, this is used to introduce a restriction on things denoted by the previous designation based on facts about them
3. when followed by a propositional statement, this used to introduce nominalization of the proposition or objectification, depending on whether the expected result is a proposition or an actuality. See C.1.5.

who — the same as the second use of ‘that’ but used for a person

is of — The common preposition “*of*” is used as a shorthand for “that *is of*.” For any sentential form that takes the general form of ‘<placeholder 1> *has* <placeholder 2>’ there is an implicit reversed form of ‘<placeholder 2> *is of* <placeholder 1>’ that has the same meaning.

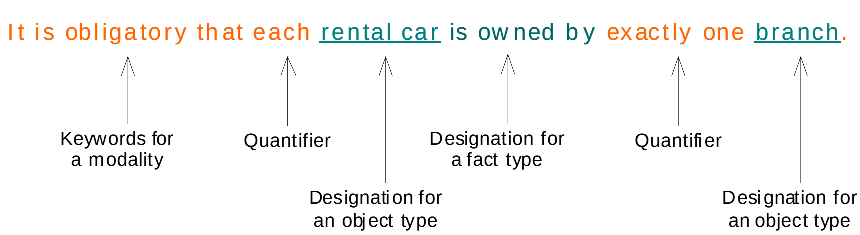
what — used to introduce a variable in a projection as well as indicate that a projection is being formulated to be considered by a question or answer nominalization. See C.1.5 below.

C.1.3 Examples

It is obligatory that each rental car *is owned by* exactly one branch.

Issue 9958 Replace form of expression with fact type form

The example above includes three key words or phrases, two designations for noun concepts and one for a fact type (from a fact type form ~~form of expression~~), as illustrated below.



Below are two statements of a single rule:

1. A rental must have at most three additional drivers.
2. It is obligatory that each rental has at most three additional drivers.

Using the font styles of SBVR Structured English, these rule statements are:

1. A rental must *have* at most three additional drivers.
2. It is obligatory that each rental *has* at most three additional drivers.

A semantic formulation of the rule can be seen in the introduction to "Logical Formulation of Semantics Vocabulary" on page 57.

The characteristic 'driver *is of age*' has the following definition: "the age of the driver is at least the EU-Rent Minimum Driving Age." Below is the definition using the SBVR Structured English styles.

Definition: the age *of* the driver *is at least* the EU-Rent Minimum Driving Age

A semantic formulation of the definition can be seen in the introduction to "Logical Formulation of Semantics Vocabulary" on page 57.

C.1.4 Qualifying Signifiers by Vocabulary and/or ~~Symbol Context~~ Subject Field

Issue 9941 Replace text

Some signifiers are used to mean different things in different vocabularies or in different contexts. In SBVR structured English a signifier can be followed by parentheses enclosing the name of a vocabulary and/or a ~~term for a symbol context~~ subject field. If both are present, they are separated by a comma. Qualifications are shown in the example rules below.

Necessity: Each customer (car rental responsibility) *is* a corporate renter or *is* an individual customer.

The signifier "customer" is used in two ways in the EU-Rent English Vocabulary. So the first rule above uses "customer" for its meaning in the ~~symbol context~~ subject field 'car rental responsibility'.

If the same rule is stated in a place where the EU-Rent English Vocabulary is not understood to be in use, the rule would be stated as follows in order to fully qualify its terms:

Necessity: Each customer (EU-Rent English Vocabulary, car rental responsibility) *is* a corporate renter (EU-Rent English Vocabulary) or *is* an individual customer (EU-Rent English Vocabulary).

C.1.5 Objectification and Nominalization

The keyword 'that' can introduce a proposition being objectified or nominalized. The following examples use the fact types 'car *is assigned to* rental, 'car assignment *involves* car', 'car assignment *is to* rental', 'rental *has* pick-up date', and 'rental *is guaranteed by* credit card'.

The first example is objectification. It states that a car assignment is an actuality denoted by the proposition that a given car is assigned to a given rental. Note that only the third use of 'that' in the example below introduces an objectification. The others introduce restrictions

Necessity: A car assignment that *involves* a car and that *is to* a rental *is* an actuality that the car *is assigned to* the rental.

An objectification uses a propositional expression to identify a state of affairs or event. States and events can then be related to times and durations or be involved in any number of fact types that concern states or events. Consider the following examples of fact types.

state of affairs *occurs before* **point in time**

state of affairs$_1$ *occurs before* **state of affairs**$_2$ *occurs*

The following rule uses the first fact type above:

A car assignment that *is to* a rental must *occur before* the pick-up date *of* the rental.

SBVR Structured English supports objectification using a convenient mechanism that is based on the word "*occurs*" being in the designation of a fact type after a placeholder. An implicit form of a fact type can be used that objectifies a propositional expression in the position of the placeholder and leaves out the word "*occurs*." In other words, the rule above can be stated like this:

A car must *be assigned to* a rental *before* the pick-up date *of* the rental.

Using these implicit forms allows objectification to occur implicitly without defining corresponding noun concepts for each fact type whose instances might be objectified. For example, using the second fact type listed above I can form the following rule even though no noun concept is defined for the fact type 'rental *is guaranteed by* credit card'.

A rental must *be guaranteed by* a credit card *before* a car *is assigned to* the rental.

The next example is a proposition nominalization. It uses the additional fact types 'report *specifies* fact' and 'rental *has* rental report'. The keyword 'that' nominalizes a fact to be specified.

Necessity: If a car *is assigned to* a rental then the rental report *of the* rental must *specify* that the car *is assigned to* the rental.

The next example is an answer nominalization. The keyword 'what' is used to put variables in a projection.

Necessity: The rental report *of* each rental must *specify* what car *is assigned to* the rental.

An expression of a statement can include the keyword 'what' multiple times, putting more variables in the projection (for example, "what car *is assigned to* what rental"). A question nominalization is formed in the same way as an answer nominalization, but nominalizes the question itself rather than an answer to it.

C.1.6 Intensional Roles

Issue 9958 Replace forms of expression with fact type forms
Issue 9731 Revise text
Issue 9712 Revise text

Some fact types about time and change have what can be called intensional roles. In English, most verbs are about their expressed subjects and objects, but in some cases, a verb involves the meaning of the expression of the subject or object. The verb takes its argument by name rather than by value. Fact types for such verbs are often about time and change.

The SBVR Structured English uses a special syntactic clue to identify placeholders for intensional roles in fact type forms ~~forms of expression~~. Normally, a placeholder is shown using a designation for a concept that generalizes its role, but for an intensional role that concept is a concept type and is shown in square brackets after designation for a noun concept that corresponds with syntactic usage of the verb. Some examples of such fact types are listed below.

thing [individual concept] ***is changed***

Definition: the extension of the individual concept is different at one point in time from what it is at a subsequent point in time

Example: “If the scheduled pick-up time of a rental is changed ….”

$thing_1$ ***becomes*** **$thing_2$ [noun concept]**

Definition: the thing is an instance of the noun concept, but having just previously not been an instance of the noun concept

Example: “If a driver of a rental becomes a barred driver before the actual drop-off date of the rental ….”

$quantity_1$ [individual concept] ***increases by*** **$quantity_2$**

Definition: the individual concept refers to a quantity at some point in time and to a different quantity at a later point in time which is greater than the first quantity by the $quantity_2$

Example: “If the odometer reading of a rental car increases by 10,000 miles during a rental ….”

~~Use of such fact types often involves the special semantic formulations noun concept formulation and fact type formulation, explained and exemplified in Part II.~~ Use of such fact types often involves a ~~fact type formulation~~ noun concept nominalization, explained and exemplified in Clause 9. Also, see examples in Annex E.

C.2 Describing a Vocabulary

A vocabulary is described in a document subclause having glossary-like entries for concepts having representations in the vocabulary. Those entries are explained in the next subclause. The introduction to a vocabulary description includes the vocabulary’s name and can further include any of the several kinds of details shown in the skeleton below.

<Vocabulary Name>

Description:

Source:

Speech Community:

Language:

Included Vocabulary:

Note:

C.2.1 The Vocabulary Name

The vocabulary name appears in the ‘Name’ Font.

C.2.2 Description

The ‘Description’ caption is used to introduce the scope and purpose of the vocabulary.

C.2.3 Source

The 'Source' caption is used if the vocabulary being described is based on a formally-defined work. For example, if the vocabulary being described is based on a glossary or other document developed independently of the formalisms of SBVR, then that glossary or other document is shown as the source.

C.2.4 Speech Community

The 'Speech Community' caption is used to name the speech community that controls and is responsible for the vocabulary.

C.2.5 Language

The 'Language' caption is used to name the language that is the basis of the vocabulary. Language names are from ISO 639-2 (English). By default, English is assumed. Note that the SBVR Structured English is based only on English, so descriptions, definitions, and other details are in English but representations being defined can be in another language.

EU-Rent Vocabulaire Française

Language: French

C.2.6 Included Vocabulary

Issue 9958 Replace forms of expressions with fact type forms

The 'Included Vocabulary' caption is used to indicate that another vocabulary is fully incorporated into the vocabulary being described. All designations and fact type forms ~~forms of expressions~~ of an included vocabulary are part of the vocabulary being described.

C.2.7 Note

The 'Note' caption labels explanatory notes that do not go under the other captions.

C.3 Vocabulary Entries

Each entry is for a single concept, called the entry concept. It starts with a primary representation which is either a designation or a fact type form ~~form of expression~~ for the concept.

Any of several kinds of captioned details can be listed under the primary representation. A skeleton of a vocabulary entry is shown below followed by an explanation of the use of each caption.

Issue 9468: Add new paragraph style (Namespace URI)
Issue 9941 Remove text, replace text

<primary representation>

Definition:

Source:

Dictionary Basis:

General Concept:

Concept Type:

~~Symbol Type:~~

Necessity:

Possibility:

Reference Scheme:

Note:

Example:

Synonym:

Synonymous Form:

See:

~~Qualifier:~~

Subject Field:

Namespace URI:

Issue 9958 Replace form of expression with fact type form
Issue 9257 Delete text / Add text
Issue 9930 Add text
Issue 9948 Replace text

C.3.1 Designation or Fact Type Form

~~C.3.1 Designation or Form of Expression~~

A primary representation of ~~(designation or fact type form form of expression)~~ ~~for any concept type~~ an entry can be a term, a name, or a fact type form. It is shown in its appropriate font style.

The primary representation for a fact type is a fact type form ~~form of expression~~. ~~In the unusual case where the same designation is used for more than one placeholder, a subscript on each placeholder can be given so that references to the roles from a definition or other text within the entry is unambiguous~~. The expression of a placeholder is generally the underlined signifier of a designation used by the placeholder to indicate that expressions substituted for the placeholder are understood to denote instances of the designated concept. ~~The fact type role meant by the placeholder is understood to range over that designated concept~~. A designation used by a placeholder for a fact type role is a designation of an object type that the fact type role ranges over. That object type can be a situational role. Sometimes the designation of the object type has the same signifier as a designation of the fact type role. In the unusual fact type form where multiple placeholders use the same designation, the expression of a placeholder can include a subscript to make the expressions of placeholders distinct within the fact type form. Subscripts also help to correlate placeholders across synonymous forms as shown in the example below.

concept$_1$ *specializes* concept$_2$

Definition: the concept$_1$ incorporates each characteristic incorporated into the concept$_2$ plus at least one differentiator

Synonymous Form: $concept_2$ *generalizes* $concept_1$

Synonymous Form: $concept_1$ *has* more general $concept_2$

Synonymous Form: $concept_2$ *has* $category_1$

The fact type forms in the example above represent one fact type that has two fact type roles. From the primary entry it is seen that each of the fact type roles ranges over the concept 'concept'. From the second synonymous form, it is seen that the second fact type role more specifically ranges over the object type 'more general concept' (which is a situational role). From the third synonymous form, it is seen that the first fact type role more specifically ranges over the object type 'category' (which is also a situational role).

The primary representation, whether a designation or fact type form, is in the vocabulary namespace for the vocabulary. Also, if a fact type form is of the pattern "<placeholder 1> has <placeholder 2>", ~~a designation for the second role is in an attributive namespace~~ the expression of <placeholder 2>, less any subscript, is taken as the signifier of a designation of the second fact type role. That designation is in an attributive namespace for the subject concept represented by the designation used for <placeholder 1>. Having a designation for the second fact type role in an attributive namespace means that the designation is recognized as representing the role when it is used in the context of being attributed to instances of the subject concept. From the example above two designations of fact type roles are found in an attributive namespace having the subject concept 'concept'. These designations have the signifiers "more general concept" and "category." Although these designations have the same signifiers as designations of the object types 'more general concept' and 'category', they are different designations. They are within the attributive namespace and represent different concepts (the fact type roles, not the object types). See examples in clause 8 under 'attributive namespace'. Also, if a fact type form is for a unary characteristic, a designation is in an attributive namespace for the concept represented by the designation used for the fact type form's placeholder.

It is recommended that quantifiers (including articles) and logical operators not be embedded within designations and fact type forms ~~forms of expression~~.

C.3.2 Definition

A definition is shown as an expression that can be logically substituted for the primary representation. It is not a sentence, so it does not end in a period.

A definition can be fully formal, partly formal or informal. It is fully formal if all of it is styled as described above. A partially-formal definition starts with a styled designation for a more general concept but other details depend on external concepts.

Styles of definition are explained separately for different types of concepts.

C.3.2.1 Definition of an Object Type ~~Noun Concept~~

Issue 9721 Replace text
Issue 9941 Replace text
Issue 9948 Change noun concept to object type

A common pattern of definition begins with a designation for a more general concept followed by the keyword 'that' (used in the second sense defined for 'that' in the Other Keywords subclause above) and then an expression of necessary and sufficient characteristics that distinguish a thing of the defined concept from other things of the more general concept. Another less used pattern also leads with a designation for a more general concept but then uses the word '*of*' with another expression as explained in the Other Keywords subclause above.

Two kinds of information are formally expressed by a fully formal definition.

1. A fact that the concept being defined is a category of a particular more general concept
2. A closed projection that defines the concept.

Only the first kind of information is formally expressed by a partially formal definition. A partially formal definition leads with a styled designation that is for a more general concept. That designation is generally followed by the keyword 'that' and then an informal expression of necessary and sufficient characteristics.

The following example shows a partially formal definition. It formally expresses the fact that the concept 'icon' is a category of the concept 'nonverbal designation', but it uses words that are external to the formally available vocabulary.

icon

Definition: nonverbal designation that is a pictorial representation

The next example is fully formal. Its formal interpretation includes that the concept ~~obligation claim~~ 'representation' specializes the concept '~~modal formulation~~ actuality' and also includes a closed projection conveying semantics of the definition.

representation
~~obligation claim~~

Definition: ~~modal formulation actuality that *claims* the modality 'obligation'~~
actuality that a given expression *represents* a given meaning

The next example is not formal at all. It defines the most general concept used by SBVR.

thing

Definition: anything perceivable or conceivable

A definition of an ~~noun concept~~ object type can generally be read as a statement using the following pattern (where "a" represents either "a" or "an"):

A <designation> is a <definition>.

For example: An icon is a ~~symbol~~ nonverbal designation that is a pictorial representation.

Issue 9458 Remove/Replace text
Issue 9586 Revise text, remove bindable target

Another style of formal definition is extensional. It uses disjunction to combine a number of concepts. For example, a ~~bindable target is anything that is a variable or a text.~~ contextualized concept is anything that is a role or a facet.

~~For example, a semantic formulation is anything that is a logical formulation or a projection. For example, a contextualized concept is anything that is a role or a facet.~~

~~**semantic formulation**~~

~~Definition: logical formulation or projection~~

contextualized concept

Definition: role or facet

~~bindable target~~

~~Definition: variable or text~~

Issue 9451 Remove text
Issue 9948 change noun concept to object type

C.3.2.2 Definition of an Individual Concept

~~A definition of an individual concept is just like a definition of a noun concept described above except that it must be a definite description of one single thing.~~

A definition of an individual concept must be a definite description of one single thing. It can start with a definite article (e.g., "the"). It can generally be read as a statement using the following pattern. The leading "The" is optionally used depending on the designation.

~~A definition of an individual concept can generally be read as a statement using the following pattern. The leading "The" is optionally used depending on the designation.~~

[The] <designation> is ~~the~~ <definition>.

It is often the case that an individual concept has no definition because it is widely understood. In such a case the 'General Concept' caption can be used to state the type of the named thing. Here is an example.

Switzerland

General Concept: country

Issue 9958 Replace form of expression with fact type form

C.3.2.2 Definition of a Fact Type

A definition given for a fact type is an expression that can be substituted for a simple statement expressed using a fact type form ~~form of expression~~ of the fact type.

The definition must refer to the placeholders in the fact type form ~~form of expression~~. This is done in order to relate the definition to the things that play a role in instances of the fact type. Whether or not the definition is formal, each reference to a placeholder appears in the 'term' font and is preceded by the definite article, "the."

Here is an informal example followed by a fully-formal one.

statement *expresses* proposition

Definition: the proposition is what is meant by the statement

sequence *is of* general concept

Definition: each thing that *is included in* the sequence *is* an *instance of* the general concept

The second definition above is formal such that it translates to a closed projection.

A definition of a fact type can generally be read using the pattern below, which is shown for a binary fact type but works for fact types of any arity ("a" represents either "a" or "an").

> A fact that a given <placeholder 1> <fact type designation> a given <placeholder 2> is a fact that <definition>.

For example: A fact that a given statement expresses a given proposition is a fact that the proposition is what is meant by the statement.

Similarly, the equivalence understood from a definition of a fact type can generally be read using the following pattern:

> A <placeholder 1> <fact type designation> a <placeholder 2> if and only if <definition>.

For example: A statement expresses a proposition if and only if the proposition is what is meant by the statement.

C.3.3 Source

The 'Source' caption is used to indicate a source vocabulary or document for a concept.

The source's designation for the concept is given in square brackets and quoted after the name of the source. It might or might not match the entry's primary representation. If the source has a name for the concept itself, the name is given in square brackets unquoted. The designation from the source is quoted if it is a term for the concept.

thing

Source: ISO 1087-1 (English) (3.1.1) ['object']

individual concept

Source: ISO 1087-1 (English) (3.2.2) ['individual concept']

The keywords "based on" indicate the definition of the concept is largely derived from the given source but had some modification, as in the following example.

language

Definition: system of arbitrary signals (such as voice sounds or written symbols) and rules for combining them as used by a nation, people or other distinct community

Source: based on AH

C.3.4 Dictionary Basis

This caption labels a definition from a common dictionary that supports the use of the primary representation. The entry source reference (written in the 'Source' style described above) is supplied at the end of the quoted definition. A dictionary basis should not be interpreted as an adopted definition.

C.3.5 General Concept

The 'General Concept' caption can be used to indicate a concept that generalizes the entry concept. This is not needed if there is a definition that starts with the general concept, but it is helpful in cases where a definition is not provided, such as is often the case for individual concepts (named things) or concepts taken from a source. Here are two examples.

Switzerland

General Concept: country

individual concept

Source: ISO 1087-1 (English) (3.2.2) ['individual concept']

General Concept: concept

C.3.6 Concept Type

Issue 9958 Replace form of expression with fact type form
Issue 9258 Replace text
Issue 9948 change noun concept to object type, add text, remove text
Issue 9721 revise text

The 'Concept Type' caption is used to specify a type of the entry concept. This is typically not used if the concept has no particular type other than what is obvious from the primary representation.

- A name is implicitly for an individual concept.
- Any term is implicitly for an ~~noun concept~~ object type.
- A fact type form ~~form of expression~~ is implicitly for a fact type.
- For a fact type form ~~form of expression~~, one placeholder implies a unary fact type and two placeholders imply a binary fact type. E.g., 'variable *has* type' is implicitly for a binary fact type.
- Where a definition formally gives a more general concept, the concept being defined specializes that more general concept.

If more than one concept type is mentioned, then they are separated by commas. Order is insignificant.

The concept type 'role' is commonly used where the primary entry is a term. The example below shows that the concept '~~negand~~ logical operand' is a role that is played by a logical formulation. Since the entry concept of a term is implicitly an object type, the additional indication that it is a role implies that it is, by definition, a situational role.

logical operand

~~**negand operand**~~

Concept Type: role

Definition: logical formulation upon which a given logical operation operates

~~Note that a definition of a role of a fact type can use "a given" to clearly mark any opposing roles in the definition, but sometimes an indefinite article is used without the word "given."~~

Any ~~noun concept~~ object type that specializes the concept 'concept' can be given as a concept type. The concept ~~'obligation claim'~~ 'obligation formulation' is a logical formulation kind, which is defined below.

logical formulation kind

Definition: concept that *specializes* the concept 'logical formulation' and that classifies a logical formulation based on the presence or absence of a main logical operation or quantification

Issue 9721 Revise text

obligation formulation

~~claim formulation~~

~~Definition:~~ ~~modal formulation that *claims* the modality 'obligation'~~

Concept Type: logical formulation kind

Issue 9941 Remove text

~~C.3.7 Symbol Type~~

~~The 'Symbol Type' caption is used to mention a category of symbol being defined. This is typically not used if the symbol has no particular type other than what is obvious from the font of the symbol being presented.~~

~~This caption is not generally used to note that a symbol is preferred. Rather, a preferred symbol designation is known because it is listed with a definition or a source. A symbol that is not preferred is shown only with a 'Synonym' or 'Synonymous Form' caption that introduces the preferred symbol designation for the concept.~~

C.3.8 Necessity and Possibility

A 'Necessity' or 'Possibility' is usually supplemental to a definition. A 'Necessity' caption is used to state something that is necessarily true. A 'Possibility' caption explains that something is a possibility that is not prevented by definition. See the vocabulary entries in Clauses 8 to 12 ~~Part II~~ for 'structural business rule statement' and 'unrestricted business rule possibility statement' (respectively) for more details.

The key phrase "it is necessary that" can be omitted from a statement of a structural rule captioned "Necessity" because it is implied by the caption. Here are examples -- two necessity claims and one possibility claim.

~~implication *has* antecedent~~

~~Definition:~~ ~~relates the implication to its conditional operand~~

~~Necessity:~~ ~~That a given implication *has* a given antecedent *is a* structural detail of the implication.~~

~~Necessity:~~ ~~Each implication *has* exactly one antecedent.~~

~~vocabulary *is mapped to* target package~~

~~Possibility:~~ ~~It is possible that a vocabulary *is mapped to* more than one target package.~~

representation

Necessity: Each representation *has* exactly one expression.

Necessity: Each representation *represents* exactly one meaning.

vocabulary namespace *maps to* package

Possibility: A vocabulary namespace *maps to* more than one package.

Definitions express characteristics that are necessary and sufficient to distinguish things denoted by a concept. Sometimes there are necessities beyond what is sufficient. The 'Necessity' caption is used to state such necessities.

C.3.9 Reference Scheme

The 'Reference Scheme' caption is used to state how things denoted by the term can be distinguished from each other based on one or more facts about the things. A reference scheme is expressed by referring to at least one role of a binary fact type and indicating whether a reference involves a single instance of the role or whether it involves the extension of related instances.

Issue 9462 Revise text
Issue 9948 Add text, replace text

An article ('a', 'an', or 'the') indicates a simple use of a role in which a single instance is used in a reference. The definite article 'the' is only appropriate where there can be at most one instance of the role. The words 'the set of ~~each~~' indicate that the extension is used. The word 'and' is used to connect the expressions of use of multiple roles by a reference scheme.

The following examples of reference schemes are taken from the SBVR Vocabularies. The first one below uses a single value of ~~a role~~ the 'closed logical formulation' role of the fact type 'closed logical formulation *means* proposition' meaning that a proposition can be identified by any closed logical formulation whose meaning is the proposition. The second uses two fact type roles. It uses a definite article because each role binding has exactly one bindable target and is for exactly one fact type role.

proposition

Reference Scheme: a closed logical formulation that *means* the proposition

role binding

Reference Scheme: the bindable target that *is referenced by* the role binding and the fact type role that *has* the role binding

---

Issue 9452 Revise text

The reference scheme for the concept of reference scheme itself uses ~~two~~ three roles extensionally.

Issue 9452 Add text
Issue 9462 Remove text/Revise text

---

reference scheme

Reference Scheme: the set of ~~each~~ fact type roles that *are simply used by* the reference scheme and the set of ~~each~~ fact type roles that *are extensionally used by* the reference scheme and the set of characteristics that *are used by* the reference scheme

C.3.10 Note

A 'Note' caption is used to label explanatory notes that do not fit within the other captions.

C.3.11 Example

The 'Example' caption labels examples involving the entry concept.

C.3.12 Synonym

Issue 9958 Replace form of expression with fact type form
Issue 9930 Add text

A synonym is another designation that can be substituted for the primary representation. It is a designation for the same concept. If the primary representation is a fact type form ~~form of expression~~, then the 'Synonymous Form' caption is used rather than the 'Synonym' caption.

The examples below show two synonyms for one concept having one definition. The preferred ~~symbol~~ designation is given as the primary representation.

implication

Definition: logical formulation that applies the logical "(MATERIALLY) IMPLIES" operation (→) to an antecedent and a consequent

Synonym: material implication

The meaning of two designations being synonyms is that they represent the same concept.

Each synonym is in the vocabulary namespace of the vocabulary.

C.3.13 Synonymous Form

Issue 9958 Replace form of expression with fact type form
Issue 9930 Add text
Issue 9948 Replace text

A synonymous form is a fact type form ~~form of expression~~ for the same fact type. The order of placeholders for fact type roles can be different.

A synonymous form can appear elsewhere as its own entry. However, this is not typically done if the synonymous form is simply a passive form of the primary representation. The following example shows a synonymous form that reverses the order of fact type roles. Because the synonymous form is simply a passive form of the primary representation, it does not appear as a separate entry.

statement *expresses* proposition

Definition: the proposition is what is meant by the statement

Synonymous Form: proposition *is expressed by* statement

~~A synonymous form does not necessarily use the same designations for placeholders as are used in the primary representation. A placeholder might or might not use a designation for a role. When this happens, in order to match the placeholder of the synonymous form to the corresponding placeholder in the primary representation, the placeholder in the synonymous form is followed by the placeholder's designation in the primary representation in square brackets.~~

A synonymous form does not necessarily use the same designations for all placeholders as are used in the primary designation. One placeholder can use a different designation. The ones using the same designation as placeholders of the primary form represent the corresponding fact type roles, and the one placeholder that does not match represents the remaining fact type role. The example below shows two entries, both for the same concept. One is expressed in terms of a role (instance) and the other is not.

Issue 9950 Remove text

---

concept *corresponds to* thing

Definition: the thing is in the extension of the concept

Synonymous Form: concept *has* instance [~~thing~~]

concept *has* instance

Synonymous Form: concept *corresponds to* thing [~~instance~~]

If the same term is used for multiple placeholders, then subscripts can be used to distinguish them.

thing_1 *is* thing_2

Synonymous Form: thing_1 *equals* thing_2

The meaning of two fact type forms ~~forms of expression~~ being synonymous is that the two represent the same fact type.

Each synonymous form is in the vocabulary namespace of the vocabulary. Designations are in attributive namespaces as explained for primary entries in C.3.1.

C.3.14 See

Where the primary representation is not a preferred representation for the entry concept, the "See:" caption introduces the preferred representation. No definition is given in this case.

Issue 9941 Replace text

C.3.15 Subject Field

Where a signifier is not unique in a vocabulary, there is a need for qualification by a subject field. The subject field of a designation is given using the “Subject Field” caption, as shown in the example below.

customer

Subject Field: Car Rental Responsibility

See: renter

customer

Subject Field: Vehicle Sales

Definition: person who purchases a rental car from EU-Rent at the end of its rental life

C.3.16 Namespace URI

Issue 9468: Add text

If the primary entry is for a namespace, the ‘Namespace URI’ caption is used to indicate a URI of the namespace. If the primary entry is for a vocabulary, the ‘Namespace URI’ caption is used to indicate a URI of a vocabulary namespace for the vocabulary. Here is an example:

Meaning and Representation Vocabulary

General Concept: vocabulary

Namespace URI: http://www.omg.org/spec/SBVR/1.0/MeaningAndRepresentation

C.4 Specifying a Rule Set

A rule set is specified in a document subclause having several individual entries for guidance. Those entries are explained in the next subclause. The introduction to a rule set includes the rule set’s name and can further include any of the several kinds of details shown in the skeleton below.

<Rule set name>

Description:

Vocabulary:

Note:

Source:

C.4.1 The Rule Set Name

The rule set name appears in the ‘name’ font.

C.4.2 Description

The ‘Description’ caption is used to describe the scope and purpose of the rules.

C.4.3 Vocabulary

The ‘Vocabulary’ caption is used to identify what vocabulary (defined in terms of SBVR) is used by statements in the rule set.

C.4.4 Source

The ‘Source’ caption is used if the rule set is based on a separately-defined work. It labels a reference to such a work, such as a legal statute.

C.4.5 Note

The ‘Note’ caption is used to label explanatory notes that do not fit within the other captions.

C.5 Guidance Entries

Issue 9475/9945 Add/change text

Each entry in a rule set is an element of guidance -- expressed as one of the following:

- An operative business rule statement
- A structural business rule statement
- ~~An admonition~~ A statement of advice of permission
- ~~An affirmation~~ A statement of advice of possibility

Business rules include only those rules under business jurisdiction. Entries can also be made for structural rules that are not under business jurisdiction. Each entry includes the statement itself and optionally includes other information labeled by the captions shown below.

Issue 9459 Change “Synonymous Form” to “Synonymous Statement”
Issue 9608 Revise title

<~~Rule Statement or Clarification~~ Guidance Statement>

Name:
Guidance Type:
Description:
Source:
Synonymous Statement:
Note:
Example:
Enforcement Level:

Use of each of the above captions is explained below.

Issue 9608 Revise title and text

C.5.1 Guidance Statement

~~C.5.1 Rule Statement or Clarification Statement~~

A ~~rule statement or clairification statement~~ guidance statement can be expressed formally or informally. A statement that is formal uses only formally styled text — all necessary vocabulary is available (by definition or adoption) such that no external concepts are required. Such a statement can be formulated as a logical formulation.

C.5.2 Name

The 'Name' caption is used to specify a name for the ~~rule or clarification~~ element of guidance. The name is then part of the formal vocabulary.

C.5.3 Guidance Type

Issue 9475/9945 Add/change text

The 'Guidance Type' caption is used to indicate the kind of element of guidance -- i.e., one of the following:

- operative business rule
- structural business rule
- ~~admonition~~ advice of permission
- ~~affirmation~~ advice of possibility
- advice of optionality
- advice of contingency

C.5.4 Description

The 'Description' caption is used to capture the expression of the element of guidance informally (as supplied by a business user).

C.5.5 Source

The 'Source' caption is used if the guidance is from a separate source. It labels a reference to that source.

Issue 9459 Change "Synonymous Form" to "Synonymous Statement"

C.5.6 Synonymous Statement

The ~~‘Synonymous Form’~~ ‘Synonymous Statement’ caption is used to state additional, equivalent statements of the guidance. For example, a given rule can be expressed in a ‘prohibitive’ form and also in an ‘obligatory’ form. As for the primary statement of the guidance, these additional statements can be formal or informal.

C.5.7 Note

The ‘Note’ caption is used to label explanatory notes that do not fit within the other captions.

C.5.8 Example

The ‘Example’ caption labels examples of application of the element of guidance.

C.5.9 Enforcement Level

The ‘Enforcement Level’ caption labels the enforcement level that applies to an operative business rule (only).

Annex D
(informative)

SBVR Structured English Patterns

Issue 9941 Replace text
Issue 9952 Revise text

This annex contains material compiled to aid the interpretation of ‘SBVR in SBVR Structured English’ vocabulary entries, as documented in Annex C and applied in the text and diagram forms of Part II and Annex E. This ‘language patterns’ material falls into two main categories:

- reading SBVR Vocabulary ~~symbols~~ designations
- reading fact types embedded in the definition text of SBVR Vocabulary ~~symbols~~ designations.

Issue 9450 Add text

A third subclause contains the brief discussion of a useful pattern that, while not often applied in the text of Part II, is illustrated in Annex E (and, in particular, in the “10 Introductory Examples” given there and in the RuleSpeak and ORM Annexes). This discussion introduces the use of a ‘short form’ fact type that can be used to simplify the formulation and representation of vocabularies and sets of elements of guidance.

When there is an associated way to depict the construct in a graphic notation, a cross-reference is provided, when applicable, to the ‘Use of UML Notation in a Business Context to Represent SBVR-based Vocabularies’ (Annex H) -- referred to here as the ‘UML style’ -- and to the ‘Concept Diagram Graphic Notation (Annex G)’ -- referred to here as the ‘CDG style’.

Issue 9952 Revise text

D.1 Reading SBVR Vocabulary ~~Symbols~~ Designations

This subclause presents the interpretation given to three kinds of designations:

- Terms
- Names
- ~~Designations for Fact Types~~ Fact symbols

D.1.1 Primary Term for a General Concept

When I see a vocabulary entry as shown in Figure D.1, I know to vocalize it as:

‘community’ is a term for a general concept. And it is the ‘primary’ term used for the concept.

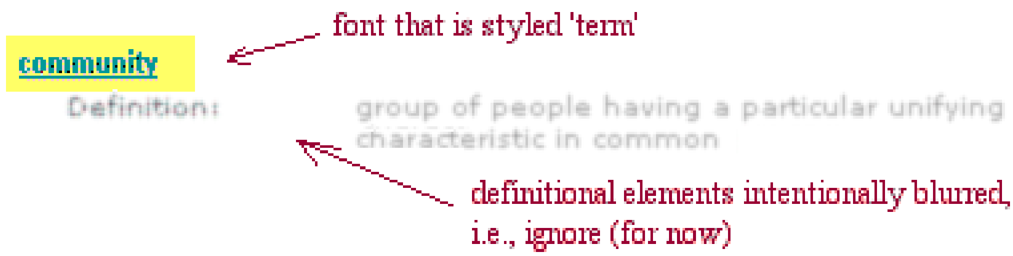


Figure D.1 - Recognizing an entry that is the primary term for a general concept.

For how to depict this in graphics, see H.1 (UML style) and G.1 (CDG style).

Commentary:

This is a typical ~~*symbol*~~ *designation* kind of entry presented as a ‘term’ -- the primary term for a general concept. For this kind of entry, draw a labeled box.

It is possible to have additional terms for a given general concept (i.e., terms that are synonyms). Even when documented in the text form (using the ‘Synonym’ caption), the non-primary terms of a concept are not typically reflected on the graphic. When it is considered useful to make explicit entries for the non-primary terms in a presentation of the vocabulary, the non-primary terms can appear using the ‘See’ caption to refer back to the concept’s primary term.

Issue 9952 Revise text

D.1.2 Primary Name for an Individual Concept

When I see a vocabulary entry as shown in Figure D.2, I know to vocalize it as:

‘Real-world numerical correspondence’ is a term that is a name for an individual concept. And it is the primary name used for the concept.

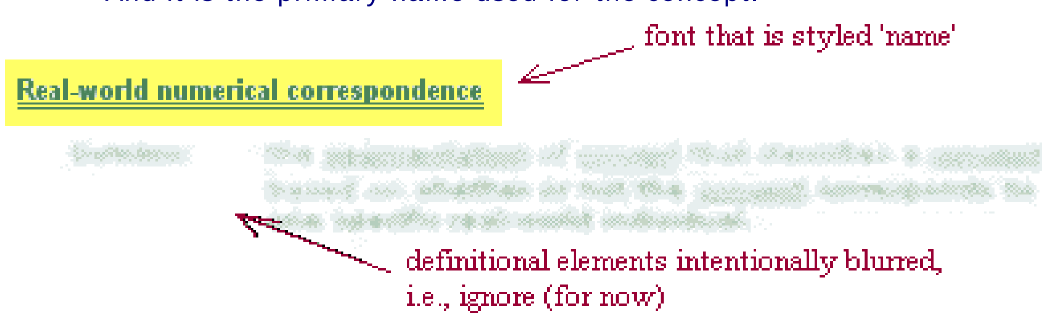


Figure D.2 - Recognizing an entry that is the primary name for an individual concept

For how to depict this in graphics, see H.2 (UML style). There is no specified way to depict this in the CDG graphic notation.

Commentary:

This is a typical ~~*symbol*~~ *designation* kind of entry presented as a ‘name’ -- the primary name for an individual concept. For this kind of entry, draw a labeled box, with the ‘name’ underlined.

It is possible to have additional names for a given individual concept (i.e., names that are synonyms). Even when documented in the text form (using the ‘Synonym’ caption), the non-primary terms of a concept are not typically reflected on the graphic. When it is considered useful to make explicit entries for the non-primary names in a presentation of the vocabulary, the non-primary names can appear using the ‘See’ caption to refer back to the concept’s primary name.

D.1.3 Primary Reading (‘Sentential Form’) for a Fact Type

D.1.3.1 Primary Reading (‘Sentential Form’) for a Fact Type -- Binary Fact Type

When I see a vocabulary entry as shown in Figure D.3, I know to vocalize it as:

> There is a fact type relating these two concepts and it uses the designation ‘shares understanding of’ when the concept terms are in this order. Optionally, alternative readings can be provided using the ‘Synonymous Form’ caption (as illustrated at the bottom of Figure D.3).

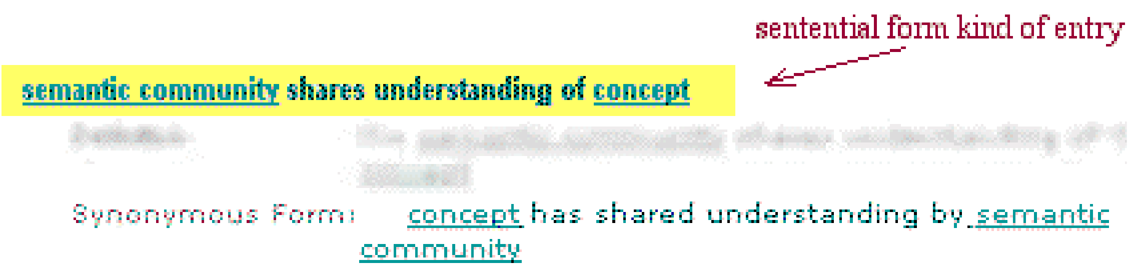


Figure D.3- Recognizing an entry that is the primary reading for a binary fact type

For how to depict this in graphics, see H.3.1 (UML style) and G.3.1 (CDG style). There is a special case of depicting a binary fact type that uses ‘has’ in the UML style. For how to depict this in graphics, see H.3.2 (UML style). There is no special way to depict this in the CDG graphic notation.

Commentary:

This is a typical *sentential form* kind of entry for a fact type -- in this case, a binary fact type. For this kind of entry, draw a labeled line between the boxes for the ~~symbols~~ designations of the participating concepts. The reading is clockwise (when the tool does not provide a graphic symbol for indicating the directionality of the reading).

It is possible to have additional readings for a given fact type (i.e., readings that are ‘synonymous forms’ of the fact type). Additional readings are optional in both the graphic and text forms. When defined in the text form, the ‘Synonymous Form’ caption is used. Even when provided in the text, more than one reading is not typically reflected on the graphic. However, having inverse readings on an association would be an extension to UML. (This can be handled legally by defining a ‘UML profile’, which allows additional information and custom graphics in a model.)

An alternative graphic style is to apply the n-ary graphic style (described below) for ***all*** fact types, including binary.

D.1.3.2 Primary Reading (‘Sentential Form’) for a Fact Type -- N-ary Fact Type

When I see a vocabulary entry as shown in Figure D.4, I know to vocalize it as:

There is a ternary fact type relating these three concepts, using 'is replaced by ... in' when the fact type uses these terms for the concepts in this sequence.

Figure D.4 - Recognizing an entry that is the primary reading for an n-ary fact type

For how to depict this in graphics, see H.3.3 (UML style) and G.3.2 (CDG style).

Commentary:

This is a *sentential form* kind of entry for a fact type -- in this case, an n-ary fact type. For this kind of entry, there are two diagrams forms. The first diagram is the box-in-box style as defined in Annex G.3.2. The second diagram (UML-style) uses a box, given a stereotype that names the category of fact type, and a label that reflects the primary reading for the fact type. The concept terms are placed in [ ].

Note-1 -- The label in the UML form does not use the UML association 'name'; the UML association 'name' is reserved for use as a 'real' name.

Note-2 -- While suggestions have been given for depicting multiple readings on a diagram, showing additional readings for n-ary fact types is not currently part of the scope of this documentation.

D.1.3.3 Primary Reading ('Sentential Form') for a Fact Type -- Unary Fact Type

When I see a vocabulary entry as shown in Figure D.5, I know to vocalize it as:

There is a unary fact type for this concept, with a designation of 'is damaged'.

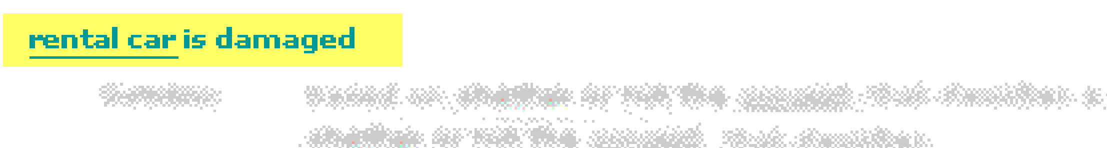


Figure D.5 - Recognizing an entry that is the primary reading for a unary fact type

For how to depict this in graphics, see H.3.4 (UML style) and G.3.3 (CDG style).

Commentary:

This is a *sentential form* kind of entry for a fact type -- in this case, a unary fact type. For this kind of entry, the two graphic notations use different forms. The first diagram above shows the box-in-box style as defined in Annex G.3.3. For the UML-style, three alternatives are offered:

1. List the designation inside the box ('attribute' style).
2. Draw in the same style as for an n-ary fact type (above).
3. Draw using the association 'diamond'.

NOTE: The notation for unary fact type would be an extension to UML, handled legally by defining a 'UML profile'.

D.1.3.4 Two Vocabulary Entries (Sentential Form and Term) for a Concept

When I see a pair of vocabulary entries as shown in Figure D.6, I know to vocalize this case as:

> These two entries are for coextensive concepts. I understand that, even though these are two entries in the vocabulary, they have the same instances.

rented car *is recovered from* non-EU-Rent site *to* branch

car recovery

Definition: actuality that a given rented car *is recovered from* a given non-EU-Rent site *to* a given branch

Figure D.6- Recognizing a pair of entries (sentential form and term) for a concept

For how to depict this in graphics, see H.8 (UML style) and G.3.4 (CDG style).

D.2 Reading Embedded Fact Types

There are also fact types that are defined when the SBVR Structured English language is used to compose the definition of a vocabulary entry. The material in this subclause documents the most common patterns used in writing entry definitions using the elements of style defined in Annex C.

The following six patterns have been documented.

- categorization fact type
- is-role-of fact type
- partitive fact type
- instance to general concept ('predefined extension')
- categorization type
- categorization scheme

D.2.1 Categorization Fact Type

When I see this:

semantic community

Definition: community whose unifying characteristic is a shared understanding (perception) of the things that they have to deal with

I know this is shorthand for:

semantic community

Concept Type: category

Definition: community whose unifying characteristic is a shared understanding (perception) of the things that they have to deal with

I know to vocalize it as:

> The concept ‘semantic community’ is a ‘category’ of the more general concept ‘community’. Furthermore, I know that what distinguishes this particular kind of community from the general case is that it is ... <distinctions brought out in the rest of the definition>

For how to depict this in graphics, see H.5 (UML style) and G.2.1 (CDG style).

D.2.2 Is-role-of Fact Type

When I see this:

renter

Concept Type: role

Definition: driver who ,,, l

I know to vocalize it as:

> The concept ‘renter’ is a role that can be played by a driver, specifically one who ... <distinctions brought out in the rest of the definition>

For how to depict this in graphics, see H.4 (UML style) and G.4 (CDG style). The CDG style does not distinguish the various ways to depict roles as in the UML style (see treatment in H.4.1, H.4.2, and H.4.3).

Issue 9941 Replace text

D.2.3 Partitive Fact Type

When I see this:

body of shared meanings$_1$ *contains* body of shared meanings$_2$

Concept Type: partitive fact type

Definition: the body of shared meanings *includes* everything in another body of shared meanings

body of shared meanings *includes* body of shared concepts

Concept Type: partitive fact type

I know to vocalize it as:

> A body of shared meanings contains other bodies of shared meanings.
>
> A body of shared meanings includes bodies of shared concepts.

For how to depict this in graphics, see H.7 (UML style). There is no specified way to depict this in the CDG graphic notation.

vocabulary$_1$ *incorporates* vocabulary$_2$

Concept Type: partitive fact type

Definition: the vocabulary$_1$ *includes* each symbol that *is included in* the vocabulary$_2$

Note: When more than one vocabulary is included, a hierarchy of inclusion can provide priority for selection of definitions.

vocabulary$_2$ *is incorporated into* vocabulary$_1$

vocabulary *includes* symbol

Concept Type: partitive fact type

symbol *is included in* vocabulary

I know to vocalize it as:

A vocabulary incorporates (another) vocabulary.

A vocabulary includes symbols.

For how to depict this in graphics, see H.7 (UML style). There is no specified way to depict this in the CDG graphic notation.

Issue 9952 Revise text

D.2.4 Instance to General Concept Fact Type (‘Predefined Extension’ Entry)

When I see this:

Canada

General Concept: country

I know to vocalize it as:

Canada is an instance of the concept ‘country’

(or, ‘Canada’ is a designation of an individual country)

For how to depict this in graphics, see the discussion of ‘Primary ~~Term (Name)~~ Name for an Individual Concept’ above.

Typically, this kind of entry is simply ‘indicated’ (or perhaps ‘adopted’), with no definition. However, when a definition is written, its styling can specify the general concept, in which case, the ‘General Concept’ caption can be omitted. For example, the entry below defines ‘Car Rental Industry’ to be an instance of ‘semantic community’

Car Rental Industry

Definition: the semantic community that *is* the group of people who work in the business of renting cars

Commentary:

When you find this pattern, draw it in the UML style using UML’s arrow style for ‘instantiation’. The notation has been adapted from standard UML notation to make it more ‘business friendly’ -- e.g., in UML, in instance (‘object’) would be labeled as, Canada: country. Predefined extension instances are not typically depicted in the box-in-box style.

D.2.5 Categorization Type

When I see this:

branch type

Definition: concept that *specializes* the concept 'branch' and that *classifies* a branch based on its hours of operation and car storage capacity

city branch

Concept Type: branch type

Definition: branch that operates in a city

I know to vocalize it as:

> The concept 'branch type' has instances that are (or are in 1:1 correspondence with) certain categories of 'branch' -- depending on the interpretation you take for this pattern.
>
> 'city branch' is a category of 'branch'.
>
> 'city branch' is (or is in 1:1 correspondence with) a 'branch type'.

For how to depict this in graphics, see H.6.2 (UML style). There is no specified way to depict this in the CDG graphic notation.

Commentary:

When you find this pattern -- a 'Definition' caption that begins,

concept that *specializes* the concept 'other-concept' and that *classifies* an other-concept based on…

-- it is a compact, textual way to say multiple things, as follows:

1. that the mentioned *other-concept* has categories for which the *other-concept* is the more general concept, and
2. that the entry being defined is itself a category of concept, one whose instances are the categories of the mentioned more general concept.

Furthermore, the vocabulary entries for the certain category include a 'Concept Type:' caption that mentions the categorization type. For example, the vocabulary entry for 'city branch' mentions 'branch type' as its Concept Type.

(The examples that illustrate how to depict the '1:1 correspondence' interpretation have not yet been developed.)

D.2.6 Categorization Scheme

When I see this:

Branches by Type

Description: segmentation that *is for* branch and *subdivides* branch based on branch type

Necessity: Branches by Type *contains* the categories 'airport branch' and 'city branch' and 'agency'.

agency

Definition: branch that *does* not *have an EU-Rent location* and *has minimal car storage* and *has on-demand operation*

Necessity: agency *is included in* Branches by Type.

airport branch

Definition: branch that *has an EU-Rent location* and *has large car storage* and *has 24-7 operation*

Necessity: airport branch *is included in* Branches by Type.

city branch

Definition: branch that *has an EU-Rent location* and *has moderate car storage* and *has long business hours*

Necessity: city branch *is included in* Branches by Type.

I know to vocalize it as:

> 'Branches by Type' is the name of a categorization scheme (or, in this case, a 'segmentation', which is a restricted case of categorization scheme). This scheme is for the general concept 'branch', presenting the instances of branch as divided into the categories that make up the scheme, according to the stated criteria. Each category's entry indicates being part of the scheme.

For how to depict this in graphics, see H.6.1 (UML style) and G.2.2 (CDG style).

Commentary:

When you find this pattern -- ~~a 'Definition' caption~~ under a 'name' ~~symbol~~ designation with a 'Definition' caption that begins,

the categorization scheme that *is for* the concept 'mentioned-other-concept' and *subdivides* mentioned-other-concept based on...

or

the segmentation that *is for* the concept 'mentioned-other-concept' and *subdivides* mentioned-other-concept based on...

-- it is a compact, textual way to say multiple things, as follows:

1. that the entry being defined is a categorization scheme (or a categorization scheme that is a segmentation), and
2. that the mentioned concept is the concept that is the scheme is *for.*

   Furthermore, each vocabulary entry for one of the categories in the scheme identifies itself as part of the scheme using a 'Necessity' caption. (Note that a category can be part of more than one scheme.)

D.3 Defining a Fact Type for Convenience

Issue 9450 Add text

The development of vocabularies and sets of elements of guidance often calls for trade-offs of redundancy (in the sense of defining a concept both directly and indirectly) against simplification of formulation and representation. Consider, for example, the first of the ten introductory examples presented in Annex E.1.4:

> It is necessary that each rental *has* exactly one requested car group.

This is easy to grasp. Now, consider the full form of this rule if the rule were based solely on a sparse EU-Rent vocabulary. The rule would then be as follows:

> It is necessary that each rental *has* exactly one car group that *is specified in* the car movement that *is included in* the rental.

As this simple example demonstrates, the full form of a rule (or advice) can become quite verbose when several fact types are involved.

The compact form of this rule makes use of the *short form* fact type 'rental *has* requested car group', a redundant concept that has been created for the purpose of simplification of formulation and representation. This fact type specifies its instances as being derived from (equivalent to) the concatenation of other fact types -- the *verbose* form -- as illustrated by the following entry that specifies the concept:

rental *has* requested car group

Necessity: A rental *has* a requested car group if and only if the requested car group *is* the car group that *is specified in* the car movement that *is included in* the rental.

This technique is particularly useful when the *short form* fact type is used in a number of elements of guidance. For another example, from Annex E, the fact type 'rented car *is assigned to* rental' is a basis element for three of the ten introductory examples.

Note, however, the choice to apply this pattern is a matter of practice. Decisions on reuse and redundancy are business decisions made by the semantic community (here, EU-Rent) to help it manage its body of shared meanings and vocabularies.

Annex E
(informative)

EU-Rent Example

Issue 9948 Replace each occurrence of ‘role of the noun concept’ with ‘role that ranges over the noun concept.’

E.1 Introduction

EU-Rent is a (fictitious) car rental company with branches in several countries.

E.1.1 Overview of EU-Rent’s Business Service

EU-Rent rents cars to its customers. Customers may be individuals or companies. Different models of car are offered, organized into groups. All cars in a group are charged at the same rates. A car may be rented by a booking made in advance or by a ‘walk-in’ customer on the day of rental. A rental booking specifies the car group required, the start and end dates/times of the rental and the EU-Rent branch from which the rental is to start. Optionally, the reservation may specify a one-way rental (in which the car is returned to a branch different from the pick-up branch) and may request a specific car model within the required group.

EU-Rent has a loyalty club. Customers who join accumulate points that they can use to pay for rentals

EU-Rent from time to time offers discounts and free upgrades, subject to conditions.

EU-Rent records ‘bad experiences’ with customers (such as unauthorized late return from rental, or damage to car during rental) and may refuse subsequent rental reservations from such customers.

E.1.2 EU-Rent Organization

EU-Rent’s organization is illustrated below:

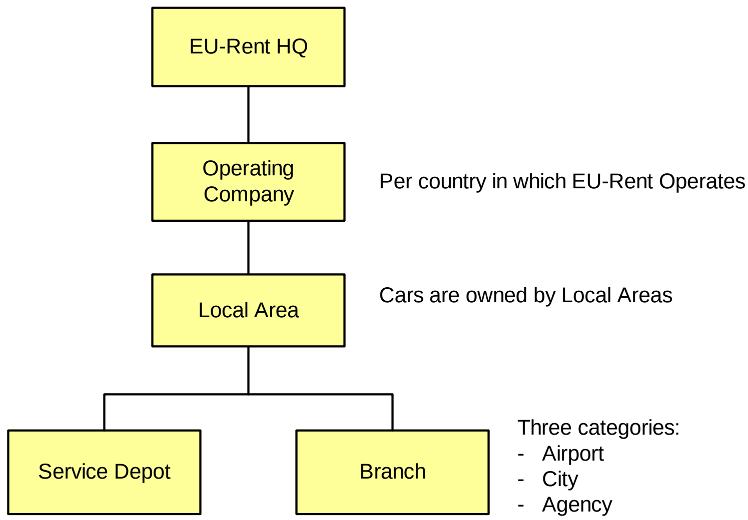


Figure E.1- EU-Rent Organization

EU-Rent's world headquarters sets global policy and owns the world-wide reservations system.
In each country in which it does business EU-Rent has an Operating Company that:

- Adapts global policy to local regulation, custom, and practice
- Selects which car models will be purchased for each car group
- Sets rental tariffs

Within each country EU-Rent manages its business by Local Area, which EU-Rent defines. A Local Area contains a number of Branches for Rental Car pick-up and return, and a number of Service Depots that maintain and repair EU-Rent's cars.

Cars are stored at Branches but owned by Local Area. They are moved between Branches within a Local Area to meet rental demand.

Cars may also be moved between Local Areas within a country; if this happens, car ownership is also transferred. Car movements happen in two ways:

- A direct transfer. This is an action internal to EU-Rent: a EU-Rent employee drives the car.
- A one-way rental between different local areas in the same country: a rental customer drives the car. [Note: if the one-way rental is between different countries, car ownership is not transferred. At a time decided by the branch manager, the car is returned to the country in which it is registered].

Branches are of three types:

- 'Airport': large Branches at major airports and some major rail terminals. They are open 24 hours per day, 7 days per week, have storage capacity for hundreds of cars, and sufficient staff to have specialized roles in the workflow.
- 'City': small Branches. They usually keep extended business hours (e.g., 7.00 a.m. to 8.00 p.m.), have storage space for tens of cars, and small numbers of staff who are interchangeable in the workflow.
- 'Agency': service desks in hotels, travel agents, etc. They have storage space for few cars, and are operated on demand by part-time staff who will typically do the entire workflows for rental and return.

E.1.3 Adopted Vocabularies

To illustrate vocabulary adoption, two car industry glossaries (also fictitious) have been introduced. One is in English; the other in German.

Issue 9955 Replace text

E.1.4 Introductory Examples

Below are some elements of guidance, selected from the main part of the case study to illustrate how guidance is expressed and how fact types represented in the vocabulary support the formulation of guidance.

The intention is to provide an initial impression of what EU-Rent's business rules look like and how they are supported by fact types as expressed in EU-Rent's vocabulary, before going into the full detail of the case study. The examples also appear, with related elements of guidance, in the main body of the case study that follows this introduction.

The entries noted as "supporting fact types" are the fact types on which the guidance is directly based. Those noted as ~~"related factual connections" indicate related parts of the vocabulary which would be of interest to a business person~~ "related facts" indicate facts known from EU-Rent's body of shared meanings that would be of interest to a business person.

Issue 9448 Change formatting
Issue 9449 Revise text
Issue 9948 Replace text
Issue 11291 Revise text

1	Structural business rule	It is necessary that each rental *has* exactly one requested car group.
	Supporting fact type	rental *has* requested car group
2	Operative business rule	It is obligatory that the rental duration *of* each rental *is at most* 90 rental days.
	Supporting fact type	rental *has* rental duration
		duration_1 *is at most* duration_2
	Related facts:	The noun concept 'rental duration' *is* a role that *ranges over* the noun concept 'duration.'
		rental duration *is measured in* rental time unit [aka RTU].
		The individual concept 'rental day' *is* an instance *of* the noun concept 'rental time unit.'
3	Operative business rule	It is obligatory that each driver *of* a rental *is qualified*.
	Supporting fact types:	rental *has* driver
		driver *is qualified*

	Related facts:	~~the noun concept 'qualified driver' is a category of the noun concept 'driver'~~
		The noun concept 'driver' *is* a facet *of* the noun concept 'person.'
4	Operative business rule	~~If the drop-off location *of* a rental *is* not the EU-Rent site *of* the return branch *of* the rental then it is obligatory that the rental *incurs* a location penalty charge.~~
		It is obligatory that the rental *incurs* a location penalty charge if the drop-off location *of* a rental *is* not the EU-Rent site that *is base for* the return branch *of* the rental.
	Supporting fact types:	rental *has* drop-off location
		rental *has* return branch
		~~branch *is located at* EU-Rent site~~
		rental *incurs* location penalty charge
		$thing_1$ *is* $thing_2$
		EU-Rent site *is base for* rental organization unit
	Related facts:	The noun concept 'return branch' *is* a role that *ranges over* the noun concept 'branch.'
		The noun concept 'branch' *is* a category *of* the noun concept 'rental organization unit.'
		The noun concept 'EU-Rent site' *is* a role that *ranges over* the noun concept 'location.'
		The noun concept 'drop-off location' *is* a role that *ranges over* the noun concept 'location.'
5	Operative business rule	It is obligatory that the rental charge *of* a rental *is calculated in* the business currency *of* the rental.
	Supporting fact types:	rental *has* rental charge
		rental charge *is calculated in* business currency
		rental *has* business currency
6	Operative business rule	It is permitted that a rental *is open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.

Supporting fact types:

~~rental has estimated rental charge~~

rental *has* rental charge

estimated rental charge *is provisionally charged to* credit card

renter *has* credit card

~~rental has renter~~

rental *has* driver

rental *is open*

renter *is responsible for* rental

Related facts:

~~'being open' is a characteristic of the concept 'rental'~~

The noun concept 'estimated rental charge' *is* a category *of* the noun concept 'rental charge.'

The noun concept 'renter' *is* a role that *ranges over* the noun concept ~~person~~ 'driver.'

The noun concept 'driver' *is* a facet *of* the noun concept 'person.'

7 Operative business rule

~~It is obligatory that at the actual return date/time of each in-country rental and each international inward rental the local area of the return branch of the rental owns the rented car of the rental.~~

It is obligatory that the local area that *includes* the return branch *of* an in-country rental or international inward rental *owns* the rented car *of* the rental *at* the actual return date/time *of* the rental.

Supporting fact types:

rental *has* actual return date/time

rental *has* return branch

branch *is included in* local area

local area *owns* rental car

state of affairs *occurs at* date/time

rental *has* rented car

Related facts: the noun concept 'rented car' *is* a role that *ranges over* the noun concept 'rental car'

the noun concept 'return branch' *is a* role that *ranges over* the noun concept 'branch'

the noun concept 'in-country rental' *is* a category *of* the noun concept 'rental'

the noun concept 'international inward rental' *is* a category *of* the noun concept 'international rental'

the noun concept 'international rental' *is* a category *of* the noun concept 'one-way rental.'

The noun concept 'one-way rental' *is* a category *of* the noun concept 'rental.'

8 Operative business rule: It is obligatory that *at* the actual pick-up date/time *of* each rental the fuel level *of* the rented car *of* the rental *is* full.

Supporting fact types:

rental *has* actual pick-up date/time

rental *has* rented car

rental car *has* fuel level

state of affairs *occurs at* date/time

Related facts: the noun concept 'rented car' *is* a role that *ranges over* the noun concept 'rental car'

fuel level *is* full or 7/8 or 3/4 or 5/8 or 1/2 or 3/8 or 1/4 or 1/8 or empty

The noun concept 'actual pick-up date/time' *is* a role that *ranges over* the noun concept 'date/time.'

Issue 9475/9945: Add/change text

9 ~~Affirmation~~ Advice of possibility: It is possible that the notification date/time *of* a bad experience that *occurs during* a rental *is after* the actual return date/time *of* the rental.

Supporting fact types:

bad experience *occurs during* rental

bad experience *has* notification date/time

rental *has* actual return date/time

date/time$_1$ *is after* date/time$_2$

Related facts: the noun concept 'notification date/time' *is* a role that *ranges over* the noun concept 'date/time'

the noun concept 'actual return date/time' *is* a role that *ranges over* the noun concept 'date/time'

10	~~Admonition~~ Advice of permission	It is permitted that the drop-off branch *of* a rental *is* not the return branch *of* the rental
	Supporting fact types:	rental *has* drop-off branch
		rental *has* return branch
		thing_1 *is* thing_2

E.2 EU-Rent Examples

The case study is presented in two parts. The first subclause illustrates EU-Rent's specification of its vocabulary business context, i.e., its use of the SBVR constructs to define the EU-Rent communities, bodies of shared meanings and vocabularies. The second subclause illustrates the contents of one of EU-Rent's vocabularies -- the EU-Rent English Vocabulary of the EU-Rent English Community (a speech community) -- along with its associated rule sets.

Limitation of scope

Some entries in the examples have been left informal in order to limit the overall size of the case study. They might, in a 'real' SBVR model, be expanded into substantial formal structures.

E.2.1 The EU-Rent Vocabulary Business Context

The entries in this subclause define the business context of EU-Rent's several vocabularies -- i.e., its communities and subcommunities, its vocabularies and bodies of shared meanings, and how these elements inter-relate. Figure E.2 presents a partial instance diagram of the concepts and facts that express EU-Rent's vocabulary business context.

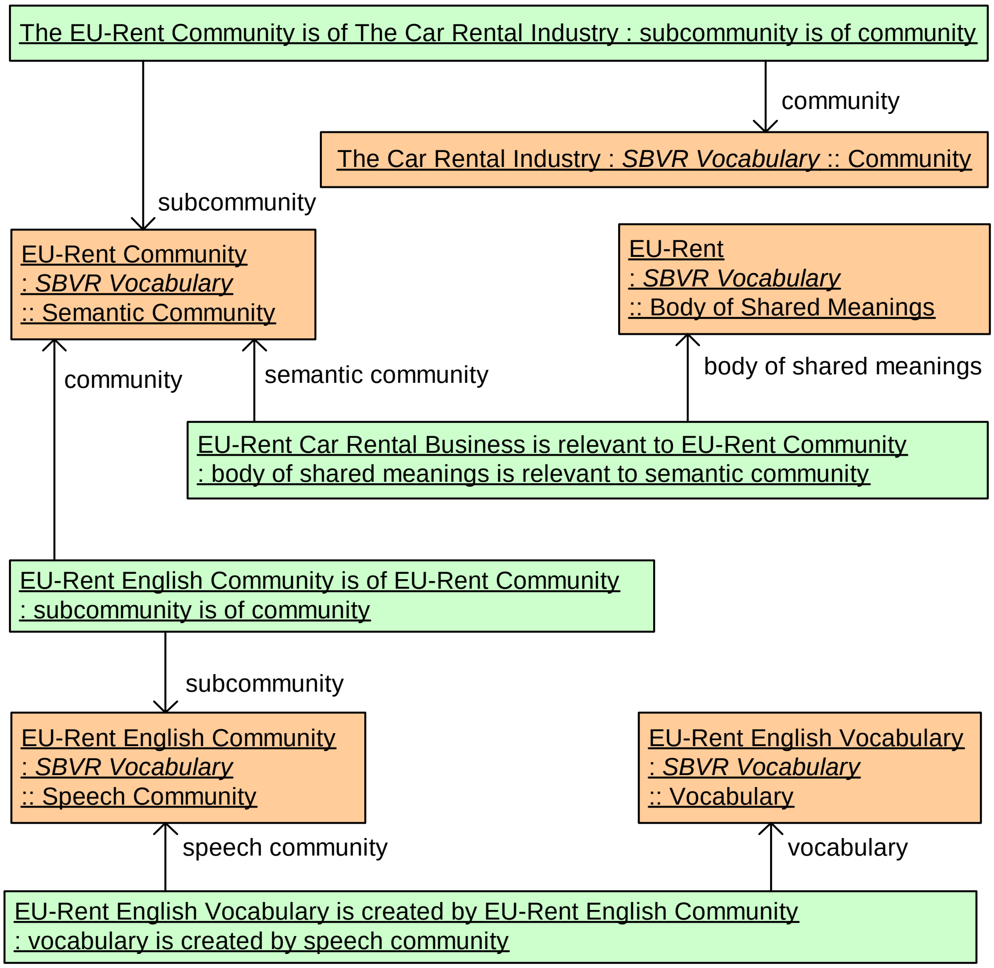


Figure E.2 - Instance diagram of concepts and facts expressing EU-Rent's vocabulary business context

E.2.1.1 EU-Rent Semantic Community

The Car Rental Industry

Definition: the semantic community that *is* the group of people who work in the business of renting cars

Car Rental Business

Definition: the body of shared meanings that *is* the set of concepts that are generally accepted as important across The Car Rental Industry

Necessity: Car Rental Business *is relevant to* The Car Rental Industry.

Necessity: Car Rental Business *is relevant to* The EU-Rent Community.

EU-Rent

Definition: the international car rental company that trades as "EU-Rent."

EU-Rent Car Rental Business

Definition: the body of shared meanings that is the set of concepts that are important to The EU-Rent Community

Necessity: EU-Rent Car Rental Business is relevant to The EU-Rent Community.

The EU-Rent Community

Definition: the semantic community that comprises all employees of EU-Rent and all others who share their body of concepts and use their vocabularies

Necessity: The EU-Rent Community is a subcommunity of The Car Rental Industry

EU-Rent HQ

Definition: the organization unit that is EU-Rent's world headquarters and management company

Description: EU-Rent HQ sets global policy and owns the world-wide reservations system.

EU-Rent HQ Staff

Definition: the community that is the set of employees of EU-Rent HQ

Necessity: The EU-Rent HQ Staff is a subcommunity of the EU-Rent community.

E.2.1.2 EU-Rent Speech Communities and Vocabularies

E.2.1.2.1 Language-independent Vocabularies

ISO Dictionary of International Symbols

Definition: the vocabulary that is defined by ISO, of graphical symbols that have consistent meanings regardless of which natural languages they are used with

Synonym: ISO-DIS

Reference Scheme: ISO-DIS index

Note: This is a fictitious standard. Work in this area is going on within ISO, but no standards have yet been published.

ISO-DIS

Synonym: ISO Dictionary of International Symbols

E.2.1.2.2 EU-Rent English Community

The EU-Rent English Community

Definition: the speech community that is within The EU-Rent Community and has English as its primary natural language

Description: Most members of The EU-Rent English Community are employees of: EU-Rent HQ, EU-Rent CA, EU-Rent GB, EU-Rent IE, EU-Rent US; trading partners of those EU-Rent companies; other EU-Rent companies who interact in English with them.

Necessity: The EU-Rent English Community is of The EU-Rent Community.

Car Rental Industry Standard Glossary

Definition: the vocabulary that is defined in English by The Car Rental Industry

Synonym: CRISG

Reference Scheme: CRISG terms

CRISG

Synonym: Car Rental Industry Standard Glossary

Merriam-Webster Unabridged Dictionary

Definition: the vocabulary that is the 2004 edition, published by Merriam-Webster

Synonym: MWU

Reference Scheme: MWU terms

MWU

Synonym: Merriam-Webster Unabridged Dictionary

EU-Rent English Vocabulary

Definition: the vocabulary that *is created by* The EU-Rent English Community

Necessity: EU-Rent English Vocabulary *incorporates* MWU.

Necessity: EU-Rent English Vocabulary *incorporates* ISO-DIS.

Necessity: EU-Rent English Vocabulary *incorporates* ISO- CRISG.

Necessity: CRISG has precedence over MWU.

Note: The necessity above means that if a signifier used in the EU-Rent English Vocabulary is implicitly understood - i.e., does not have an owned or explicitly adopted definition - it should first be looked up in CRISG, and if it is not there, then in MWU.

E.2.1.2.3 EU-Rent German Community

The EU-Rent German Community

Definition: the speech community that *is within* The EU-Rent Community and has German as its primary natural language

Description: Most members of The EU-Rent German Community are employees of: EU-Rent DE; trading partners of EU-Rent DE; other EU-Rent companies who interact, in German, with EU-Rent DE

Necessity: The EU-Rent German Community *is of* The EU-Rent Community.

Deutsches Universalwörterbuch

Definition: the vocabulary that is the 2003 edition published by Duden

Synonym: DUW

Reference Scheme: DUW terms

DUW

Synonym: Deutsches Universalwörterbuch

Glossar für Autovermietunggeschäft

Definition: the vocabulary that *is defined in German by* The Car Rental Industry

Synonym: GFA

Reference Scheme: GFA terms

GFA

Synonym: Glossar für Autovermietunggeschäft

EU-Rent German Vocabulary

Definition:	the vocabulary that *is created by* the EU-Rent German Community
Necessity:	EU-Rent German Vocabulary *incorporates* DUW.
Necessity:	EU-Rent German Vocabulary *incorporates* GFA.
Necessity:	EU-Rent German Vocabulary *incorporates* ISO-DIS.
Necessity:	GFA has precedence over DUW.
Note:	The necessity above means that if a signifier used in the EU-Rent German Vocabulary does not have an owned or explicitly adopted definition, it should first be looked up in GFA, and if it is not there, then in DUW.

E.2.2 The EU-Rent English Vocabulary and Rules

E.2.2.1 Concepts and Vocabulary

E.2.2.1.1 Car Movement

This subclause illustrates the creation of a ‘building block’ of concepts and related vocabulary, defined once and used in more than one context. car movement is used in both rental and car transfer (logistical movement of a car by EU-Rent staff).

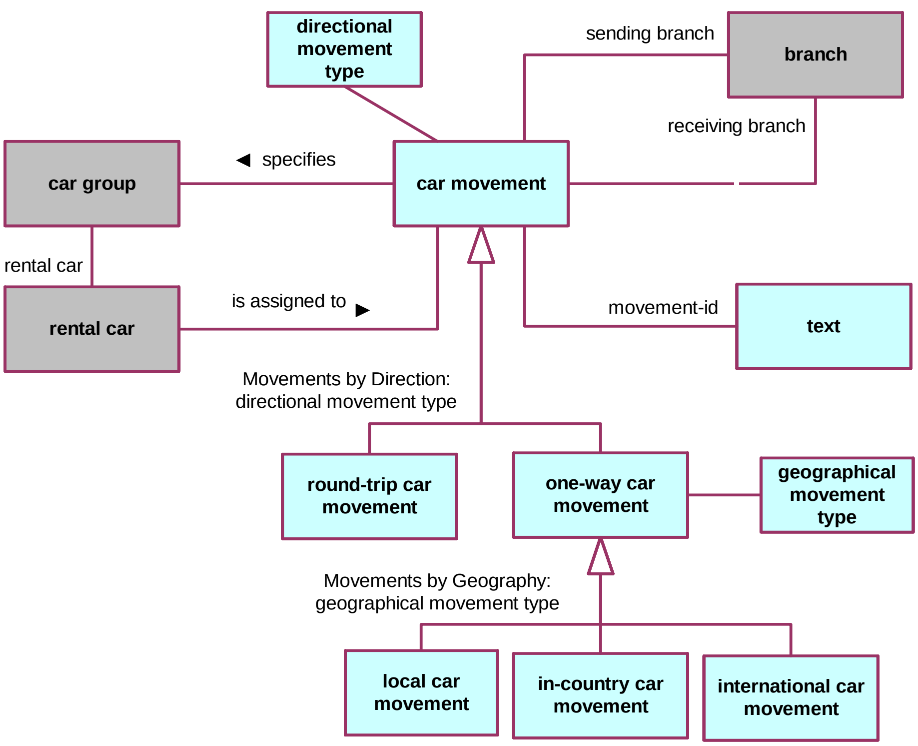


Figure E.3 - Car Movements

Issue 9948 Revise text

car movement

Definition: planned movement of a rental car of a specified car group from a sending branch to a receiving branch

Reference Scheme: movement-id

Description: A car movement meets the business requirement that a car of a given group has to be moved between branches ("we need to move a full-size car from the London City branch to the Heathrow Airport branch"). A specific car will be assigned to it at some time, not necessarily when the requirement is first identified.

Note: car movements play roles in both 'rental' and 'car transfer' and car movements are scheduled in these roles.

Issue 9449 Revise text

---

car movement *has* movement-id

Necessity: Each car movement *has* exactly one movement-id.

car movement *has* receiving branch

Necessity: Each car movement *has* exactly one receiving branch.

car movement *has* sending branch

Necessity: Each car movement *has* exactly one sending branch.

car movement *specifies* car group

Synonymous Form: car group *is specified in* car movement

Necessity: Each car movement *specifies* exactly one car group.

directional movement type

Concept Type: categorization type

Definition: concept *that specializes* the concept 'car movement' and that *classifies* a car movement based on whether the car is moved to a different branch or not

geographical movement type

Concept Type: categorization type

Definition: concept *that specializes* the concept 'one-way car movement' and that *classifies* a one-way car movement based on whether it crosses local area or international boundaries

in-country car movement

Concept Type: geographical movement type

Definition: one-way car movement that *is* not *in-area* and *is* not *international*

Note: This means that the movement is between two local areas in the same country.

Necessity: in-country car movement *is included in* Movements by Geography.

international car movement

Concept Type: geographical movement type

Definition: one-way car movement that *is international*

Necessity: international car movement *is included in* Movements by Geography.

local car movement

Concept Type: geographical movement type

Definition: one-way car movement that *is in-area*

Necessity: local car movement *is included in* Movements by Geography.

movement-id

Concept Type: role

Definition: text that *is assigned by EU-Rent as unique identifier of* car movement

Note: A given car could be moved more than once between the same two branches, perhaps even on the same day. Movement-id is needed to provide a reliable reference scheme.

Movements by Direction

Definition: segmentation that *is for* the concept 'car movement' and *subdivides* car movements based on directional movement type

Necessity: Movements by Direction *contains* the categories 'one-way movement' and 'round-trip movement.'

Movements by Geography

Definition: segmentation that *is for* the concept 'one-way car movement' and *subdivides* one-way car movements based on geographical movement type

Necessity: Movements by Geography *contains* the categories 'in-country car movement' and 'international car movement' and 'local car movement.'

one-way car movement

Concept Type: directional movement type

Definition: car movement that *is* not *round-trip*

Necessity: one-way car movement *is included in* Movements by Direction.

sending branch

Concept Type: role

Definition: branch that is the origin of a car movement

rental car *is assigned to* car movement

Necessity: At most one rental car *is assigned to* each car movement.

Necessity: The rental car that *is assigned to* a car movement *is of* ~~the~~ some ~~car group~~ ~~*specified by*~~ ~~the car movement.~~ car model that *is included in* the car group that *is specified in* the car movement

receiving branch

Concept Type: role

Definition: branch that is the destination of a car movement

round-trip car movement

Concept Type: directional movement type

Definition: car movement that *is round-trip*

Necessity: round-trip car movement *is included in* Movements by Direction.

Characteristics

car movement *being in-area*

Concept Type: characteristic

Definition: car movement *having* receiving branch that *is included in* the local area *of* the sending branch *of* the car movement

car movement *being international*

Concept Type: characteristic

Definition: car movement *having* country *of* sending branch that *is* not the country *of* receiving branch *of* the car movement

car movement *being round-trip*

Concept Type: characteristic

Definition: car movement *having* sending branch that *is* the receiving branch *of* the car movement

E.2.2.1.2 Car Transfers

"Car Transfer" illustrates two features of SBVR usage:

- Use of a "building block" as defined in car transfer includes one-way car movement.
- A trade-off of redundancy (in the sense of defining a concept both directly and indirectly) against simplification of logical formulation and representation.

For example, EU-Rent defines transferred car as the concept 'rental car that *is assigned to* the one-way car movement that *is included in* a car transfer.'

Note that both of these are matters of practice, not mandated by SBVR. Decisions on reuse and redundancy are business decisions made by the semantic community (here, EU-Rent) to help it manage its body of shared meanings and vocabularies.

Generally, derivable necessities are not restated. For example, a car movement has exactly one sending branch; a car transfer includes exactly one one-way car movement; the pick-up branch of a car transfer is the sending branch of the included one-way car movement. There is no restated necessity: car transfer *has* exactly one pick-up branch.

Again, this is a matter of practice. If a semantic community would prefer the derivable necessities to be explicit, SBVR will support it.

Issue 9450 Revise text

"Redundant" concepts are specified using structural rules (necessities). For example:

car transfer *has* transferred car

Necessity: The transferred car *of* a car transfer *is* the rental car that *is assigned to* the one-way car movement that is included in the car transfer.

One extension of the approach is carry-over of the segmentation of car transfers by geographical movement type (local, in-country, and international). The segmentation and categorization type are not repeated. The categories of car transfer are specified as corresponding to the respective categories of car movement.

Note: fact types derived from inclusion of fixed period and one-way car movement (e.g., car transfer *has* transfer pick-up branch) are not shown on the diagram, but are defined in the text below.

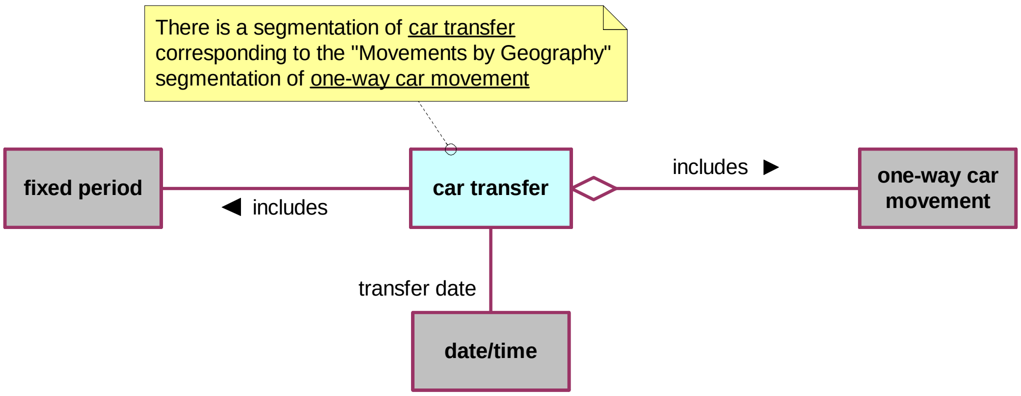


Figure E.4 - Car Transfers

car transfer

Definition: logistical action by EU-Rent, transferring a rental car between branches on a given day

car transfer *has* transfer date

Necessity: each car transfer *has* exactly one transfer date.

Issue 9450 Revise text

---

car transfer *has* transfer drop-off branch

Necessity: A car transfer *has* a transfer drop-off branch if and only if the transfer drop-off branch *is* the receiving branch *of* the one-way car movement that *is included in* the car transfer.

car transfer *has* transfer drop-off date/time

Necessity: A car transfer *has* a transfer drop-off date/time if and only if the transfer drop-off date/time *is* the actual end date/time *of* the fixed period that *is included in* the car transfer.

car transfer *has* transfer pick-up branch

Necessity: A car transfer *has* a transfer pick-up branch if and only if the transfer pick-up branch *is* the sending branch *of* the one-way car movement that *is included in* the car transfer.

car transfer *has* transfer pick-up date/time

Necessity: A car transfer *has* a transfer pick-up date/time if and only if the transfer pick-up date/time *is* the actual start date/time *of* the fixed period that *is included in* the car transfer.

car transfer *has* transferred car

Necessity: A car transfer *has* a transferred car if and only if the transferred car *is* the rental car that *is assigned to* the one-way movement that *is included in* the car transfer.

car transfer *includes* car movement

Necessity: each car transfer *includes* exactly one car movement.

car transfer *includes* fixed period

Necessity: each car transfer *includes* exactly one fixed period.

Note: EU-Rent does not schedule car transfers within their transfer dates. It wants to know, at the end of the transfer, the actual pick-up and drop-off times. By the time EU-Rent is interested in the period of the transfer, it is in the past - and so is fixed.

in-country car transfer

Definition: car transfer that *includes* an in-country car movement

international return

Definition: car transfer that *includes* an international car movement

local car transfer

Definition: car transfer that *includes* a local car movement

transfer date

Concept Type: role

Definition: date that a car transfer *is scheduled for*

Note: The transfer date is usually scheduled in advance. The pick-up date/time and drop-off date/time are the actual times during the day, notified when the transfer is completed.

Issue 11291 Revise text

---

transfer drop-off branch

Concept Type: role

Definition: branch at which the transferred car *of* a car transfer *is ~~picked up~~ dropped off*

transfer drop-off date/time

Concept Type: role

Definition: date/time when the transferred car *of* a car transfer *is actually dropped off*

transfer pick-up branch

Concept Type: role

Definition: branch from which the transferred car *of* a car transfer *is picked up*

transfer pick-up date/time

Concept Type: role

Definition: date/time when the transferred car *of* a car transfer *is actually picked up*

transferred car

Concept Type: role

Definition: rental car relocated by a car transfer

E.2.2.1.3 EU-Rent Locations

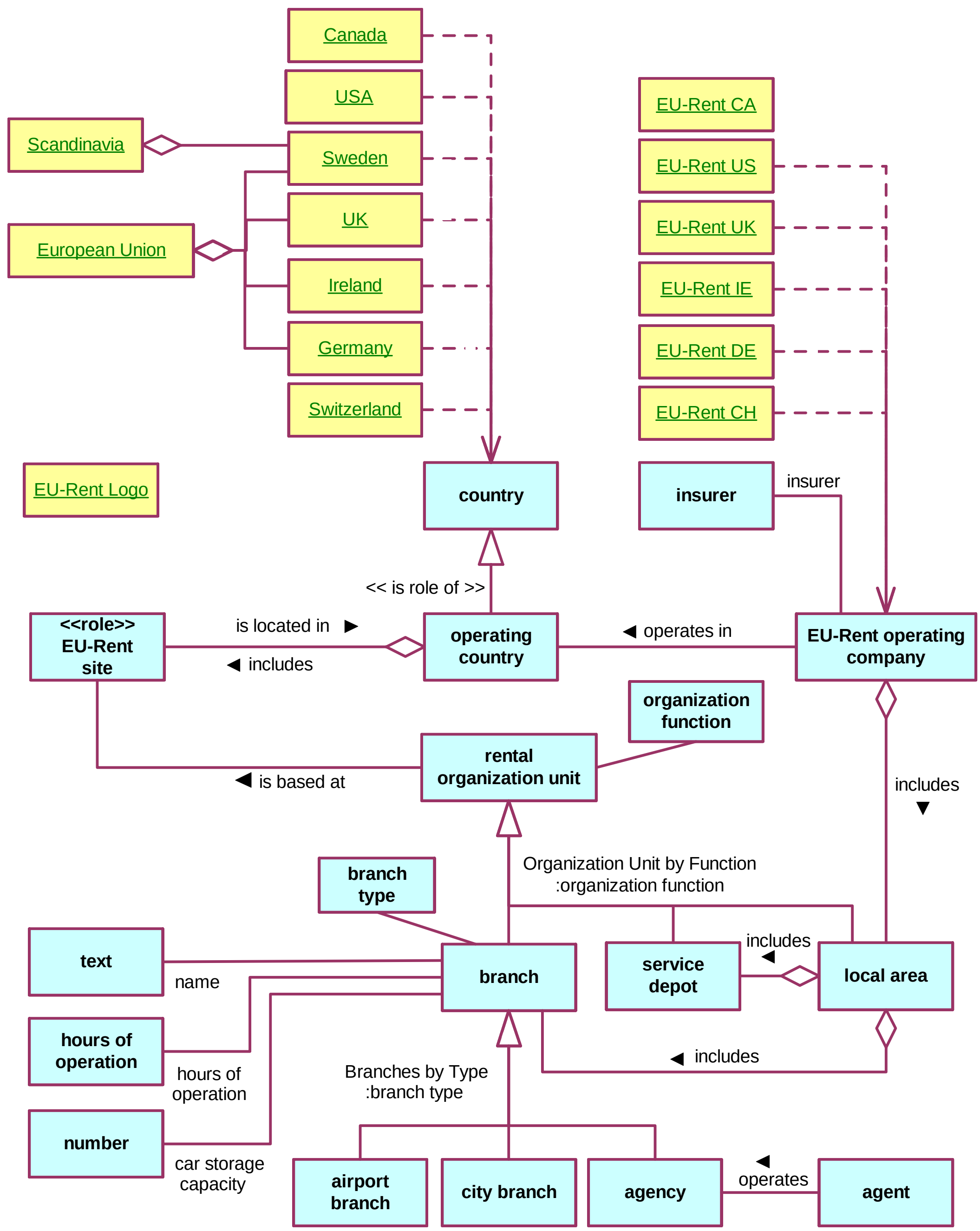


Figure E.5- EU-Rent Locations

agency

Concept Type: branch type

Definition: branch that *does* not *have an EU-Rent location* and *has minimal car storage* and *has on-demand operation*

Necessity: the concept 'agency' *is included in* Branches by Type.

agent

Definition: organization external to EU-Rent that carries out car rental business on behalf of EU-Rent

Example: Hotel, travel agent.

agent *operates* agency

Note: Operation is usually by part-time staff who will carry out the entire workflows for rental and return.

airport branch

Concept Type: branch type

Definition: branch that *has an EU-Rent location* and *has large car storage* and *has 24-7 operation*

Note: This kind of branch is usually at or near a major airport or rail terminal and has sufficient staff to have specialized roles in the workflow.

Necessity: the concept 'airport branch' *is included in* Branches by Type.

branch

Concept Type: organization function

Definition: rental organization unit that *has rental responsibility*

Necessity: the concept 'branch' *is included in* Organization Units by Function.

branch *has* car storage capacity

Concept Type: is-property-of fact type

branch *has* country

Necessity: The country *of* a branch *is* the operating country *of* the operating company that *includes* the local area that *includes* the branch.

branch *has* hours of operation

Concept Type: is-property-of fact type

branch *has* name

Concept Type: is-property-of fact type

Issue 9449 Add text / revise text

---

branch *is included in* local area

Concept Type: partitive fact type

Synonymous Form: local area *includes* branch

Necessity: Each branch *is included in* exactly one local area.

branch type

Definition: concept that *specializes* the concept 'branch' and that *classifies* a branch based on hours of operation and car storage capacity

Branches by Type

Definition: segmentation that *is for* the concept 'branch' and *subdivides* branches based on branch type

Necessity: Branches by Type *contains* the categories 'airport branch' and 'city branch' and 'agency.'

car storage capacity

Concept Type: role

Definition: number *of* rental cars that *can be stored at* the EU-Rent site that *is the base for* a branch

Note: Some of the capacity at a branch's site might be taken up by cars that are not available for rental -- e.g., cars awaiting service or transfer to other branches; hotel guests' cars.

Issue 11291 Replace text

---

city branch

Concept Type: branch type

Definition: branch that *has an EU-Rent location* and *has moderate car storage* and *has long business hours*

Note: This kind of branch is usually in a city center and has small numbers of staff who are interchangeable in the workflow.

Necessity: The concept 'city branch' *is included in* Branches by Type.

country

Source: MWU (1,2b) ["country"]

Note: Has pre-defined population (below)

EU-Rent Logo

Definition: upper-case 'EU' followed by a hyphen followed by upper-case 'R' followed by lower-case 'ent' all in the company's custom-designed font and using the company's standard color on a white ground

EU-Rent operating company

Definition: operating company of EU-Rent

Synonym: operating company

Note: In each operating country EU-Rent has an EU-Rent operating company that:

• adapts global policy to local regulation, custom, and practice

• selects which car models will be purchased for each car group

• sets rental tariffs

Note: Has pre-defined population (below)

EU-Rent operating company *includes* local area

Synonymous Form: local area *is included in* EU-Rent operating company

EU-Rent operating company *operates in* operating country

Synonymous Form: operating company *has* operating country

~~Necessity:~~ ~~Each EU-Rent site *is located in* exactly one operating country.~~

EU-Rent site

Concept Type: role

Definition: location *used by* EU-Rent

EU-Rent site *is base for* rental organization unit

Synonymous Form: rental organization unit *is based at* EU-Rent site

EU-Rent site *is located in* operating country

Synonymous Form: operating country *includes* EU-Rent site

Necessity: Each EU-Rent site *is located in* exactly one operating country

European Union

Definition: the geopolitical area that *is composed of* Sweden and Germany and Ireland and UK and ...

hours of operation

Definition: the times during which a facility is open for business

Example: 24 hours per day, 7 days a week; 7:00 am to 8:00 pm; on demand.

Issue 9874 Add text

---

insurer

Source: MWU ["insurer']

Issue 9449 Revise text

---

local area

Concept Type: organization function

Definition: rental organization unit that *has area responsibility*

Description: A local area contains a number of branches for rental car pick-up and return and a number of service depots that maintain and repair EU-Rent's rental cars.

Necessity: service depot *is included in* Organization Units by Function.

~~**local area *includes* branch**~~

~~Synonymous Form:~~ ~~branch *is included in* local area~~

local area *includes* service depot

Synonymous Form: service depot *is included in* local area

local area *is included in* operating company

Concept Type: partitive fact type

Synonymous Form: operating company *includes* local area

Necessity: Each local area *is included in* exactly one EU-Rent operating company.

location

Source: MWU (1a) [“location’]

name

Concept Type: role

General Concept: text

Source: MWU (1a) [“name”]

operating company

See: EU-Rent operating company

Issue 9874 Add text

operating company *has* insurer

Issue 9449 Add text

operating country

Concept Type: role

Definition: country in which EU-Rent does business

Necessity: Each operating country *has* exactly one currency

organization function

Concept Type: categorization type

Definition: concept that *specializes* the concept ‘rental organization unit’ and that *classifies* a rental organization unit by its functional role in EU-Rent

Organization Units by Function

Definition: segmentation that *is for* the concept ‘rental organization unit’ and *subdivides* rental organization units based on organization function

Necessity: Organization Units by Function *contains* the categories ‘branch’ and ‘local area’ and ‘service depot’.

rental organization unit

Concept Type: role

Definition: organization unit that operates part of EU-Rent’s car rental business

rental organization unit *is based at* EU-Rent site

Concept Type: associative fact type

Necessity: Each rental organization unit *is based at* exactly one EU-Rent site.

service depot

Concept Type: organization function

Definition: rental organization unit that *has servicing responsibility*

Necessity: service depot *is included in* Organization Units by Function.

service depot *is included in* local area

Concept Type: partitive fact type

Necessity: Each service depot *is included in* exactly one local area.

Scandinavia

Definition: the geographic area that *is composed of* Sweden and ...

E.2.2.1.3.1 Characteristics

rental organization unit *having 24-7 operation*

Concept Type: characteristic

Definition: the rental organization unit *has* hours of operation that are 24 hours per day, 7 days per week

rental organization unit *having a EU-Rent location*

Concept Type: characteristic

Definition: the rental organization unit *is based at* an EU-Rent site that is owned by EU-Rent

Note: Some things *are based at* EU-Rent sites that are owned by third parties such as hotels and travel agents.

rental organization unit *having area responsibility*

Concept Type: characteristic

Definition: the rental organization unit includes organization units for which it has responsibility to coordinate operations and ensure resources

rental organization unit *having large car storage*

Concept Type: characteristic

Definition: the rental organization unit *has* car storage that *accommodates hundreds of* rental cars

rental organization unit *having long business hours*

Concept Type: characteristic

Definition: the rental organization unit *has* hours of operation that *correspond to an extended business day*

Example: 7:00 am to 8:00 pm, six days per week.

rental organization unit *having minimal car storage*

Concept Type: characteristic

Definition: the rental organization unit *has* car storage that *can accommodate a small number of* rental cars

rental organization unit ***having moderate car storage***

Concept Type: characteristic

Definition: the rental organization unit *has* car storage that *can accommodate tens of* rental cars

rental organization unit ***having on-demand operation***

Concept Type: characteristic

Definition: the rental organization unit *has* hours of operation that *are flexible in response to customer demand*

rental organization unit ***having rental responsibility***

Concept Type: characteristic

Definition: the rental organization unit is responsible for operation of customer-facing rental business

rental organization unit ***having servicing responsibility***

Concept Type: characteristic

Definition: the rental organization unit is responsible for maintenance and servicing of rental cars

Pre-defined population: country

Canada

Concept Type: individual concept

General Concept: country

Germany

Concept Type: individual concept

General Concept: country

Synonym: DE

Ireland

Concept Type: individual concept

General Concept: country

Sweden

Concept Type: individual concept

General Concept: country

Switzerland

Concept Type: individual concept

General Concept: country

Synonym: CH

UK

Concept Type: individual concept

General Concept: country

Synonym: United Kingdom

United States

Concept Type: individual concept

General Concept: country

Synonym: USA

Pre-defined population: EU-Rent operating company

EU-Rent CA

Definition: the EU-Rent operating company that *is located in* Canada

EU-Rent DE

Definition: the EU-Rent operating company that *is located in* Germany

EU-Rent IE

Definition: the EU-Rent operating company that *is located in* Ireland

EU-Rent UK

Definition: the EU-Rent operating company that *is located in* UK

EU-Rent US

Definition: the EU-Rent operating company that *is located in* United States

E.2.2.1.4 Car Specifications

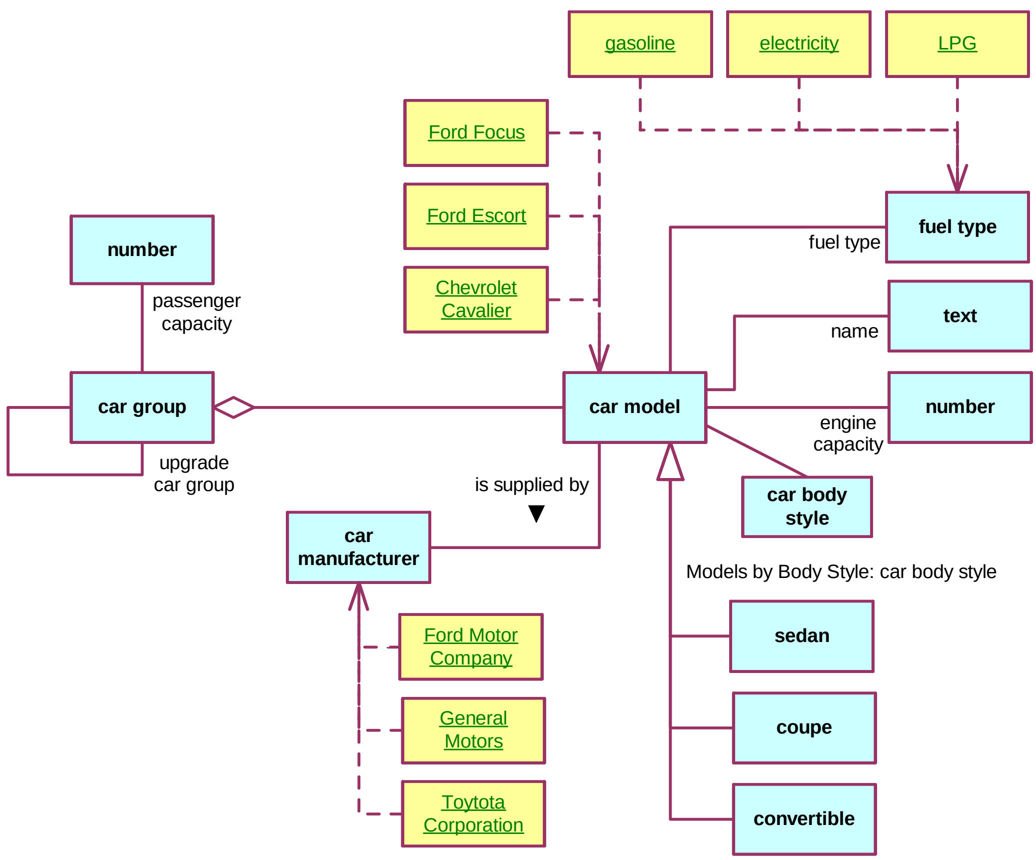


Figure E.6 - Car Specifications

car body style

Concept Type: categorization type

Definition: concept that *specializes* the concept 'car model' and that *classifies* a car model *based on* industry-defined criteria

Source: CRISG (2a) ["body style"]

Issue 9449 Revise text

---

car group

Source: CRISG ["rate group"]

Note: Different models of car are offered for rental, organized into groups which establish a price point.

Necessity: Each car model that *is included in* a car group is charged at the rental rates of the car group.

car group *has* passenger capacity

Concept Type: is-property-of fact type

Definition: the car group *has* the passenger capacity that defines the capacity of a car of the car group, with the driver counting as One

car group *has* upgrade car group

Definition: the car group *has* an upgrade car group that is used when no car of a requested car group is available

car manufacturer

Definition: producer of cars that EU-Rent has decided to do business with

Note: Has pre-defined population (below)

car model

Source: CRISG ["car model"]

Note: Has pre-defined population (below)

Note: Cars of a given model are all built to the same specification, e.g., body style, engine size, fuel type. EU-Rent bases its model names on those assigned by the car manufacturers, but sometimes has to extend them to distinguish models with different engine sizes and numbers of doors.

Reference Scheme: name *of* car model

car model *is included in* car group

Synonymous Form: car group *includes* car model

Necessity: Each car model *is included in* exactly one car group.

car model *has* engine capacity

Concept Type: is-property-of fact type

Issue 9467 Revise text / remove text

car manufacturer *supplies* car model

~~Synonymous Form:~~ ~~car manufacturer *supplies* car model~~

Necessity: Each car model *is supplied by* exactly one car manufacturer.

car model *has* fuel type

Concept Type: is-property-of fact type

Definition: Some car models can have more than one fuel – e.g., can switch between gasoline and LPG, or between electricity and gasoline.

Necessity: Each car model *has* at least one fuel type.

car model *has* name

Concept Type: is-property-of fact type

convertible

Concept Type: car body style

Source: CRISG ["convertible"]

Necessity: convertible *is included in* Models by Body Style.

coupe

Concept Type: car body style

Source: CRISG ["coupe"]

Necessity: coupe *is included in* Models by Body Style.

engine capacity

Concept Type: role

Definition: number that indicates the engine cylinder capacity in cubic centimeters

Source: CRISG ["engine size"]

fuel type

Source: CRISG ["fuel type"]

Note: Has pre-defined population (below)

Models by Body Style

Definition: segmentation that *is for* the concept 'car model' and *subdivides* car models based on car body style

Necessity: Models by Body Style *contains* the categories 'convertible' and 'coupe' and 'sedan'.

passenger capacity

Concept Type: role

Definition: number that is the count of adults, including the driver, that the car can comfortably hold

sedan

Concept Type: car body style

Source: CRISG ["sedan"]

Necessity: sedan *is included in* Models by Body Style.

upgrade car group

Concept Type: role

Definition: car group from which cars may be offered for rental if there are no cars available in another requested car group

Pre-defined Population: car model

Chevrolet Cavalier

Concept Type: individual concept

General Concept: car model

Ford Focus

Concept Type: individual concept

General Concept: car model

Ford Escort

Concept Type: individual concept

General Concept: car model

Pre-defined Population: car group

Economy

Source: CRISG ["economy group"]

General Concept: car group

Compact

Source: CRISG ["compact group"]

General Concept: car group

Intermediate

Source: CRISG ["intermediate group"]

General Concept: car group

Full Size

Source: CRISG ["fullsize group"]

General Concept: car group

Pre-defined Population: car manufacturer

Ford Motor Company

Concept Type: individual concept

General Concept: car manufacturer

General Motors

Concept Type: individual concept

General Concept: car manufacturer

Toyota Corporation

Concept Type: individual concept

General Concept: car manufacturer

Pre-defined Population: fuel type

Electricity

Concept Type: individual concept

Source: CRISG ["electric fuel"]

Gasoline

Concept Type: individual concept

Source: CRISG ["gasoline"]

General Concept: fuel type

Synonym: petrol [UK]

Synonym: benzin [DE]
Synonym: essence [FR]

LPG
Concept Type: individual concept
General Concept: fuel type
Source: CRISG ["liquefied petroleum gas"]

E.2.2.1.5 Rentals

There are some trade offs of redundancy and reuse (and, hence, a bigger vocabulary) against some simplification of formulation and expression.

Note: fact types derived from inclusion of rental period and car movement (e.g., rental *has* pick-up branch) are not shown on the diagram, but are defined in the text.

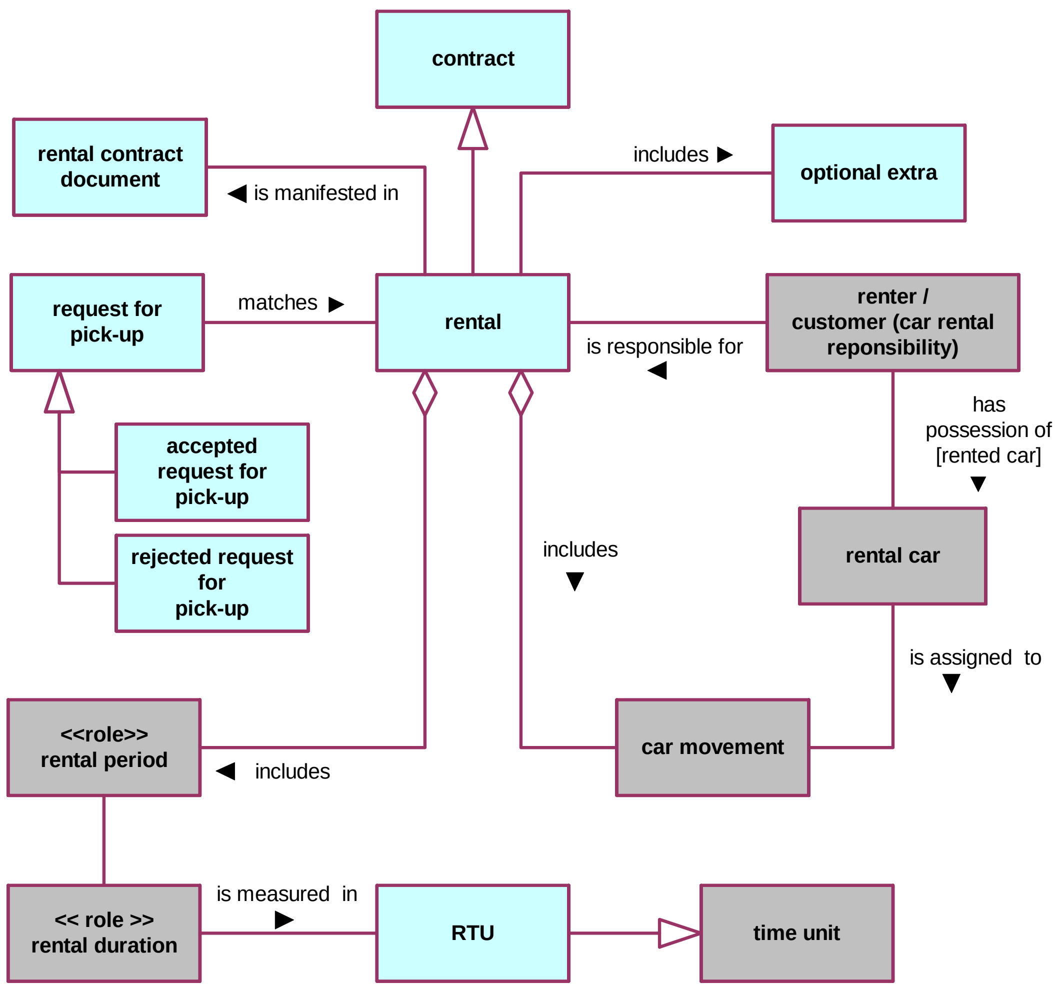


Figure E.7 - Rentals

accepted request for pick-up

Definition:	request for pick up that is accepted by EU-Rent
Necessity:	The request for pick-up *matches* exactly one rental.
Necessity:	The renter *presents* a valid credit card.
Necessity:	The renter *provides* current contact details.
Necessity:	Each driver *of* the rental *has* a valid driver license.

Issue 9449 Revise text

actual pick-up date/ time

Concept Type:	role
General Concept:	date/time
Definition:	date/time when the rented car *of* a rental is picked up by the renter

actual return date/time

Concept Type:	role
General Concept:	date/time
Definition:	date/time when a rented car is returned to EU-Rent

current contact details

Concept Type:	role
Definition:	contact details that have been confirmed as up-to-date by the renter

optional extra

Definition:	Item that may be added to a rental at extra charge if the renter so chooses
Example:	One-way rental, fuel pre-payment, additional insurances, fittings (child seats, satellite navigation system, ski rack)
Source:	CRISG ["optional extra"]

Issue 11291 Revise text

pick-up branch

Concept Type:	role
General Concept:	branch
Necessity:	The pick-up branch *of* a rental is not *changed*.
Note:	If the renter wishes to change the pick-up branch of a rental, EU-Rent regards it as a cancellation and a new rental.

rejected request for pick-up

Definition:	request for pick up that is rejected by EU-Rent

rental

Definition:	contract with a renter specifying use of some car of a car group for a rental period and a car movement

Dictionary Basis:	contract for use of a rental car by a renter for an agreed period under the rental company's terms and conditions for rental. [CRISG]

rental contract document

Definition:	information artifact that is the manifestation of a rental

rental duration

Concept Type:	role
Definition:	duration used to calculate a rental charge

rental duration *is measured in* rental time unit

Necessity:	Each rental duration *is measured in* a whole number of rental time units.
Example:	A rental with an end date/time that was 11 days and 7 hours after its start date/time would have a rental duration of 1 x 1-week RTU plus 5 x 1-day RTU. If EU-Rent were to introduce a 3-day RTU, this would change to 1 x 1-week RTU plus 1 x 3-day RTU plus 2 x 1-day RTU.

Issue 9450 Revise text

rental *has* actual pick-up date/time

Definition:	rental *has* actual pick-up date/time
Necessity:	A rental *has* an actual pick-up date/time if and only if the actual pick-up date/time *is* the start date/time *of* the rental period that *is included in* the rental.

rental *has* actual return date/time

Necessity:	A rental *has* an actual return date/time if and only if the actual return date/time *is* the end date/time *of* the rental period that *is included in* the rental.

rental *has* pick-up branch

Necessity:	A rental *has* a pick-up branch if and only if the pick-up branch *is* the sending branch *of* the car movement that *is included in* the rental.

rental *has* rental duration

Necessity:	A rental *has* a rental duration if and only if the rental duration *is* the duration *of* the period that *is* the rental period that *is included in* the rental.

rental *has* requested car group

Necessity:	A rental *has* a requested car group if and only if the requested car group *is* the car group that *is specified in* the car movement that *is included in* the rental.
Possibility:	The requested car group *of* an advance rental *is changed before* the actual pick-up date/time *of* the advance rental.
Necessity:	The requested car group *of* an advance rental *is* not *changed after* the actual pick-up date/time *of* the advance rental.

rental *has* return branch

Necessity: A rental *has* a return branch if and only if the return branch *is* the receiving branch *of* the car movement that *is included in* the rental

rental *has* scheduled pick-up date/time

Necessity: A rental *has* a scheduled pick-up date/time if and only if the scheduled pick-up date/time *is* the scheduled start date/time *of* the rental period that *is included in* the rental.

rental *has* scheduled return date/time

Necessity: A rental *has* a scheduled return date/time if and only if the scheduled return date/time *is* the scheduled end date/time *of* the rental period that *is included in* the rental.

Issue 9449 Revise text

---

rental *includes* car movement

Concept Type: partitive fact type

Synonymous Form: car movement *is included in* rental

Necessity: Each rental *includes* exactly one car movement

Note: The car movement may be changed by changing the return branch.

rental *includes* rental period

Concept Type: partitive fact type

Synonymous Form: rental period *is included in* rental

Necessity: Each rental *includes* exactly one rental period.

Note: The rental period may be changed by rescheduling at the renter's request, by early or late arrival for rental, and by late return from rental.

rental *is manifested in* rental contract document

Concept Type: associative fact type

rental period

Concept Type: role

Definition: variable period that *is included in* a rental

rented car

Concept Type: role

Definition: rental car that is assigned a rental

rented car *is assigned to* rental

Synonymous Form: rental *has* rented car

Necessity: A rented car *is assigned to* a rental if and only if the rented car *is* the rental car that *is assigned to* the car movement that *is included in* the rental.

renter *has possession of* rented car

Definition: the renter has the rented car for use on rental

Synonymous Form: rented car *is in the possession of* renter

request for pick-up

Definition: request from a renter to pick up the rental car of a rental that has been reserved by him

request for pick-up *matches* rental

Necessity: The rental *is assigned* (has a rental car assigned to it).

Necessity: The pick-up branch *of* the rental is the branch at which the request is made.

Necessity: The renter *of* the rental is the person making the request for pick-up.

Necessity: The scheduled start date *of* the rental is the day of the request for pick-up.

Note: This entry is partly informal in order to limit the case study size.

requested car group

Concept Type: role

Definition: car group that is requested for a rental

Necessity: *At* a given date/time each rental *has* exactly one requested car group.

return branch

Concept Type: role

Definition: branch stipulated in the rental contract for return of the rented car

Note: If the renter does not return the car to the location of this branch, a penalty charge will be levied.

Necessity: Each rental *has* exactly one return branch *at* a given date/time.

Possibility: The return branch *of* a rental *is changed before* the actual return date/time *of* the rental.

scheduled pick-up date/time

Definition: date/time at which a rented car is scheduled to be picked up from EU-Rent

Note: *The possibilities and necessities for changing the* start date/time *of* variable period *apply to* scheduled pick-up date/time *of* rental.

scheduled return date/time

Definition: date/time at which a rented car is scheduled to be returned to EU-Rent

Note: *The possibilities and necessities for changing the* end date/time *of* variable period *apply to* scheduled return date/time.

valid credit card

Concept Type: role

Definition: credit card that *is acceptable for payment of* the rental charges *of* the rental for which it is presented

Necessity: The card is of a type that EU-Rent accepts.

Necessity: “Expiry date” on the valid credit card *is after* the scheduled end date/time *of* the rental.

Necessity: “Cardholder” is the person presenting the card.

Note: This entry is informally defined in order to limit the case study size.

valid driver license

Concept Type:	role
Definition:	driver license that is acceptable for the rental for which it is presented
Necessity:	"Expiry date" on the valid driver license *is after* the scheduled end date/time *of* the rental.
Necessity:	"Driver" is the person presenting the license.
Necessity:	The rented car falls within "vehicle types."
Necessity:	The license is legally acceptable in the country of the pick-up branch.
Note:	This entry is informally defined in order to limit the case study size.

E.2.2.1.6 Rental Categorization

This subclause defines some categorizations of rental, which enable some subsequent simplification of formulations and representations.

It also introduces the use of characteristic type for defining states - see rental state and advance rental state.

.

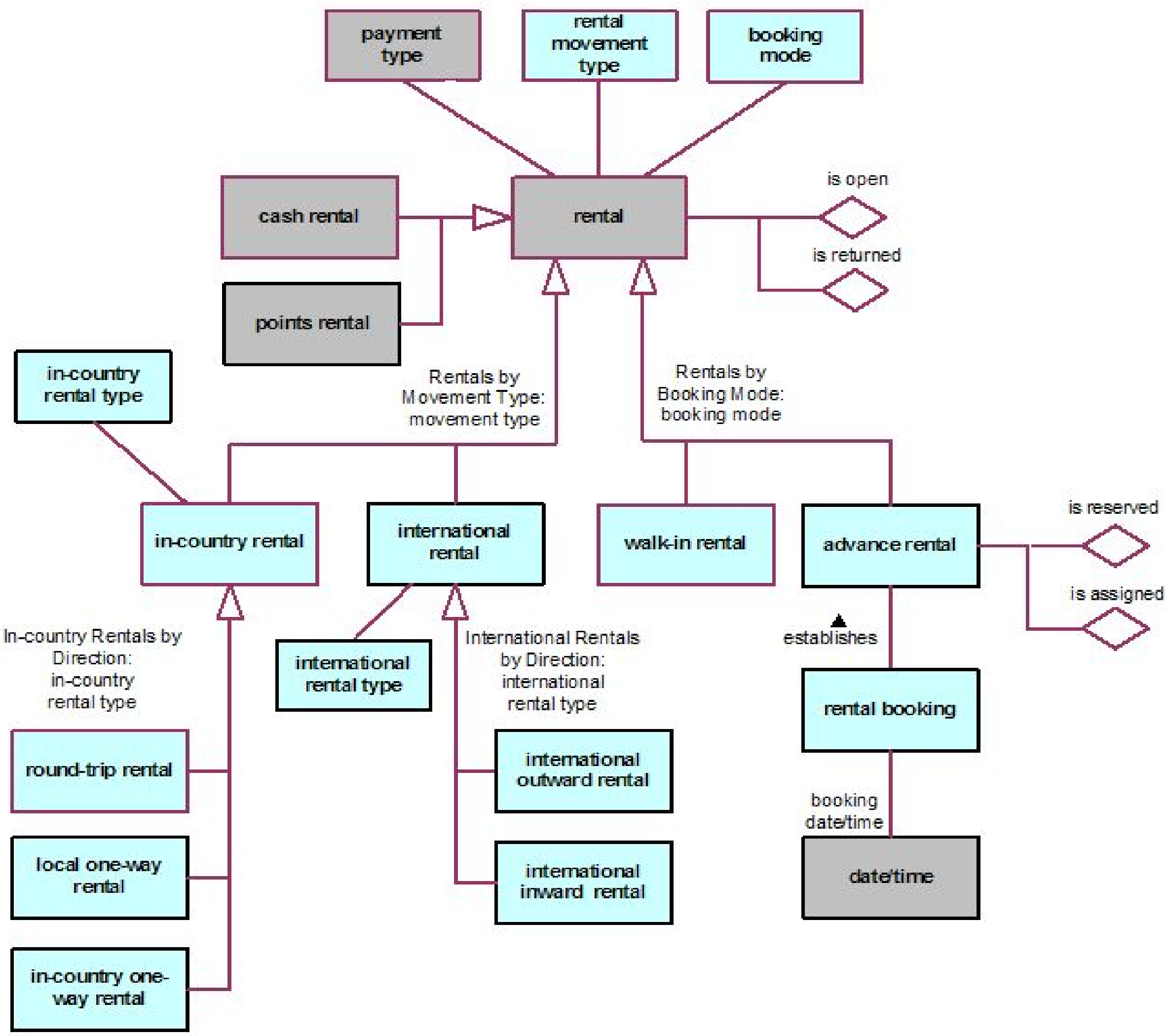


Figure E.8- Rental Categories

advance rental

Concept Type:	booking mode
Definition:	rental that is contracted with EU-Rent on or earlier than the day before the scheduled pick-up date/time *of* the rental
Necessity:	Each advance rental *specifies* exactly one car group *at* a given date/time.
Possibility:	The car group *specified for* an advance rental *is changed before* the actual pick-up date/time *of* the advance rental.
Necessity:	The car group *specified for* an advance rental *is* not *changed after* the actual pick-up date/time *of* the advance rental.
Necessity:	The concept 'advance rental' *is included in* Rentals by Booking Mode.

Issue 11291 Revise verb form

advance rental *is assigned*

Concept Type: advance rental state

Definition: advance rental *having* a car movement that *has an assigned* rented car that *has not yet been picked up*

advance rental *is reserved*

Concept Type: advance rental state

Definition: advance rental *having* a car movement that does not *have* an *assigned* rented car

advance rental state

Concept Type: characteristic type

booking date/time

Concept Type: role

Definition: date/time when rental booking *is accepted by* EU-Rent

booking mode

Concept Type: categorization type

Definition: concept that *specializes* the concept 'rental' and that *classifies* a rental whether it is booked in advance or not

car model *is requested for* rental

Synonymous Form: rental *requests* car model

Necessity: Each rental *requests* at most one car model.

Possibility: The car model *specified for* an advance rental *is changed before* the actual pick-up date/time *of* the advance rental.

Necessity: The car model *specified for* an advance rental *is* not *changed after* the actual pick-up date/time *of* the advance rental.

Issue 11291 Revise text

in-country one-way rental

Concept Type: rental movement type

Definition: one-way rental that *includes* an in-country car movement

Note: This type of rental is between branches in different local areas in the same country.

Necessity: The concept 'in-country one-way rental' *is included in* In-Country Rentals by Direction.

Issue 9449 Add text

in-country rental

General Concept: rental

in-country rental type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'in-country rental' and that *classifies* an in-country rental based on whether it is round trip, within a local area or between local areas in the same country

In-country Rentals by Direction

Definition: segmentation that *is for* the concept 'rental' and *subdivides* rentals based on movement type

Necessity: In-Country Rentals by Direction *contains* the categories 'round-trip rental' and 'local one-way rental' and 'in-country one-way rental.'

Issue 11291 Revise text

---

international rental

Concept Type: rental movement type

Definition: one-way rental that *includes* an international car movement

Note: This type of rental is between branches in different countries.

Necessity: The concept international rental *is included in* Rentals by Movement Type.

international inward rental

Concept Type: international rental type

Definition: international rental that *has* country *of* the return branch *of* the rental that *is* the country of registration *of* the rented car *of* the rental

International Rentals by Direction

Definition: segmentation that *is for* international rental and *subdivides* rental based on international rental type

Necessity: International Rentals by Direction *contains* the categories 'international inward rental' and 'international outward rental.'

international outward rental

Concept Type: international rental type

Definition: international rental that *has* country *of* the pick-up branch *of* the rental that *is* the country of registration *of* the rented car *of* the rental

international rental type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'international rental' and that *classifies* an international rental based on whether its direction is to or from the country *of* registration *of* the rented car

---

Issue 11291 Revise text

---

local one-way rental

Concept Type: rental movement type

Definition: one-way rental that *includes* a local car movement

Note: This type of rental is between branches within a local area.

Necessity: The concept local one-way rental *is included in* In-Country Rentals by Direction.

one-way rental

Concept Type: rental that *includes* a one-way car movement

rental booking

Source: CRISG ["reservation"]

Synonym: reservation

Definition: acceptance by EU-Rent of a request from a renter for an advance rental.

Note: The request informs EU-Rent of the car group required, the scheduled pick-up date/time and scheduled return date/time, and the pick-up branch and return branch, and provides details of the renter.
Optionally, a specific car model within the required car group may be requested.

Issue 9467 Remove text

rental booking ***establishes*** advance rental

Concept Type: associative fact type

~~Synonymous Form: advance rental *is established by* rental booking~~

Necessity: Each advance rental *is established by* exactly one rental booking.

rental booking ***has*** booking date/time

Concept Type: is-property-of fact type

Necessity: Each rental booking *has* exactly one booking date/time.

Necessity: The booking date/time *of* the rental booking that *establishes* a cash rental *is before* the scheduled pick-up date/time *of* the rental.

Necessity: The booking date/time *of* the rental booking that *establishes* a points rental *is* at least 5 days *before* the scheduled pick-up date/time *of* the rental.

Issue 9449 Revise text
Issue 11291 Revise verb form

rental ***is open***

Concept Type: rental state

Definition: the rental *has* a rented car that *is in possession of* the renter and the end date/time *of* the grace period *of* the rental *is in the future*

rental ***is returned***

Concept Type: rental state

Definition: the rented car *of* the rental *has been returned from rental to* a branch

rental movement type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'rental' and that *classifies* a rental based on whether it is within a country or between countries

rental state

Concept Type: characteristic type

Rentals by Booking Mode

Definition: segmentation that *is for* the concept 'rental' and *subdivides* rentals based on booking mode

Necessity: Rentals by Booking Mode *contains* the categories 'advance rental' and 'walk-in rental.'

Rentals by Movement Type

Definition: segmentation that *is for* the concept 'rental' and *subdivides* rentals based on rental movement type

Necessity: Rentals by Movement Type *contains* the categories 'in-country rental' and 'international rental.'

reservation

Synonym: rental booking

Issue 11291 Revise text

round-trip rental

Concept Type: rental movement type

Definition: rental that *includes* a round-trip car movement

Note: In this type of rental the pick-up branch is the return branch.

Necessity: The concept round-trip rental *is included in* In-Country Rentals by Direction.

walk-in rental

Concept Type: booking mode

Definition: rental that is contracted with EU-Rent on the day that the car is picked up

Necessity: The concept walk-in rental *is included in* Rentals by Booking Mode.

E.2.2.1.7 Rental Pricing

Issue 9449 Replace figure

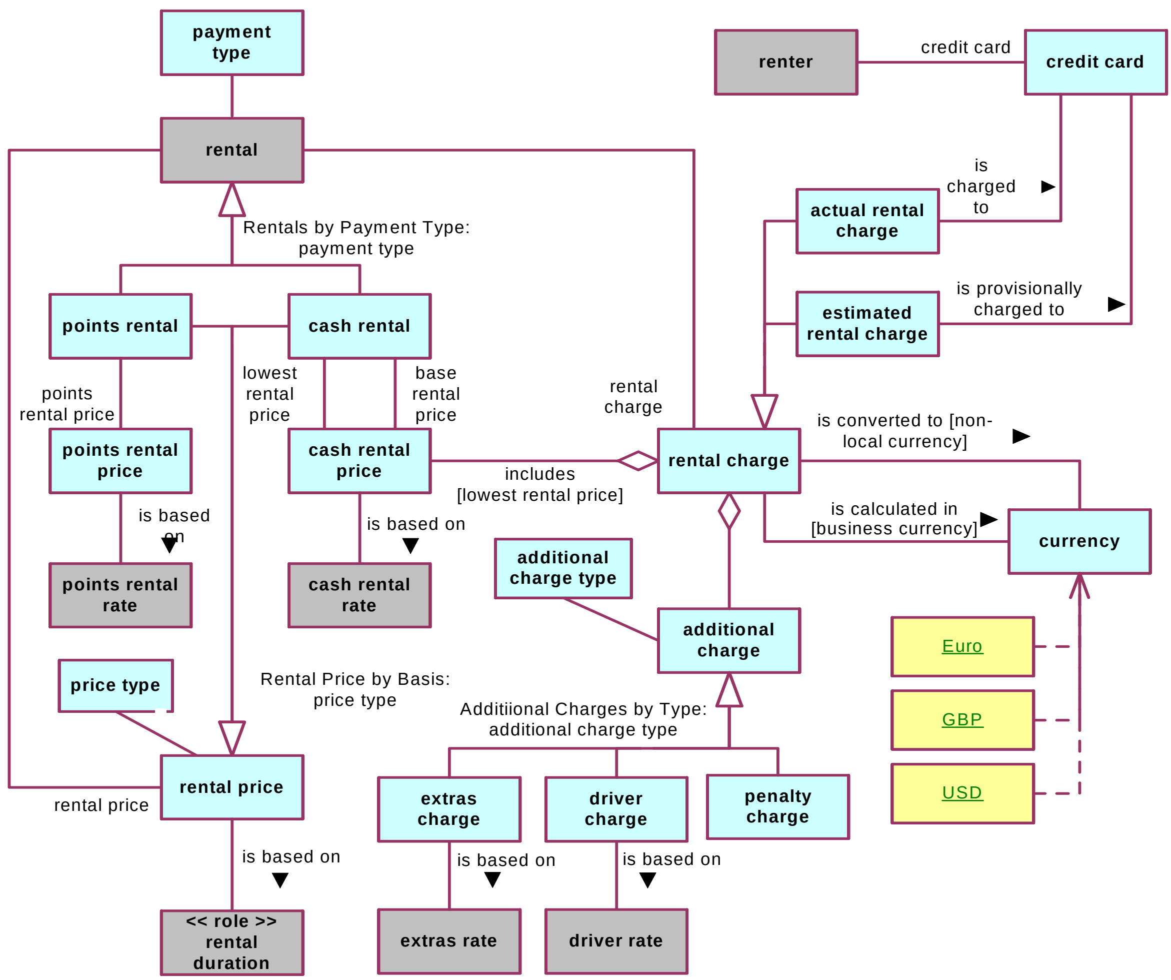


Figure E.9 - Rental Pricing

actual rental charge

Definition: rental charge that is calculated at end of rental

additional charge

Definition: charge *included in* rental charge in addition to lowest rental price

additional charge type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'additional charge' and that *classifies* an additional charge based on why it was incurred - option selected by the renter, additional driver, or penalty charge

additional charge *is included in* rental charge

Additional Charges by Charge Type

Definition: segmentation that *is for* the concept 'additional charge' and *subdivides* additional charges using additional charge type

Necessity: Additional Charges by Basis *contains* the categories 'extras charge' and 'driver charge' and 'penalty charge.'

base rental price

Concept Type: role

Definition: price charged for the use of the rented car of a rental, before any additional charges are added

Description: The base rental price is the sum of the rental rates for the requested car model for the RTUs (rental time units) that make up the rental duration. The base rental price can be calculated in money or loyalty club points.

The rental duration is broken down into integral numbers of RTUs, working from the largest RTU towards the smallest (see example).

Necessity: If the rental duration is not for an exact number of days, the final part-day is charged as a Rental Day.

Note: Hourly tariff lines are used only for calculating late charges.

Example: A 10-day rental is broken down into four rental time units: 1 x week + 3 x 1-day. The base rental price is the sum of the prices for the four RTUs.

base rental price *is based on* rental duration

business currency

Concept Type: role

Definition: currency in which EU-Rent undertakes financial transactions

Description: currency *of* an operating country.

Issue 11291 Revise text

---

cash rental

Concept Type: payment basis

Definition: rental that *is charged in money*

Necessity: The concept cash rental *is included in* Rentals by Payment Type.

Issue 9874 Add text

Issue 11291 Revise text

---

cash rental *has* base rental price

cash rental price

Concept Type: price type

Definition: base rental price that *is in money*

Necessity: The concept cash rental price *is included in* Rental Prices by Basis.

cash rental price *is based on* cash rental rate

Necessity: Each cash rental price *is based on* exactly one cash rental rate.

cash rental *honors* lowest rental price

Necessity: Each cash rental *honors* exactly one lowest rental price.

Necessity: The lowest rental price *of* a rental *is honored after* the booking date/time *of* the booking that *establishes* the rental.

Necessity: The lowest rental price *of* a rental *is honored before* the actual return date/time *of* the rental.

charge

Source: MWU 5b1 [“charge”]

credit card

Dictionary Basis: MWU, 1: a small card (as one issued by hotels, restaurants, stores, or petroleum companies) authorizing the person or company named or its agent to charge goods or services

currency

Source: MWU 2a [“currency”]

Note: Has predefined population (see below)

Issue 11291 Revise text

---

driver charge

Concept Type: additional charge type

Definition: additional charge that *is for* additional drivers *authorized for* a rental

Necessity: The concept drivers charge *is included in* Additional Charges by Basis

driver charge *is based on* driver rate

estimated rental charge

Definition: rental charge estimated at start of rental

estimated rental charge *is provisionally charged to* credit card

extras charge

Concept Type: additional charge type

Definition: additional charge that *is for* optional extra

Necessity: The concept extras charge *is included in* Additional Charges by Basis

extras charge *is based on* extras rate

lowest rental price

Concept Type: role

Definition: cash rental price that *is* most favorable to the renter *of* a cash rental

Description: Between the booking date/time of a rental and its actual return date/time, pricing changes (e.g., tariff changes, discounts, promotions) may occur.

The lowest rental price is the most favorable price for the renter that results from any such changes.

Honoring the lowest rental price applies only while the car group and duration of the rental remain unchanged.

Necessity:	A cash rental price *of* a rental that is calculated because of EU-Rent price changes and that *is less than* the lowest rental price *of* the rental replaces the lowest rental price *of* the rental.
Necessity:	A cash rental price *of* a rental that is calculated because of changes to the car group or rental duration *of* the rental replaces the lowest rental price *of* the rental.
Necessity:	The lowest rental price *of* a rental *is* not replaced *after* the actual return date/time *of* the rental.

lowest rental price *is included in* rental charge

non-local currency

Concept Type:	role
Definition:	currency that *is* not the currency *of* a rental

payment type

Concept Type:	categorization type
Definition:	concept that *specializes* the concept 'rental' and that *classifies* a rental based on whether it is paid for by credit card or loyalty club points

penalty charge

Concept Type:	additional charge type
Definition:	additional charge that *is for non-compliance with the terms of* a rental
Necessity:	The concept penalty charge *is included in* Additional Charges by Basis

points rental

Concept Type:	payment basis
Definition:	rental that is charged in loyalty club points
Necessity:	Each points rental *has* a points rental price.
Necessity:	The renter *of* each points rental *is* a club member.
Necessity:	The concept points rental *is included in* Rentals by Payment Type.

points rental price

Concept Type:	price type
Definition:	base rental price that is in loyalty club points
Necessity:	The concept points rental price *is included in* Rental Prices by Basis.

points rental price *is based on* points rental rate

price

Source:	MWU (1) ["price"]

price type

Concept Type:	categorization type

Definition: concept that *specializes* the concept 'base rental price' and that *classifies* a base rental price based on whether it is calculated in money or loyalty club points

rental charge

Concept Type: role

Definition: charge that is the total amount estimated or charged for a rental

Issue 9450 Revise text

rental *has* business currency

Necessity: A rental *has* a business currency if and only if the business currency *is* the currency *of* the operating country *of* the operating company that *includes* the local area that *includes* the pick-up branch *of* the rental.

Issue 9449 Add text

rental *has* rental charge

rental charge *is calculated in* business currency

Necessity: Each rental charge *of* each rental *is calculated in* the business currency *of* the rental.

rental charge *is converted to* non-local currency

Description: If a renter requests it, the rental charge for a rental can be shown on the contract and/or the invoice in a currency other than the currency in which it is calculated. This is done by converting the rental charge to the non-local currency.

Rental Prices by Basis

Definition: segmentation that *is for* the concept 'base rental price' and *subdivides* base rental prices based on price type

Necessity: Rentals by Payment Type *contains* the categories 'cash rental price' and 'points rental price.'

Rentals by Payment Type

Definition: segmentation that *is for* the concept 'rental' and *subdivides* rentals based on payment basis

Necessity: Rentals by Payment Type *contains* the categories 'cash rental' and 'points rental.'

renter *has* credit card

E.2.2.1.7.1 Pre-defined Population: currency

Euro

Concept Type: individual concept

General Concept: currency

Synonym: EUR

GBP

Concept Type: individual concept

General Concept: currency
Synonym: British Pound

USD
Concept Type: individual concept
General Concept: currency
Synonym: United States Dollar

E.2.2.1.8 Tariff

To keep the EU-Rent case study to a manageable size, the relationship of tariff to operating country has been greatly simplified. In reality there would be a standard tariff structure, replicated for each operating country and each populated with a different set of values. This is a data design issue, and not much is lost from the illustration of concepts and vocabulary by omitting it.

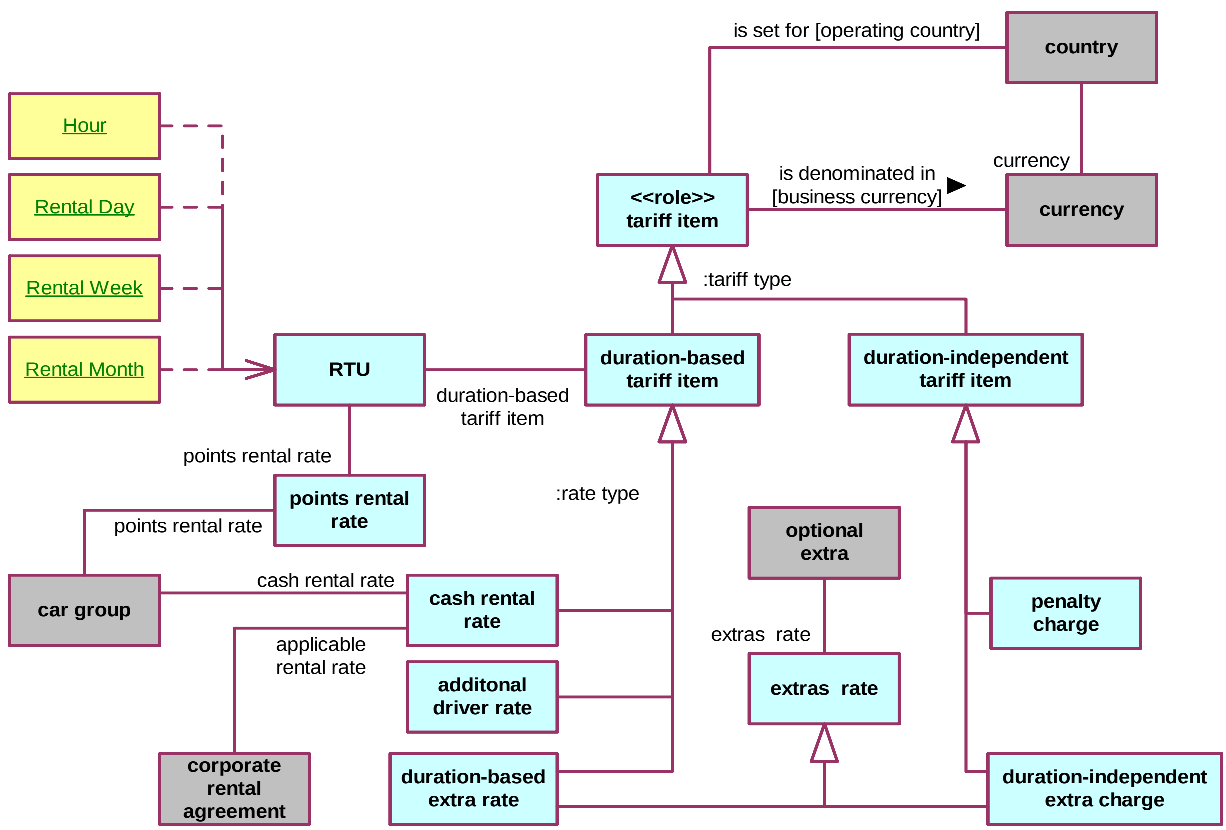


Figure E.10- Rental Tariffs

additional driver rate

Concept Type: rate type
Definition: duration-based tariff item that is for each additional driver *of* a rental

applicable rental rate

Concept Type: role

Definition: cash rental rate that is *applicable to* a *corporate rental agreement*

cash rental rate

Concept Type: rate type

Definition: duration-based tariff item that *is for rental of a car of* a given car group

cash rental rate *is for* car group

corporate rental agreement *has* applicable rental rate

Issue 9449 Add text

---

country *has* currency

duration-based extra rate

Concept Type: rate type

Definition: duration-based tariff item that *is for* an optional extra

duration-based tariff item

Concept Type: tariff type

Definition: tariff item that is charged to a rental per RTU in the duration of the rental

Example: daily cash rental rate for car group, daily cost of child seat.

duration-independent extra charge

Definition: duration-independent tariff item that *is for* an optional extra

duration-independent tariff item

Concept Type: tariff type

Definition: tariff item that is the basis for a charge to a rental regardless of the duration of the rental

Example: charge for fuel consumed; charge for one-way rental

Note: The tariff item may be the basis for calculation rather than a fixed amount. For example, a charge for fuel is calculated per liter or per gallon.

extras rate

Definition: generalization *of* duration-dependent extra charge and duration-independent extra charge

extras rate *is for* optional extra

penalty charge

Definition: duration-independent tariff item that is a penalty charge for some breach of the conditions of a rental

points rental rate

Concept Type: role

Definition: number that represents the loyalty club points charged per RTU to rent a car of a given car group

points rental rate *is for* car group

points rental rate *is for* RTU

rate type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'duration-based tariff item' and that *classifies* a duration-based tariff item based on what type of service is being charged for

rental time unit

See: RTU

RTU

Definition: time unit that is an atomic (integer) unit of time for which a car can be rented

Synonym: rental time unit

Dictionary Basis: CRISG ["RTU"]

Note: Has pre-defined population - see below.

RTU *has* duration-based tariff item

RTU *has* points rental rate

Synonymous Form: points rental rate *is for* RTU

tariff item

Concept Type: role

Definition: number that represents the price in some business currency of some element of a rental

Source: MWU 2b ["tariff"]

Example: weekly rate for a car of a given car group; cost of additional insurance; penalty charge for drop-off at location other than the return branch

Note: This entry is informally defined in order to limit the case study size. In a 'real' SBVR model, tariff items would have validity periods, and would include special offers that would override standard rates for limited periods.

tariff item *is denominated in* business currency

tariff item *is set for* operating country

tariff type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'tariff item' and that *classifies* a tariff item based on whether it is a per-RTU change or not

Predefined Population - **RTU**

Hour

Definition: "the 24th part of a mean solar day **:** 60 minutes of mean solar time"

General Concept:	RTU
Source:	MWU 2b ["hour"]

Rental Day

Definition:	24-hour period, starting at actual pick-up time of rental
Note:	Not the scheduled pick-up date/time
General Concept:	RTU
Example:	Day beginning at 3:45 p.m.

Rental Week

Definition:	7 consecutive rental days
General Concept:	RTU

Rental Month

Definition:	28 consecutive rental days
General Concept:	RTU

E.2.2.1.9 Rental Problems

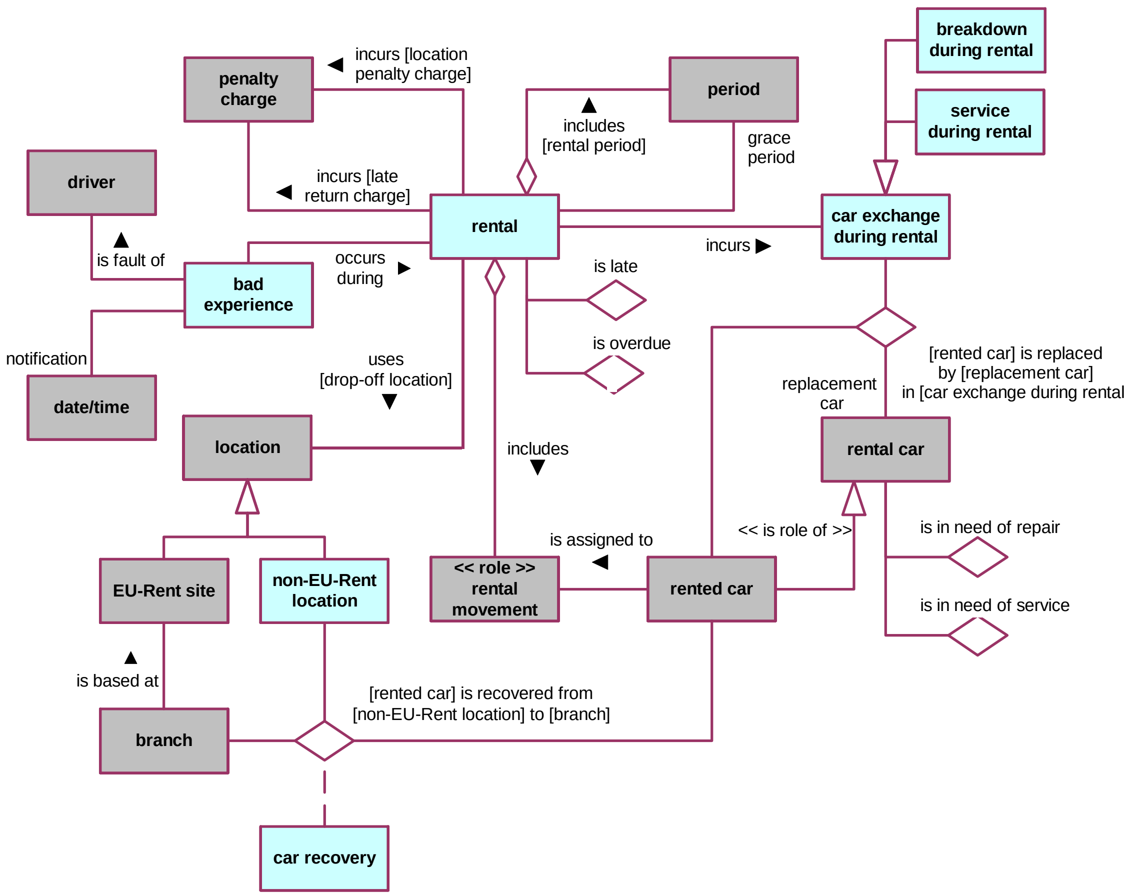


Figure E.11- Rental Problems

actual return branch

See: drop-off branch

Issue 9449 Revise text / add text
Issue 11291 Replace text

bad experience

Definition:	undesirable occurrence during a rental that is the fault of one of the drivers
Example:	Speeding offence, unpaid parking fine, damage to car caused by careless driving.
Note:	This entry is informally defined in order to limit the case study size.

bad experience *has* notification date/time

bad experience *is fault of* driver

bad experience *occurs during* rental

breakdown during rental

Definition: car exchange during rental *of* a rented car that has operational problems

car exchange during rental

Definition: situation where the rented car of a rental cannot be used for the remainder of the rental duration

car recovery

Definition: actuality that a given rented car *is recovered from* a given non-EU-Rent site *to* a given branch

charge

Source: CRISG ["charge"]

drop-off branch

Concept Type: role

Definition: branch to which a rented car is actually returned

Note: A car may be returned to a branch other than the one agreed in the rental. EU-Rent will accept the car, but will charge a location penalty.

Synonym: actual return branch

drop-off location

Concept Type: role

Definition: location where the rented car *of* a rental is dropped off

grace period

Concept Type: role

Definition: period that *has* start date/time that *is* scheduled end date/time *of* rental period and end date/time that *is the earlier of* (scheduled end date/time *of* rental period plus one hour, closing time *of* return branch *of* rental)

late return charge

Concept Type: role

Definition: penalty charge that *is made for a* rental *that is late*

Description: The late charge is calculated using the hourly tariff for the car group to which the car belongs, for durations of up to 5 hours after the end of the grace period. Part-hours are rounded up. The daily tariff is used for durations between 5 and 24 hours.
If, after the end of the grace period, the renter contacts EU-Rent to extend the rental, the late return charge is calculated from the end of the grace period to the date/time when the rental extension is agreed.

Note: If the car is not returned within 48 hours after the end of the grace period, and the renter has not contacted EU-Rent to extend the rental, the insurance lapses and EU-Rent will report the car to the police as stolen and uninsured.

location penalty charge

Concept Type: role

Definition: penalty charge that *is made for* each rental that *has* a drop-off location that *is* not the EU-Rent site *of the* return branch *of the* rental

Description: The location penalty charge is calculated in three parts: a fixed penalty; cost of retrieving the car if the location is a non-EU-Rent site (e.g., an airport car park); cost of moving the car to the return branch specified in the rental. Car movement costs are taken from a standard scale based on the distance between branches and per-mile (or per-kilometer) costs for car groups.

non-EU-Rent location

Concept Type: role

Definition: location that *is* not the location *of* a rental organization unit

notification date/time

Concept Type: role

Definition: date/time at which something is notified to EU-Rent

Issue 11291 Revise verb form

rental *is late*

Concept Type: rental state

Definition: rental *having* a rented car that *is in possession of* the renter and the end date/time *of* the grace period *of* the rental *is in the past* and *is less than 24 hours in the past*

rental *is overdue*

Concept Type: rental state

Definition: rental *having* a rented car that *is in possession of* the renter and the end date/time *of* the grace period *of* the rental *is more than 24 hours in the past*

rental car *is in need of repair*

Concept Type: rental state

Definition: rental car *having* damage or breakdown that renders it unusable for rental

rental car *is in need of service*

Concept Type: rental state

Definition: rental car *having* service reading *that is* at least 5000 miles.

rental car state

Concept Type: characteristic type

Issue 9450 Revise text

rental *has* drop-off branch

Necessity: A rented car *has* a drop-off branch if and only if the drop-off branch *is* the branch that *is based at* the EU-Rent site that *is* the drop-off location *of* the rental.

rental *incurs* car exchange during rental

rental *incurs* late return charge

rental *incurs* location penalty charge

rental *has* grace period

Note: late return charges are not applied until after the grace period.

rental ~~*uses*~~ *has* drop-off location

rented car *is recovered from* non-EU-Rent location *to* branch

rented car *is replaced by* replacement car *in* car exchange during rental

replacement car

Concept Type: role

Definition: rental car *in* a car exchange during rental that is used after the exchange

service exchange

Definition: car exchange during rental *of* a rented car that *is due for service*

unauthorized drop-off location

Definition: location that *is used to drop off* the rented car *of* a rental and that *is* not the EU-Rent site *of* the return branch *of* the rental

E.2.2.1.10 Rental Cars

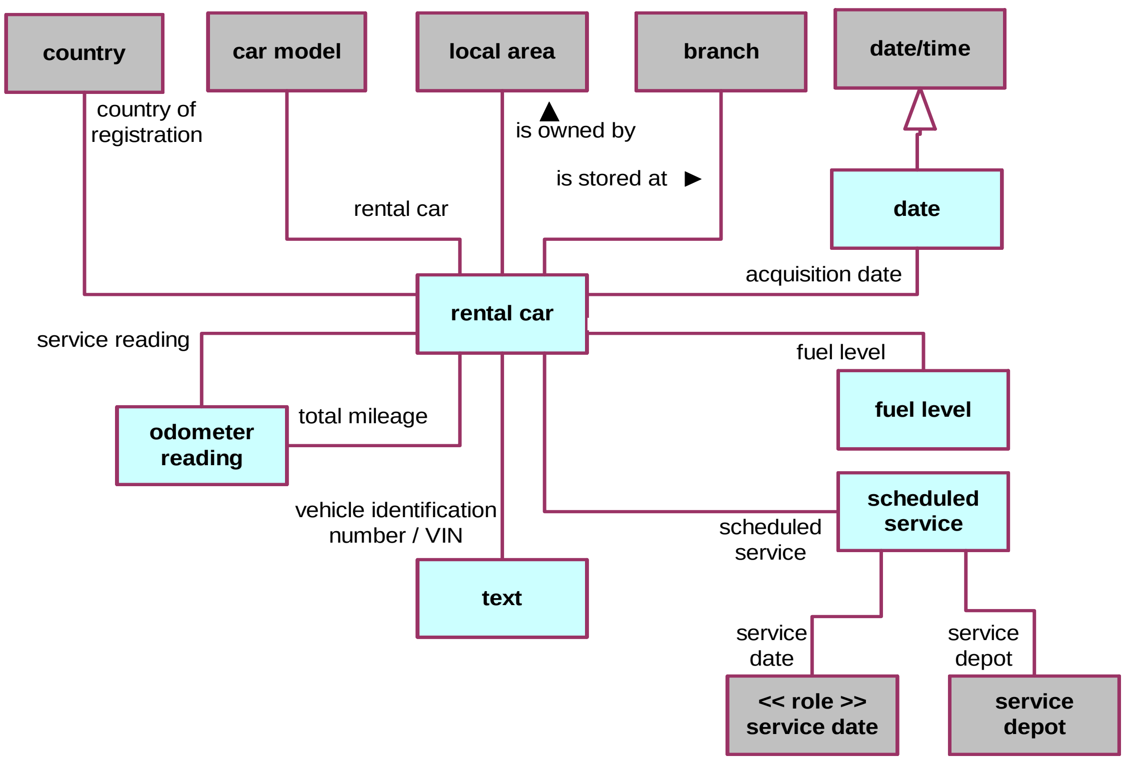


Figure E.12 - Rental Cars

acquisition date

Concept Type: role

Definition: date on which EU-Rent took ownership of some thing

car

See: rental car

country of registration

Concept Type: role

Definition: country in which something is registered with the relevant authorities

fuel level

Definition: full or 7/8 or 3/4 or 5/8 or 1/2 or 3/8 or 1/4 or 1/8 or empty

Source: CRISG ["fuel level"]

odometer reading

Concept Type: role

General Concept: number

Source: CRISG ["odometer reading"]

rental car

Source: MWU (1/1d) ["car"], CRISG ("rental car")

Definition: vehicle owned by EU-Rent and rented to its customers

Synonym: car

rental car *has* acquisition date

Concept Type: is-property-of fact type

Synonymous Form: rental car *is acquired on* acquisition date

rental car *has* country of registration

Concept Type: is-property-of fact type

rental car *has* odometer reading

Concept Type: is-property-of fact type

Issue 9874 Add text
Issue 11291 Add entry

---

rental car *has* scheduled service

rental car *has* service reading

Concept Type: is-property-of fact type

rental car *has* vehicle identification number

Concept Type: is-property-of fact type

Necessity: Each rental car *has* exactly one vehicle identification number.

rental car *has* fuel level

Definition: is-property-of fact type

rental car *is of* car model

Concept Type: is-property-of fact type

Necessity: Each rental car *is of* exactly one car model.

rental car *is of* car group

Concept Type: associative fact type

Necessity: A rental car *is of* a car group if and only if the rental car *is of* some car model that *is included in* the car group.

Issue 9467 Revise text / remove text

local area *owns* rental car

Necessity: Each rental car *is owned by* exactly one local area.

~~Synonymous Form:~~ ~~local area *owns* rental car~~

rental car *is stored at* branch

Necessity: Each rental car *is stored at* at most one branch.

scheduled service

Definition: maintenance service *for* a rental car that *is scheduled at* a (EU-Rent) service depot

service date

General Concept: role

Definition: date of scheduled service

scheduled service *has* service date

scheduled service *has* service depot

service reading

Concept Type: role

Definition: odometer reading since the car was last serviced

Note: When the service reading reaches 5000 miles (8000 km), the car will be scheduled for service.

vehicle identification number

Concept Type: role

Definition: text that is the unique identifier of a particular vehicle

Synonym: VIN

VIN

Synonym: vehicle identification number

E.2.2.1.11 Customers

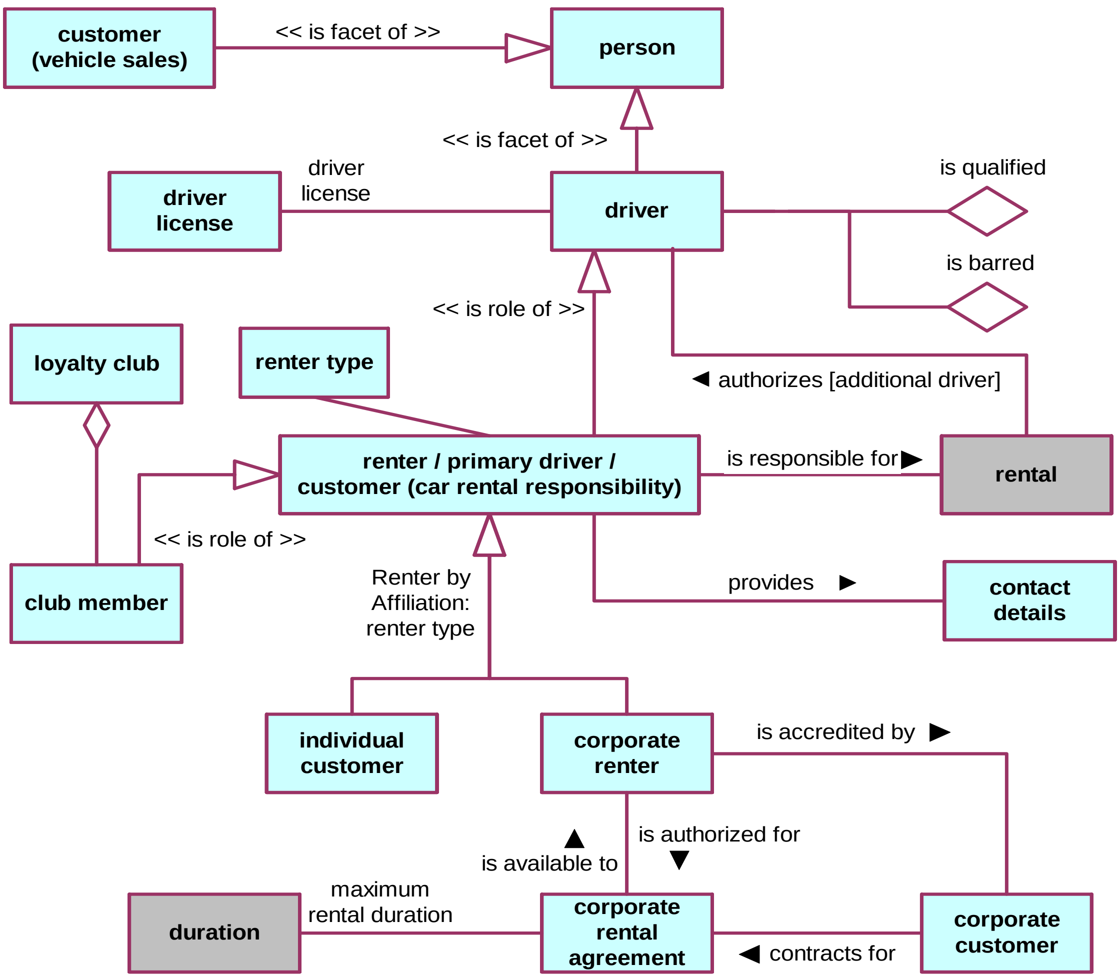


Figure E.13 - Customers

additional driver

Source: CRISG ["additional driver"]

Concept Type: role

Definition: driver *of* a rental who *is* not the renter *of* the rental

additional driver *is authorized in* rental

Necessity: Each rental *authorizes* at most 3 additional drivers.

Synonymous Form: rental *authorizes* additional driver

Issue 9941 Revise text

Car Rental Responsibility

Source: CRISG ["rental responsibility"]

General Concept: subject field

club member

Definition: renter who has joined EU-Rent's loyalty club

contact details

Definition: address, telephone number, and (if available) email address

corporate customer

Dictionary Basis: relating or belonging to a corporation MWU ["corporate"]

Dictionary Basis: person or company who buys goods and services MWU ["customer"]

corporate customer *contracts for* corporate rental agreement

corporate rental agreement

Definition: contract under which a corporate renter can rent a car at a negotiated set of rates

Note: Each rental under a corporate rental contract is an individual contract, and the corporate renter is personally responsible for it.

Note: This entry is informally defined in order to limit the case study size.

corporate rental agreement *has* maximum rental duration

Issue 11291 Revise text

corporate renter

Concept Type: renter type

Definition: renter who is a representative of a corporate customer, accredited to rent cars under the terms of its corporate rental agreement, who has booked at least one rental

Necessity: Each corporate renter *is* a person who *is accredited by* a corporate customer and who *is responsible for* at least one rental.

Necessity: The concept corporate renter *is included in* Renters by Affiliation

corporate renter *is accredited by* corporate customer

corporate renter *is authorized for* corporate rental agreement

Synonymous Form: corporate rental agreement *is available to* corporate renter

Necessity: Each corporate renter *is authorized for* at least one corporate rental agreement.

customer

Subject Field: Car Rental Responsibility

Concept Type: role

See: renter

customer

Subject Field: Vehicle Sales

Concept Type: facet

Definition: person who purchases a rental car from EU-Rent at the end of its rental life

---

Issue 9449 Revise text
Issue 11291 Revise verb form

---

driver

Concept Type: facet

Definition: person who has been identified as one who can drive the rented car *of* a rental

driver *is barred*

Concept Type: driver state

Definition: driver *being prohibited from renting a car from EU-Rent*

Note: A barred driver is a person known to EU-Rent as a driver (either a renter or an additional driver), who has at least 3 bad experiences.

driver *is qualified*

Concept Type: driver state

Definition: the driver is over 21 years old and *has* a valid driver license and is not under any pending legal action that could adversely affect his driver's license or insurability.

driver *has* driver license

driver license

Source: CRISG ["driver license"]

driver state

Concept Type: characteristic type

individual customer

Concept Type: renter type

Definition: renter who is not a corporate renter, who meets at least one of the following criteria: has completed a rental within the last 5 years; has a rental currently in progress; has made a rental booking

Necessity: Each individual customer *is* a given person who *is* not a corporate renter and who *is responsible for* at least one rental that *is* a Reserved Rental or an Assigned Rental or an Open Rental or a Returned Rental that *has* an end date that *is less than* 5 years *earlier than* the current day date.

Necessity: The concept individual customer *is included in* Renters by Affiliation.

loyalty club

Definition: EU-Rent's incentive scheme for its frequent renters

Note: A customer who joins the loyalty club accumulates points that s/he can use to pay for a rental.

loyalty club *includes* club member

Concept Type: partitive fact type

maximum rental duration

Concept Type: role

Definition: duration that is the upper limit for rental duration *of* each rental made under the terms of a corporate rental agreement

person

Source: MWU (1a) ["person"]

primary driver

See: renter

Issue 9450 Add text

---

rental *has* driver

Necessity: A rental *has* a driver if and only if the driver *is* the renter that *is responsible for* the rental or an additional driver that *is authorized in* the rental.

Issue 9948 Revise text

---

renter

Source: CRISG ["renter"]

Concept Type: role

Definition: ~~person~~ driver contractually responsible for a rental

Synonym: customer (car rental responsibility)

Synonym: primary driver

Issue 9449 Add text
Issue 11291 Revise text

---

renter *is responsible for* rental

Concept Type: associative fact type

Synonymous Form: rental *has* renter

Necessity: Each rental *has* exactly one renter.

Necessity: The renter *of* a rental is not *changed*.

Note: If the renter wishes to change the rental to a different renter, EU-Rent regards it as a cancellation and a new rental.

renter provides current contact details

renter type

Concept Type: categorization type

Definition: concept that *specializes* the concept 'renter' and that *classifies* a rental based on whether it is an individual customer or a corporate renter

Renters by Affiliation

Definition: segmentation that *is for* the concept 'renter' and *subdivides* renters based on renter type

Necessity: Renters by Affiliation *contains* the categories 'individual customer' and 'corporate renter.'

Issue 9941 Revise text

Vehicle Sales

Source: CRISG ["car sales"]

General Concept: subject field

E.2.2.2 EU-Rent Guidance expressed using the EU-Rent English Vocabulary

Issue 9475/9945 Replace text
Issue 9955 Remove text

This subclause presents elements of guidance (business rules ~~admonitions,~~ and ~~affirmations~~ advices) that accompany the "EU-Rent English Vocabulary" -- as described in Annex C ('C.4 Specifying a Rule Set').

Note - The guidance in this subclause is expressed in the EU-Rent English Vocabulary; a working subset of this is provided in the preceding subclause. If the statements are difficult to understand at face value – e.g., seem ambiguous, or don't quite fit everyday understanding of the words used – it is important to check the definitions ~~in the vocabulary~~ before challenging the guidance statements.

Many of the guidance statements are supported by descriptions, which reflect EU-Rent users' informal statements of the guidance.

Issue 9449 Revise text

The examples in this subclause are generally presented in the form "It is obligatory that …," "It is necessary that …," etc. This emphasizes the application of the modal claim (obligation, necessity, etc.) to the underlying fact type, but sometimes provides a cumbersome representation. SBVR Structured English (see Annexes H and C) also supports more direct representation. For example, the operative business rule:

It is obligatory that each driver *of* a rental *is qualified*.

can be represented as

Each driver *of* a rental must *be qualified.*

For a treatment of these examples in RuleSpeak®, a widely-used business rule notation, see Annex F.

E.2.2.2.1 Introduction

<EU-Rent English Vocabulary Rules>

Vocabulary: EU-Rent English Vocabulary

<EU-Rent English Vocabulary Levels of Enforcement>

Level of enforcement is a categorization scheme for business rules defined (or adopted) by the organization that owns the rules. EU-Rent's categories are listed below.

Enforcement Level: strict

Definition: strictly enforced: if the rule is violated, the sanction or other consequences always ensue.

Enforcement Level: deferred

Definition: deferred enforcement: strictly enforced, but enforcement may be delayed — e.g., waiting for resource with required skills.

Enforcement Level: pre-authorized

Definition: pre-authorized override: enforced, but exceptions allowed, with prior approval for actors with before-the-fact override authorization.

Enforcement Level: post-justified

Definition: post-justified override: if not approved after the fact, the sanction or other consequences will ensue.

Enforcement Level: override

Definition: override with explanation: comment must be provided when the violation occurs.

Enforcement Level: guideline

Definition: guideline: suggested, but not enforced.

E.2.2.2.2 Rule Set -- Rental Rules

Issue 9449 Revise text

It is necessary that each rental *has* exactly one requested car group.

Guidance Type: structural business rule

Description: The renter may request a change of car group up to pick-up time, but a car group must always be specified

Supporting fact type: rental *has* requested car group

It is necessary that each rental *includes* exactly one rental period.

Guidance Type: structural business rule

Description: The renter may request a change to the start and/or end of the rental period, or cause a de-facto change by late return of the car, but a rental period must always be specified

Supporting fact type: rental *has* rental period

It is necessary that each rental *has* exactly one return branch.

Guidance Type:	structural business rule
Description:	The renter may request a change to the return branch, or cause a de-facto change by return of the car to an unauthorized branch, but a return branch must always be specified
Supporting fact type:	rental *has* return branch

It is necessary that the scheduled pick-up date/time *of* each advance rental *is after* the booking date/time *of* the rental booking that *establishes* the advance rental.

Guidance Type:	structural business rule
Description:	When a rental reservation is made (establishing an advance rental) the rental scheduled pick-up date/time must be later than the actual date/time of reservation
Supporting fact types:	rental booking *has* booking date/time rental booking *establishes* advance rental rental *has* scheduled pick-up date/time date/time_1 *is after* date/time_2
Related facts:	the noun concept 'cash rental' *is* a category *of* the noun concept 'rental' the noun concept 'advance rental' *is* a category *of* the noun concept 'rental'

E.2.2.2.3 Rule Set -- Charging / Billing / Payment Rules

Issue 9449 Revise text
Issue 9948 Revise text

It is permitted that a rental *is open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.

Guidance Type:	operative business rule
Description:	While a renter has possession of a car, there is a provisional charge to EU-Rent against his credit card. This will be replaced by an actual charge at the end of the rental
Enforcement Level:	Strict
Supporting fact types:	rental *has* rental charge estimated rental charge *is provisionally charged to* credit card renter *has* credit card rental *has* driver rental *is open* renter *is responsible for* rental
Related facts:	~~'*being open*' *is a characteristic of* the noun concept 'rental'~~ The noun concept 'estimated rental charge' *is* a category *of* the noun concept 'rental charge.' The noun concept 'renter' *is* a role that *ranges over* the noun concept ~~person~~ 'driver.' The noun concept 'driver' *is* a facet *of* the noun concept 'person.'

It is necessary that the rental charge *of* each rental *is calculated in* the business currency *of* the rental.

Guidance Type:	structural business rule
Note:	This is a constraint imposed by credit card issuers.

Supporting fact types: rental *has* rental charge
rental charge *is calculated in* business currency *of* rental
rental *has* business currency

If the renter *of* a rental *requests* a price conversion then it is obligatory that the rental charge *of* the rental *is converted to* the currency *of* the price conversion.

Guidance Type: operative business rule

Description: EU-Rent will provide the customer with a bill in another currency, but the actual billing is done in the business currency, and converted.

Enforcement Level: strict

Supporting fact types: rental *has* renter
rental *has* rental charge

Related fact: a price conversion *is* the rental charge *of* a rental *denominated in* a currency *requested by* the renter

It is necessary that each cash rental *honors* the lowest rental price *of* the cash rental.

Guidance Type: structural business rule

Description: Between the booking date/time of a cash rental and its actual return date/time, pricing changes (e.g., tariff changes, discounts, promotions) may occur.
The lowest rental price is the most favorable price for the renter that results from any such changes.
Honoring the lowest rental price applies only while the car group and duration of the rental remain unchanged.

The structural business rule above can be elaborated as three detailed structural business rules.

It is necessary that a cash rental price *for* a cash rental that is calculated because of EU-Rent price changes and that *is less than* the lowest rental price *honored by* the rental replaces the lowest rental price *honored by* the rental.

It is necessary that a cash rental price *for* a cash rental that is calculated because of changes to the requested car group or rental duration *of* a rental replaces the lowest rental price *honored by* the rental.

It is necessary that the lowest rental price *honored by* a rental *is* not replaced *after* the actual return date/time *of* the rental.

Issue 9874 Revise text
Issue 9449 Revise text

There is no need for a structural business rule that the lowest rental price is not replaced before the booking date, because the rental does not exist before that date.

Supporting fact types: cash rental *honors* lowest rental price
cash rental *has* base rental price
rental *has* actual return date/time
rental *has* requested car group
rental *has* rental duration

state of affairs *occurs after* date/time

Related facts: the noun concept 'cash rental' *is* a category *of* the noun concept 'rental'

the noun concept 'lowest rental price' *is* a role that *ranges over* the noun concept 'cash rental price'

the noun concept 'cash rental price' *is* a category *of* the noun concept 'base rental price'

E.2.2.2.4 Rule Set -- Driver Rules

Issue 9459 Change "Synonymous Form' to Synonymous Statement"
Issue 9449 Revise text
Issue 11291 Revise text

It is permitted that a rental *is* open only if each driver *of* the rental *is* not a barred driver.

Synonymous Statement: It is prohibited that a rental *is open* if a driver *of* the rental *is* a barred driver.

Guidance Type: operative business rule

Description: While a rented car is in possession of the renter, no driver for the rental can be a barred driver.

Enforcement Level: pre-authorized

Supporting fact types: rental *has* primary driver

rental *has* additional driver

Related facts: '*being open*' *is* a characteristic *of* the noun concept 'rental'

'*being barred*' *is* a characteristic *of* the noun concept 'driver'

the noun concept 'primary driver' *is* a role that *ranges over* the noun concept 'driver'

the noun concept 'additional driver' *is* a role that *ranges over* the noun concept 'driver'

It is obligatory that each driver *of* a rental *is qualified*.

Guidance Type: operative business rule

Description: To be accepted as a driver on a rental, a person must comply with EU-Rent's definition of "driver is qualified."

Enforcement Level: Strict

Supporting fact types: rental *has* primary driver

rental *has* additional driver

driver *is qualified*

Related facts: ~~'*being qualified*' *is* a characteristic *of* the concept 'driver'~~

the noun concept 'primary driver' *is* a role that *ranges over* the noun concept 'driver'

the noun concept 'additional driver' *is* a role that *ranges over* the noun concept 'driver'

E.2.2.2.5 Rule Set -- Pick-up / Return Rules

This subclause illustrates a trade-off of a larger body of shared concepts, and corresponding vocabulary, against simpler formulation of business rules.

The business rules here could have been stated more elaborately; e.g., one of the examples below is:

It is obligatory that the country *of* the return branch *of* each international inward rental *is* the country of registration *of* the rented car *of* the rental.

It could have been stated as

"If the country *of* the pick-up branch *of* a rental *is* not the country of registration *of* the rented car *of* the rental then it is obligatory that the country *of* the return branch *of* the rental *is* the country of registration *of* the rented car."

Defining categories of rental, as used below, simplifies the expression of rules at the expense of additional concepts and larger vocabulary to be managed.

This kind of trade-off is a business choice of the semantic community.

It is obligatory that *at* the actual return date/time *of* each in-country rental and each international inward rental the local area *of* the return branch *of* the *rental owns* the rented car *of* the rental.

Guidance Type:	operative business rule
Description:	When a car is moved between branches in different local areas in the same country, or is returned to its country of registration after being dropped off abroad, ownership moves between local areas with it. This is so whether it is a one-way rental or a transfer made by EU-Rent.
Note:	Ideally, this would be a structural rule, defining ownership at the end of rentals, but EU-Rent cannot always control car movements as it would like to.
Enforcement Level:	pre-authorized
Supporting fact types:	rental *has* actual return date/time
	rental *has* return branch
	branch *is included in* local area
	local area *owns* rental car
	state of affairs *occurs at* date/time
Related facts:	the noun concept 'rented car' *is* a role that *ranges over* the noun concept 'rental car'
	the noun concept 'return branch' *is a* role that *ranges over* the noun concept 'branch'
	the noun concept 'in-country rental' *is* a category *of* the noun concept 'rental'
	the noun concept 'international inward rental' *is* a category *of* the noun concept 'international rental'
	the noun concept 'international rental' *is* a category *of* the noun concept 'rental'

It is obligatory that the country *of* the return branch *of* each international inward rental *is* the country of registration *of* the rented car *of* the rental.

Guidance Type:	operative business rule
Description:	When a one-way rental has dropped a car off in a different country, that car may then be used for only one kind of rental – a one-way rental back to its country of registration.
Note:	If a one-way rental back to country of registration does not occur within a short time, the branch manager will have a EU-Rent employee transfer the car.
Enforcement Level:	pre-authorized
Supporting fact types:	branch *has* country
	rental *has* return branch
	rental car *has* country of registration

Related facts: the noun concept 'rented car' *is* a role *of* the concept 'rental car'

the noun concept 'international inward rental' *is* a category *of* the noun concept 'rental'

the noun concept 'return branch' *is* a role that *ranges over* the noun concept 'branch'

the noun concept 'country of registration' *is* a role that *ranges over* the noun concept 'country'

It is necessary that if a rental *is open* and the rental *is* not an international inward rental then the rented car *of* the rental *is owned by* the local area *of the* pick-up branch *of* the rental.

Guidance Type: structural business rule

Note: This ensures that the local area that owned the car at the start of a rental retains responsibility for it until it is dropped off at an EU-Rent branch.
It also ensures that a car's ownership is retained within its country of registration.

Supporting fact types for the three business rules above:

rental *has* pick-up branch

local area *includes* branch

rental car *is owned by* local area

Related facts: the noun concept 'rented car' *is* a role that *ranges over* the noun concept 'rental car'

'international inward rental' *is* a category *of* 'international rental'

'international rental' *is* a category *of* 'rental'

'being open' *is* a characteristic *of* 'rental'

'pick-up branch' *is* a role *of* 'branch'

'return branch' *is* a role *of* 'branch'

Issue 11291 Revise text

If the actual return date/time *of* a rental *is after* the end date/time *of* the grace period *of* the rental then it is obligatory that the rental *incurs* a late return charge.

Guidance Type: operative business rule

Note: The grace period of a rental ends one hour after the rental's scheduled return date/time or at close of business of the return branch, whichever is earlier.

Enforcement Level: Strict

Supporting fact types: rental *has* ~~start~~ actual return date/time

rental *has* grace period

period *has* end date/time

date/time$_1$ *is after* date/time$_2$

rental *incurs* late return charge

Related facts: the noun concept 'actual return date/time' *is* a role that *ranges over* the noun concept 'date/time'

the noun concept 'grace period' *is* a role that *ranges over* the noun concept 'period'

the noun concept 'end date/time' *is* a role that *ranges over* the noun concept 'date/time'

~~the noun concept 'actual pick-up date/time' *is* a role that *ranges over* the noun concept 'date/time'~~

Issue 9449 Revise text

If the drop-off location of a rental is not the EU-Rent site that is base for the return branch of the rental then it is obligatory that the rental incurs a location penalty charge.

Guidance Type: operative business rule

Description: If a rented car is returned to a location that is not the specified return branch of the rental, that branch will accept the car but a location penalty charge will be applied to the rental.

Enforcement Level: Strict

Supporting fact types: rental has drop-off location

rental has return branch

branch is located at EU-Rent site

rental incurs location penalty charge

EU-Rent site is base for rental organization unit

Related facts: The noun concept 'return branch' is a role that ranges over the noun concept 'branch.'

The noun concept 'branch' is a category of the noun concept 'rental organization unit.'

The noun concept 'EU-Rent site' is a role that ranges over the noun concept 'location.'

The noun concept 'drop-off location' is a role that ranges over the noun concept 'location.'

Issue 9459 Change "Synonymous Form" to "Synonymous Statement"

If a rental is assigned then it is obligatory that the rented car of the rental is stored at the pick-up branch of the rental.

Synonymous Statement: It is prohibited that the rented car of an assigned rental is not stored at the pick-up branch of the rental.

Guidance Type: operative business rule

Description: A rental car must physically be at the pick-up branch when it is assigned to a rental.

Note: This is an example of a rule created to ensure that real-world influences do not cause problems in EU-Rent's business. In this case, EU-Rent knows that sometimes cars are not brought to branches when they are supposed to be, so it insists that cars assigned to rentals are physically present. It does not permit cars that are "due to be returned to this branch tomorrow" to be assigned.

After assignment to a rental, the car must stay at the branch until pick-up time.

This doesn't mean that the car can't be moved. It means that if a car is to be moved, it must be unassigned from any rental and another car assigned in its place.

Enforcement Level: Override

Supporting fact types: rental car is stored at branch

Related facts: the noun concept 'rented car' is a role that ranges over the noun concept 'rental car'

'being assigned' is a characteristic of the noun concept 'advance rental'

the noun concept 'advance rental' is a category of the noun concept 'rental'

the noun concept 'pick-up branch' is a role that ranges over the noun concept 'branch'

At the actual start date/time *of* each rental it is obligatory that the fuel level *of* the rented car *of* the rental *is* full.

Guidance Type:	operative business rule
Description:	A rented car must have a full tank of fuel at the rental pick-up time.
Note:	This is an example of a rule created to ensure that real-world influences do not cause problems in EU-Rent's business. In this case, two requirements are met. First, a car must have some fuel in it for the customer to drive it away. Second, starting fully-fuelled means that EU-Rent can easily estimate how much fuel is to be charged for at the end of the rental.
Enforcement Level:	post-justified
Supporting fact types:	rental *has* start/date time rental *has* rental car rental car *has* fuel level state of affairs *occurs at* date/time
Related facts:	the concept 'rented car' *is* a role *of* the concept 'rental car' fuel level *is* full or 7/8 or 3/4 or 5/8 or 1/2 or 3/8 or 1/4 or 1/8 or empty

E.2.2.2.6 Rule Set -- Points Rental Rules

It is necessary that the booking date/time *of* a points rental *is at least 5 days before* the scheduled start date/time *of* the rental.

Guidance Type:	structural business rule
Description:	
Supporting fact types:	rental *has* booking date/time rental *has* scheduled start date/time date/time_1 *is before* date/time_2
Related facts:	the noun concept 'points rental' *is* a category *of* the noun concept 'rental' the noun concept 'scheduled start date time' *is* a role that *ranges over* the noun concept 'date/time' the noun concept 'booking date/time' *is* a role that *ranges over* the noun concept 'date/time'

It is necessary that the renter *of* each points rental *is* a club member.

Guidance Type:	structural business rule
Note:	Only club members have points balances against which points rentals can be charged. Bookings for points rentals are not accepted from non-members.
Supporting fact type:	rental *has* renter
Related facts:	the noun concept 'points rental' *is* a category *of* the noun concept 'rental' the noun concept 'club member' *is* a role that *ranges over* the noun concept 'renter'

E.2.2.2.7 Rule Set -- Rental Period Rules

Issue 9459 Change "Synonymous Form" to "Synonymous Statement"
Issue 9448 Change formatting
Issue 9449 Revise text

It is obligatory that the start date *of* each reserved rental *is in the future*.

Synonymous Statement:	It is prohibited that the start date *of* a reserved rental *is in the past*.
Guidance Type:	operative business rule
Description:	A rental should not be booked or rescheduled with a start date/time earlier than the actual date/time of the booking or rescheduling.
Note:	On any given day, rentals that are due to be picked up that day should not be "reserved," but "assigned" - i.e., they should have cars assigned to them.
Enforcement Level:	pre-authorized
Supporting fact types:	rental *has* start date
	date/time *is in the future*
Related facts:	the noun concept 'reserved rental *is a category of* the noun concept 'rental'
	the noun concept 'start date' *is* a role that *ranges over* the noun concept 'date/time'

It is obligatory that the rental duration *of* each rental *is at most* 90 rental days.

Guidance Type:	operative business rule
Description:	EU-Rent doesn't allow rentals to be reserved for longer than 90 days or be extended beyond 90 days.
Note:	There are other legitimate ways to define what a duration is. Standards organizations, including ISO, are working on standards for measurement, including measurement of time. When there is a clear consensus on such standards, SBVR will adopt them as defaults. In the interim, individual enterprises will define for themselves consistent ways to represent measurements within their own vocabularies. EU-Rent has elected to style duration as a name denoting an instance of duration. But, being aware that other organizations might have taken a different approach to defining these kinds of measurements, EU-Rent will be watchful that, in an interchange that involves measurements, there may be things needing adjustment.
Enforcement Level:	pre-authorized
Supporting fact type	rental *has* rental duration

If $rental_1$ *is* not $rental_2$ and the renter *of* $rental_1$ *is* the renter *of* $rental_2$ then it is obligatory that the rental period *of* $rental_1$ does not overlap the rental period *of* $rental_2$.

Guidance Type:	operative business rule
Description:	A renter can have at most one open rental – i.e., can have only one rental car at a time.
Enforcement Level:	pre-authorized
Supporting fact types:	rental *has* renter
	rental *has* rental period
	$period_1$ *overlaps* $period_2$

E.2.2.2.8 Rule Set -- Servicing Rules

It is obligatory that each rental car *in need of service has* a scheduled service.

Guidance Type:	operative business rule
Description:	A rental car that has done more than 5000 miles since its last service is in need of service and has to be scheduled for service.
Note:	For countries that measure distance in kilometers, the figure is 8000
Enforcement Level:	Deferred
Supporting fact type:	rental car *has* scheduled service
Related fact:	'*being in need of service*' *is a characteristic of* 'rental'

It is obligatory that the service reading *of* a rental car *is* at most 5500 miles.

Guidance Type:	operative business rule
Description:	A car must not be run for more than 5500 miles without being serviced.
Note:	For countries that measure distance in kilometers, the figure is 8800
Enforcement Level:	pre-authorized
Supporting fact types:	rental car *has* service reading

If the rented car *of* an open rental *is in need of service* or *is in need of repair* then it is obligatory that the rental *incurs* a car exchange during rental.

Guidance Type:	operative business rule
Description:	During a rental, if the rental car's service reading exceeds 5000 miles (8000 km), the renter must take the car to a branch
Enforcement Level:	pre-authorized
Supporting fact types:	rental *has* rented car
	rental *incurs* car exchange during rental
Related facts:	the noun concept 'rented car' *is* a role that *ranges over* the noun concept 'rental car'
	'*being open*' *is* a characteristic *of* the *concept* 'rental'

E.2.2.2.9 Rule Set -- Transfer Rules

At the transfer drop-off date/time *of* a car transfer it is obligatory that the transferred car *of* the car transfer *is owned by* the local area that *includes* the transfer drop-off branch *of* the car transfer.

Guidance Type:	operative business rule
Description:	When a car is moved between branches in different local areas in the same country, ownership moves to the local area of the receiving branch.
Enforcement Level:	Strict
Supporting fact types:	car transfer *has* transfer drop-off date/time
	car transfer *has* transfer drop-off branch
	car transfer *has* transferred car
	local area *includes* branch
	rental car *is owned by* local area
	state of affairs *occurs at* date/time

Related facts:	the noun concept 'transfer drop-off date/time' *is* a role that *ranges over* the noun concept 'date/time'
	the noun concept 'transfer drop-off branch' *is* a role that *ranges over* the noun concept 'branch'
	the noun concept 'transferred car' *is* a role that *ranges over* the noun concept 'rental car'

Issue 9459 Change "Synonymous Form" to "Synonymous Statement"

It is obligatory that the country *of* the transfer drop-off branch *of* an international return *is* the country of registration *of* the transferred car *of* the international return.

Synonymous Statement:	It is prohibited that the country *of* the transfer drop-off branch *of* an international return *is* not the country of registration *of* the transferred car *of* the international return.
Guidance Type:	operative business rule
Description:	When, as a result of a one-way rental, a car has been dropped off in a different country, it can be moved only back to its country of registration
Enforcement Level:	pre-authorized
Supporting fact types:	car transfer *has* transfer drop-off branch
	car transfer *has* transferred car
	branch *has* country
	rental car *has* country of registration
	$thing_1$ *is* $thing_2$
Related facts:	the noun concept 'transferred car' *is* a role that *ranges over* the noun concept 'rental car'
	the noun concept 'international return' *is* a category *of* the noun concept 'car transfer'
	the noun concept 'transfer drop-off branch' *is* a role that *ranges over* the noun concept 'branch'

At the drop-off date/time *of* an international return it is obligatory that the transferred car *of* the international return *is owned by* the local area that *includes* the transfer drop-off branch *of* the international return.

Guidance Type:	operative business rule
Description:	When a car is moved between branches in different local areas in the same country, ownership moves to the local area of the receiving branch.
Enforcement Level:	pre-authorized
Supporting fact types:	car transfer *has* transfer drop-off branch
	car transfer *has* transferred car
	local area *includes* branch
	state of affairs *occurs at* date/time
Related facts:	the noun concept 'transferred car' *is* a role that *ranges over* the noun concept 'rental car'
	the noun concept 'international return' *is* a category *of* the noun concept 'car transfer'
	the noun concept 'transfer drop-off branch' *is* a role that *ranges over* the noun concept 'branch'

E.2.2.2.10 EU-Rent ~~Admonitions and Affirmations~~ Advices expressed in EU-Rent's English Vocabulary

Issue 9475/9945 Replace text

It is possible that the notification date/time *of* a bad experience that *occurs during* a rental *is after* the actual return date/time *of* the rental.

Guidance Type: ~~affirmation~~ advice of possibility

Note: This is an unconditional expression - "after" has no business intent to imply there is a prohibition on "on or before"

Description: A 'bad experience' may not be known at rental return.
The notification of, say, police action for a moving traffic offense may be received by EU-Rent some time after rental return.

Supporting fact types: bad experience *occurs during* rental

bad experience *has* notification date/time

rental *has* actual return date/time

date/time_1 *is after* date/time_2

Related facts: the noun concept 'notification date/time' *is* a role that *ranges over* the noun concept 'date/time'

the noun concept 'return date/time' *is* a role that *ranges over* the noun concept 'date/time'

It is permitted that the rental car that *is moved by* a car transfer *is in need of service*.

Guidance Type: ~~admonition~~ advice of permission

Description: A car that is in need of service (i.e., has a service reading higher than 5000 miles) may be transferred between branches.
All relevant rules apply. One that is important is that the car's service reading must not go over 5500 miles. So, if the distance between the branches would take the service reading over this limit, the transfer would not be allowed.

Note: Such a transfer would require that any service scheduled for the car be cancelled and a service scheduled at a service depot in the local area of the receiving branch.

Supporting fact types: rental car *is moved by* car movement

car transfer *includes* one-way car movement

Related facts: the concept 'one-way transfer movement' *is* a role *of* the concept 'car movement'

'*being in need of service*' *is* a characteristic *of* the concept 'rental car'

It is permitted that a renter *has* more than one advance rental.

Guidance Type: ~~admonition~~ advice of permission

Description: A renter may make multiple rental bookings, each establishing an advance rental.

Note: There is an operative business rule that governs this permission. A renter is allowed to have only one car at a time in his possession, so the rental periods of his advance rentals must not overlap.

Supporting fact type: renter *has* rental

Related facts: the concept 'advance rental' *is* a category *of* the concept 'rental'

It is permitted that the renter *of* a rental *is* an additional driver *of* a rental.

Guidance Type:	~~admonition~~ advice of permission
Description:	The person who is the renter for a rental may be an additional driver for another rental – even if both rentals are open at the same time.
Note:	There is an operative business rule that governs this permission. A person cannot be the renter and an additional driver on the same rental.
Supporting fact types:	rental *has* renter rental *has* additional driver
Related facts:	the concept 'renter' *is* a role *of* the concept 'driver' the concept 'additional driver' *is* a role *of* the concept 'driver'

It is permitted that the drop-off branch *of* a rental *is* not the return branch *of* the rental.

Guidance Type:	~~admonition~~ advice of permission
Description:	There is no rule that allows an EU-Rent branch to refuse a rental return because it is not the return branch for the rental.
Note:	EU-Rent wants its cars back at the end of rental. It will accept return at any branch. It will charge a location penalty if the branch is not the return branch of the rental -- but, in any case, it wants its car back.
Supporting fact types:	rental *has* drop-off branch rental *has* return branch thing_1 *is* thing_2

Issue 9955 Replace text

E.2.2.2.11 EU-Rent Business Rules related to Business Processes

Business processes are outside the scope of SBVR and are the subject of another OMG RFP, "Business Process Definition Metamodel" (BPDM).

In practice, however, business rules are closely related to business processes. This subclause suggests how some process-related rules could be formulated, without encroaching on BPDM territory.

For example, "business process" and "event" are not explicitly defined in SBVR. Of course, an enterprise-specific vocabulary could ~~define~~ include them.

In this subclause, "process" and "event" are implied in the business rules and vocabulary. This area is likely to change with further development of BPDM, and agreement within the OMG on how BPDM and SBVR should be integrated. It is suggested that these examples should be revisited then.

E.2.2.2.11.1 Example illustrating pre-conditions

It is obligatory that a request for pick-up *is accepted* only if an assigned rental *matches* the request for pick-up and the renter that *is responsible for* the assigned rental *has* a valid credit card and the renter *provides* current contact details and each driver *of* the rental *has* a valid driver license.

Guidance Type:	operative business rule
Enforcement Level:	strict

Note: The (implied) process is “rental pick-up.” If the conditions are not met, the request is not accepted and the procedure is not started.

Note: This rule in could be (and, in practice, probably should be) split into four simpler rules, each giving one precondition.

Supporting fact types: request for pick-up *matches* rental

renter *has* credit card

renter *provides* contact details

driver *has* driver license

Related facts: the noun concept ‘assigned rental’ *is* a role that *ranges over* the noun concept ‘rental’

the noun concept ‘valid credit card’ *is* a role that *ranges over* the noun concept ‘credit card’

the noun concept ‘current contact details’ *is* a category *of* the noun concept ‘contact details’

the noun concept ‘valid driver license’ *is* a role that *ranges over* the noun concept ‘driver license’

‘*being accepted*’ *is* a characteristic *of* ‘request for pick-up’

E.2.2.2.11.2 Example illustrating post-conditions

Issue 9753 Change to subheadings
Issue 9449 Revise text
Issue 11291 Revise text

At the actual start date/time *of* a rental it is obligatory that the estimated rental charge *is provisionally charged to* some credit card *of* the renter that *is responsible for* the rental and that the renter *has possession of* the rented car *of* the rental.

Guidance Type: operative business rule

Enforcement Level: strict

Note: The actual start/time date of a rental is the expected end event of an (implied) “rental pick-up” process - the process whose pre-conditions were illustrated in the preceding example. If the pick-up is successful - so that the rental actually starts - the provisional charge will have been made and the renter will have the car.

Note: If the procedure fails (say, the charge to the credit card is not accepted, the renter is not able to use the controls of the car, or the car is not working and no substitute is available) the estimated charge is not made, the renter does not have possession of an EU-Rent car, and the rental does not start.

Note: As for the previous example, the business rule in the example could be split into two simpler rules, each giving one postcondition.

Supporting fact types: estimated rental charge *is provisionally charged to* credit card

renter *has* credit card

renter *is responsible for* rental

renter *has possession of* rented car

rental *has* rented car

E.2.2.2.11.3 Examples illustrating Invariants provided by Structural Rules

It is necessary that the cash rental price *of* each rental that *is the responsibility of* a corporate renter *is based on* the cash rental rates *of* the corporate rental agreement that *is available to* the corporate renter.

Guidance Type:	structural business rule
Supporting fact types:	rental *has* base rental price
	cash rental price *is based on* cash rental rate
	corporate rental agreement *has* applicable rental rate
	renter *is responsible for* rental
	corporate rental agreement *is available to* corporate renter
Related facts:	the noun concept 'applicable rental rate' *is* a role that *ranges over* the noun concept 'cash rental rate'
	the noun concept 'corporate renter' *is* a category *of* the noun concept 'renter'
	the noun concept 'cash rental price' *is* a category *of* the noun concept 'base rental price

It is necessary that the rental duration *of* each rental that *is the responsibility of* a corporate renter *is* not *greater than* the maximum rental duration *of* the corporate rental agreement that *is available to* the corporate renter.

Guidance Type:	structural business rule
Supporting fact types:	rental *has* rental duration
	corporate rental agreement *has* maximum rental duration
	renter *is responsible for* rental
	corporate rental agreement *is available to* corporate renter
Related facts:	the noun concept 'maximum rental duration' *is* a role that *ranges over* the noun concept 'duration'
	the noun concept 'rental duration' *is* a role that *ranges over* the noun concept 'duration'
	the noun concept 'corporate renter' *is* a category *of* the noun concept 'renter'

E.2.2.2.11.4 Examples illustrating Business Rules that cause actions related to Events

It is obligatory that the insurer *of* each operating company *is notified of* each overdue rental that *has* a pick-up branch that *is in* the operating company.

Guidance Type:	operative business rule
Enforcement Level:	deferred
Note:	The (implied) event is "scheduled return date/time + 24 hours." As well as changing the status of a rental, it would cause the action "notify insurer."
Supporting fact types:	operating company *has* insurer
	rental *has* pick-up branch
	pick-up branch *is in* operating company
Related facts:	'being overdue' *is* a characteristic *of* 'rental'

If the drop-off location *of* a rental *is* not the EU-Rent site *of* a branch then it is obligatory that the rented car *of* the rental *is recovered from the* drop-off location *to* some branch.

Guidance Type: operative business rule

Enforcement Level: deferred

Note: The (implied) event is "notification of rental drop-off at a location that is not a branch" - it would cause the action "recover car"

Supporting fact types: rental *has* drop-off location

rental organization unit *is based at* EU-Rent site

rented car *is recovered from* non-EU-Rent location *to* branch

$thing_1$ *is* $thing_2$

Related facts: the noun concept 'drop-off location' *is* a role that *ranges over* the noun concept 'location'

the noun concept 'non-EU-Rent-location' *is* a category *of* the noun concept 'location'

the noun concept 'EU-Rent-site' *is* a role that *ranges over* the noun concept 'location'

the noun concept 'branch *is* a role that *ranges over* the noun concept 'rental organization unit'

E.2.3 Common Vocabulary

This subclause illustrates some common SBVR vocabulary that could be adopted by enterprise-specific vocabularies.

In reality this vocabulary would be larger, containing many common terms and fact type forms ~~forms of expression~~ useful in describing enterprises. Here we have included some extracts that are directly relevant to the EU-Rent example in this annex.

E.2.3.1 General

text

Source: Unicode 4.0.0 Glossary ['Character Sequence']

General Concept: expression

$thing_1$ *is* $thing_2$

Definition: The $thing_1$ and the $thing_2$ *are the same* thing

thing [individual concept] *is changed*

Definition: the extension of the individual concept is different at one point in time from what it is at a subsequent point in time

E.2.3.2 Numbers

Issue 10568 Remove Note

integer

Definition: number with no fractional part

~~Note:~~ ~~The Integer Namespace, in the Namespace Registration Vocabulary, has designations for all of the integers~~

integer$_1$ *is less than* integer$_2$

Definition: The integer$_1$ *is numerically less than* the integer$_2$

Synonymous Form: integer$_1$ < integer$_2$

Synonymous Form: integer$_2$ *is greater than* integer$_1$

Synonymous Form: integer$_2$ > integer$_1$

nonnegative integer

Definition: integer that is greater than or equal to zero

Synonym: whole number

E.2.3.3 Time

actual date/time

Concept Type: role

Definition: date/time at which a state of affairs occurs

Description: Used in business rules such as "the rental start date requested on a rental reservation must not be earlier than the actual date/time of submission of the reservation."

date

Definition: date/time that is to the precision of year-month-day

---

Issue 9449 Revise text

---

date/time

Dictionary Basis: the point of time at which a transaction or event takes place or is appointed to take place: a given point of time MWU ["date" 2,2]

Dictionary Basis: a point or period when something occurs **:** the moment of an event, process, or condition MWU ["time" 2,2A]

date/time$_1$ *is after* date/time$_2$

date/time$_1$ *is before* date/time$_2$

date/time *is in the future*

Definition: date/time *being after* the date/time of the current moment

Example: Each reserved rental (rental that does not yet have a car assigned) should have a scheduled pick-up date/time that *is in the future*.

date/time *is in the past*

Definition: date/time *being before* the date/time of the current moment

Each returned rental (rental for which the car has been returned to EU-Rent) should have an actual return date/time that *is in the past*.

duration

~~Concept Type:~~

Definition: quantity of elapsed time of a period, measured in some time unit(s)

Issue 9344 Add text
Issue 9449 Delete text / Add text

duration$_1$ *is at most* duration$_2$

Synonymous Form: duration$_2$ *is more than* duration$_1$

Synonymous Form: duration$_1$ *is less than or equal to* duration$_2$

duration *is measured in* time unit

Definition: Each duration *is measured in* at least one time unit.

end date/time

Concept Type: role

Definition: date/time at which period concludes

period

~~Concept Type:~~

Definition: A time interval measured from a start date/time to an end date/time

Necessity: The start date/time of each period is before the end date/time.

Example: “From 23-April-2004/11:30 to 27-April-2004/17:50”

Note: period is related to, but different from, duration. For the example above, the duration is “4 days, six hours and 20 minutes”. Different periods can have the same duration.

period *has* duration

Concept Type: is-property-of fact type

period *has* end date/time

Concept Type: is-property-of fact type

period *has* start date/time

Concept Type: is-property-of fact type

period$_1$ *overlaps* period$_2$

Definition: (the start date/time *of* period$_1$ *is after* the start date/time *of* period$_2$ and *before* the end date/time *of* period$_2$) or (the end date/time *of* period$_1$ *is after* the start date/time *of* period$_2$ and *before* the end date/time *of* period$_2$).

start date/time

Concept Type: role

Definition: date/time at which period begins

state of affairs *occurs after* date/time

Concept Type: associative fact type

state of affairs *occurs at* date/time

Concept Type: associative fact type

state of affairs *occurs before* date/time

Concept Type: associative fact type

state of affairs$_1$ *occurs before* state of affairs$_2$ *occurs*

Concept Type: associative fact type

E.2.3.3.1 Example of a reusable structure in common vocabulary

Fixed and variable periods, described below, are structures that can play roles included in other concepts. For example, “variable period” (with all its necessities and possibilities) is included in EU-Rent’s rental, with the role name “rental period.”

fixed period

Definition: period that cannot *be changed*

Example: Period in the past, e.g., the OMG Burlingame meeting time.

Example: Period defined by clock or calendar, e.g., “first ten days in May.”

Example: Period in the future fixed by fiat, e.g., trip for which you have bought air tickets that cannot be rescheduled or refunded.

fixed end date/time

Concept Type: role

Definition: date/time that *is the end of* a fixed period

fixed start date/time

Concept Type: role

Definition: date/time that *is the start of* a fixed period

fixed period *has* fixed end date/time

Necessity: The end date/time *of* a fixed period *is* not *changed*.

fixed period *has* fixed start date/time

Necessity: The start date/time *of* a fixed period *is* not *changed*.

variable period

Definition: period that *can be rescheduled*

Example: period *of* a EU-Rent rental

variable period *has* actual start date/time

Necessity: Each variable period *has* at most one actual start date/time

Necessity: The actual start date time *of* a variable period *is* not *changed*.

variable period *has* actual end date/time

Necessity: Each variable period *has* at most one actual end date/time

Necessity: The actual end date time *of* a variable period *is* not *changed*.

variable period *has* scheduled start date/time

Necessity: Each variable period *has* exactly one scheduled start date/time

Possibility: The scheduled start date/time *of* a variable period *is changed before* the actual start date/time *of* the variable period.

Necessity: The scheduled start date/time *of* a variable period *is* not *changed after* the actual start date/time *of* the variable period.

Note: Additional constraints may be added in specific contexts - e.g., in EU-Rent the cut-off for changing the start date of a points rental is 5 days before its scheduled start date/time.

variable period *has* scheduled end date/time

Necessity: Each variable period *has* exactly one scheduled end date/time

Possibility: The scheduled end date/time *of* a variable period *is changed before* the actual end date/time *of* the variable period.

Necessity: The scheduled end date/time *of* a variable period *is* not *changed after* the actual end date/time *of* the variable period.

Note: Additional constraints may be added in specific contexts - e.g., EU-Rent won't allow the scheduled end date of a rental to be changed so that the rental would have duration of more than 90 days.

variable period *has* duration

Description: Duration of a variable period is measured in one of three ways, depending on what is known at the time of measurement:

(1) Before the actual start date/time the duration of a variable period is measured from scheduled start date/time to scheduled end date/time.

(2) At any date/time between actual start date/time and actual end date/time the duration of a variable period is measured from actual start date/time to scheduled end date/time.

(3) At any date/time after the actual end date/time the duration of a variable period is measured from actual start date/time to actual end date/time (i.e., the period is then fixed).

Annex F
(informative)

The RuleSpeak® Business Rule Notation

RuleSpeak® is an existing, well-documented[1] business rule notation developed by Business Rule Solutions, LLC (BRS) that has been used with business people in actual practice in large-scale projects since the second half of the 1990s.

Annex C presented a business rule notation within SBVR Structured English that features prefixing rule keywords onto appropriate propositions. RuleSpeak can also use the constructs of SBVR Structured English, but embeds equivalent keywords within the propositions themselves (mixfix).

As discussed in Annex A, more than one notation for expressing business rules is possible using SBVR Structured English. (This is probably also true for other notations compliant with SBVR). Regardless of how expressed, equivalent semantics can be captured[2] and formally represented as logical formulations.

The following selected examples using the EU-Rent case study illustrate use of RuleSpeak. The complete set of examples for EU-Rent in RuleSpeak is provided in subclause F.3, with additional comments.

Issue 9448 Change formatting
Issue 9449 Revise text

1	Structural business rule	It is necessary that each rental *has* exactly one requested car group.
	RuleSpeak version	Each rental always *has* exactly one requested car group.
2	Operative business rule	It is obligatory that the rental duration *of* each rental *is at most* 90 rental days.
	RuleSpeak version	The rental duration *of* a rental must not *be more than* 90 rental days.
3	Operative business rule	It is obligatory that each driver *of* a rental *is* qualified.
	RuleSpeak version	A driver *of* a rental must *be* qualified.

1. [Ross2003], Clauses 8-12. Versions of RuleSpeak have been available on the Business Rule Solutions, LLC website (www.BRSolutions.com) since the late 1990s. Public seminars have taught the syntax to thousands of professionals starting in 1996 (www.AttainingEdge.com). The original research commenced in 1985, and was originally published in 1994 [Ross1997].
2. For a business-oriented, SBVR-compliant approach, see [Ross2005], Clauses 4-5.

4	Operative business rule	~~If the drop-off location *of* a rental *is* not the EU-Rent site *of* the return branch *of* the rental then it is obligatory that the rental *incurs* a location penalty charge.~~ It is obligatory that the rental *incurs* a location penalty charge if the drop-off location *of* a rental *is* not the EU-Rent site that *is base for* the return branch *of* the rental.
	RuleSpeak version	A rental must *incur* a location penalty charge if the drop-off location *of* the rental *is* not the EU-Rent site that *is base for* the return branch *of* the rental.
5	Operative business rule	It is necessary that the rental charge *of* a rental must *be calculated in* the business currency *of* the rental.
	RuleSpeak version	The rental charge *of* a rental *is* always *calculated in* the business currency *of* the rental.
6	Operative business rule	It is permitted that a rental *is open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.
	RuleSpeak version	A rental may *be open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.
7	Operative business rule	It is obligatory that *at* the actual return date/time *of* each in-country rental and each international inward rental the local area that *includes* the return branch *of* the rental *owns* the rented car *of* the rental.
	RuleSpeak version	The local area that *includes* the return branch *of* an in-country rental or international inward rental must *own* the rented car *of* the rental *at* the actual return date/time *of* the rental.
8	Operative business rule	It is obligatory that *at* the actual pick-up date/time *of* each rental the fuel level *of* the rented car *of* the rental *is* full.
	RuleSpeak version	The fuel level *of* the rented car *of* a rental must *be* full *at* the actual pick-up date/time *of* the rental.
	Issue 9475/9945: Change text	
9	~~Affirmation~~ advice of possibility	It is possible that the notification date/time *of* a bad experience that *occurs during* a rental *is after* the actual return date/time *of* the rental.
	RuleSpeak version	The notification date/time *of* a bad experience that *occurs during* a rental *is* sometimes *after* the actual return date/time *of* the rental.
10	~~Admonition~~ advice of permission	It is permitted that the drop-off branch *of* a rental *is* not the return branch *of* the rental.

RuleSpeak version The drop-off branch *of* a rental need not *be* the return branch *of* the rental.

F.1 Expressions in RuleSpeak

Issue 9475/9945 Change text

RuleSpeak builds on the same expression forms described in Annex C (C.1), with the minor difference that distinct keywords are used for the Modal Operations related to business rules. The following subclause presents the RuleSpeak alternative rule keywords for Rules and ~~Affirmations and Admonitions~~ Advices.[3]

F.1.1 Modal Operations in RuleSpeak

Issue 9753 Revise text

~~Modality~~ Modal claim type	Statement form	SBVR Structured English keywords	RuleSpeak keywords
obligation ~~claim~~ formulation	'obligative statement' form	it is obligatory that *p*	*r* must *s*
obligation ~~claim~~ formulation embedding a logical negation	'prohibitive statement' form 'restricted permission statement' form	it is prohibited that *p* it is permitted that *p* only if *q*	*r* must not *s* *r* may *s* only *t*
permissibility ~~claim~~ formulation	'unrestricted permission statement' form	it is permitted that *p*	*r* may *s* *r* need not *s*
necessity ~~claim~~ formulation	'necessity statement' form	it is necessary that *p*	*r* always *s*
necessity ~~claim~~ formulation embedding a logical negation	'impossibility statement' form 'restricted possibility statement' form	it is impossible that *p* it is possible that *p* only if *q*	*r* never *s* *r* can *s* only *t*
possibility ~~claim~~ formulation	'unrestricted possibility statement' form	it is possible that *p*	*r* sometimes *s* *r* can *s*

3. It is important to note that use of these keywords must be in a context that is clearly indicated to be for Rules and ~~Affirmations/ Admonitions~~ Advices only.

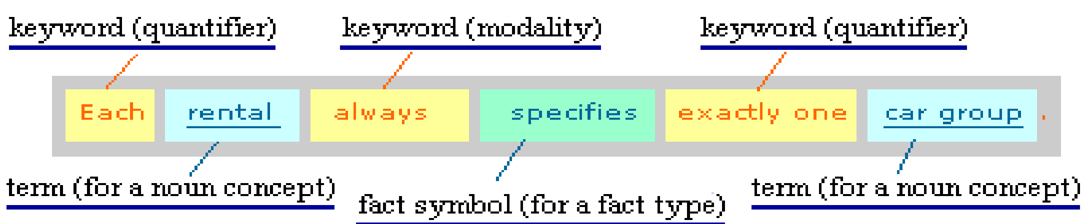
NOTES:

1. *p* and *q*, and *r*, *s*, and *t*, are all parts of the same proposition, say u.
2. In a permissibility ~~claim~~ formulation or a possibility ~~claim~~ formulation, the ‘only’ is always followed immediately by one of the following:
   (a) an ‘if’ (yielding ‘only if’).
   (b) a preposition.
   An example of a business rule statement using the ‘only [preposition]’ form is the following:
   A spot discount for a rental may be given only *by* a branch manager.

F.1.2 Example in RuleSpeak

Each rental always *specifies* exactly one car group.

The example above includes three keywords or phrases, two terms, and one fact symbol, as illustrated below.



Issue 9475/9945 Change text

As noted above, every Operative Business Rule or ~~Affirmation or Admonition~~ Advice can be stated by using one of the following embedded keywords.

must	*or* should	rule keyword
must not	*or* should not	rule keyword

Issue 9580 Revise text
Issue 9475/9945 Change text

---

may ... only	*often as in* may … only if	rule keyword
may	*or* need not	~~affirmation/admonition~~ advice keyword

Every Structural Rule or ~~Affirmation or Admonition~~ Advice can be stated by using one of the following embedded keywords.

always		rule keyword
never		rule keyword
can … only	*often as in* can … only if	rule keyword
sometimes	*or* not always	~~affirmation/admonition~~ advice keyword

Special-purpose keywords for indicating specific kinds of Structural Rules include the following. In these forms, “always” is assumed implicit.

is to be considered		for derivation or inference
is to be computed as		for computation
is to be fixed at [number]	or is to be [number]	for establishing constants

Among the most basic usage rules and guidelines of RuleSpeak are the following. (Note that these usage rules and ~~affirmations/admonitions~~ advices are given using proper RuleSpeak notation.)

Issue 9580 Revise text
Issue 9475/9945Change text

1. ‘Should’ may be used in place of ‘must’ in expressing a business rule only if one of the following is true:
   - The business rule does not have an enforcement level.
   - The business rule has an enforcement level, and that enforcement level is consistent with the English sense of ‘should’.

   Comment: To say this differently:

   ‘Should’ must not be used in place of ‘must’ in expressing a business rule if all of the following are true:
   - The business rule has an enforcement level.
   - The enforcement level of the business rule is inconsistent with the English sense of ‘should’.
2. ‘May’ must be used in the sense of “permitted to” in RuleSpeak. ‘May’ must not be used in the sense of “might.”[4]
3. An ~~affirmation or admonition~~ advice must not include a rule keyword.
4. A statement expressing a rule or ~~affirmation or admonition~~ advice should not begin with a condition.

   Comment: ‘Condition’ as used here means a qualification set off by ‘if’, ‘while’, ‘when’, etc. (e.g., if a rental is open...).
5. A double negative should be avoided in expressing a rule in RuleSpeak.

   Comment: Double negatives, especially using two ‘not’s, are generally undesirable in good English usage, and often prove particularly troublesome in rule statements.

   Example.

   Rule: A withdrawal *from* an account must not *be made* if the account *is* not *active*.

   Revised rule: A withdrawal *from* an account may *be made* only if the account *is active*.

   Comment: The revised rule is expressed in the form of a ‘restricted permissive statement’.

4. [Ross2003], p. 130.

F.2 Concepts, Definitions, and Rules: RuleSpeak Practices

SBVR is very flexible in supporting alternative practices with respect to rules and definitions. This flexibility is enabled by the underlying logical formulations and their underpinning in formal logic.

Two core RuleSpeak practices with respect to definitions are the following.[5]

1. **"Essence" by Definitions.** A definition should always focus on the core essence of a concept – that is, on *fundamental* meaning that is unlikely to change. Such meaning is expressed as naturally as possible. The form of language used in common dictionaries is strongly preferred.

2. **"Boundaries" by Rules.** *All* constraints should be expressed as rules separate from definitions. Such rules generally define the 'boundary conditions' of a concept; that is, when something is or is not an instance of the concept. Since specific boundaries for a concept (e.g., "gold customer") can change over time, they should not be embedded in definitions. An additional advantage – crucial for communication with and among business people – is that the underlying vocabulary can be kept as compact and as focused as possible.

Experience in large-scale projects indicates that these core practices:

- Ensure good business communication.
- Produce friendly and highly stable definitions.
- Scale extremely well for complex business problems featuring hundreds or thousands of rules.

RuleSpeak might therefore be characterized as more 'rule-ish' than the approach described in Annex E. RuleSpeak is well-suited for practitioners who want to:

- Move faster to rule capture.
- Use more natural (less formal) wordings for definitions.

These issues are pragmatic concerns for business rule projects. It is important to remember, of course, that under SBVR either approach (and others) can produce identical semantics 'under the covers' (i.e., in logical formulations).

F.2.1 Example in RuleSpeak

A EU-Rent definition and set of related specifications taken from Annex E concerning "agency" (a type of "branch") serve to illustrate. The RuleSpeak approach is outlined subsequently.

F.2.1.1 Sample Definition and Related Specifications for "Agency" from Annex E

'Agency': service desks in hotels, travel agents, etc. They have storage space for few cars, and are operated on demand by part-time staff who will typically do the entire workflows for rental and return.

agency

Concept Type: branch type

Definition: branch that does not *have an EU-Rent location* and *has minimal car storage* and *has on-demand operation*

5. [Ross2005], Clause 4, pp 51-52.

rental organization unit *having an EU-Rent location*

Concept Type: characteristic

Definition: rental organization unit that *is based at* an EU-Rent site that *is owned by* EU-Rent

Note: Some things *are based at* EU-Rent sites that *are owned by third parties such as hotels and travel agents.*

rental organization unit *having minimal car storage*

Concept Type: characteristic

Definition: rental organization unit that *has* car storage that *can accommodate a small number of* rental cars

rental organization unit *having on-demand operation*

Concept Type: characteristic

Definition: rental organization unit that *has* hours of operation that *are flexible in response to customer demand*

F.2.1.2 RuleSpeak Approach for the "Agency" Example

Issue 9399 Replace text

---

1. Find a suitable definition from a standard dictionary, or if available, an industry glossary, to serve as the basis for the definition. The Merriam-Webster Unabridged Dictionary offers the following for "agency," an appropriate basis for an 'essence' definition.

   *4a: an establishment engaged in doing business for another *an advertising agency* *an employment agency**

agency

Concept Type: branch type

Definition: another company engaged in conducting EU-Rent business operations

2. Define Fact Types for 'agency'. For this example, assume an agency has the following (binary) fact types by virtue of being a branch. These fact types would probably be indicated as properties.
   - branch *has* location
   - branch *has* car storage capacity
   - branch *has* operating mode

Comments:

- RuleSpeak does not emphasize using characteristics for building definitions.
- In practice, fact types are generally not given definitions in RuleSpeak. ~~If all noun concepts represented by roles for a fact type are well-defined, a definition for the fact type itself generally adds very little~~. When the meaning of a fact type is the dictionary meaning for the verb phrase in the context of well-defined noun concepts for the things that play the roles, a definition for the fact type itself generally adds very little.
- For the sake of simplicity, assume that location, car storage, and operating mode already have suitable definitions.

3. Define the appropriate structural rule(s) to establish (current) 'boundaries' for the concept 'agency'. Note that these 'boundaries' might be modified, expanded, or contracted over time.

   All of the following are always true for an agency:
   - It *has* a third-party location.
   - It *has* a minimal car storage capacity.
   - Its operation mode *is* on-demand.

4. Specify structural rules for derived terms (e.g., "third-party location," "minimal," etc.).

   A location *is to be considered* a **third-party location** if *located at* an EU-Rent site that *is owned by* a third party.

   The car storage capacity *of* a branch is to be considered **minimal** if *less than* … *[condition(s)]*

5. Ensure all non-derived terms have "essence" definitions.

on-demand

Definition: flexible in response to customer demand

Comment: Derived concepts are generally not given definitions in RuleSpeak since the structural rule(s) for them are, literally, *definitive*.

F.2.2 Structural Rules vs. Operative Rules

In RuleSpeak, the distinction between structural rules and operative rules is viewed as follows.[6]

- *Structural rules* prescribe criteria for how the business chooses to organize ("structure") its business semantics. Such rules express criteria for correct decisions, derivations, or business computations. Structural rules supplement definitions.
- *Operative business rules* focus directly on the propriety of conduct in circumstances (business activity) where willful or uninformed actions can fall outside the boundaries of behavior deemed acceptable. Unlike structural rules, operative rules can be violated *directly*.

The distinction is clear-cut in most cases; in some, it is more difficult. For example, consider "booking" in the EU-Rent case study. "Booking" (like "order," "reservation," "registration," etc.) is essentially a 'made-up' device of the business. It is an artifact of knowledge that exists 'simply' to help manage complex, expensive resources.

Issue 9949 Revise/Remove text

Therefore, rules about creating bookings (e.g., that the requested pick-up date-time is to be **after** the booking date-time) are to be viewed as structural. If not followed (applied) in attempting to make a booking, no booking results. In other words, since bookings are a knowledge 'thing', the business can establish definitive rules for them. These are the "boundary" rules discussed earlier.

Now consider "actual pick-up date-time," the date-time when possession of a rental car is actually handed over to a rental customer (or is *said* to have been anyway). EU-Rent might want to avoid post-dating handovers -- i.e., have a rule that the actual pick-up date-time is to be after the booking date-time.

6. [Ross2005], Clauses 5 and 6.

This case is quite different. “Actual pick-up date-time” reflects activity (or the communication thereof) outside the realm of knowledge artifacts -- i.e., conduct that takes place in the ‘real world’. Because such rules can be broken (by people), they are operative.

Refer to the Rule Speak best practices presented in [Ross2005].[7]

~~RuleSpeak Best Practice: Carefully distinguish what should be (according to the structural rules) vs. what really is, based on actual business decisions / actions. One or more operative rules are then specified to constrain 'what really is' against 'what should be'. These latter rules, being operative, are the ones that can be violated.~~

F.3 Complete Set of EU-Rent Examples in RuleSpeak

Issue 9475/9945 Change text

This subclause provides one-by-one RuleSpeak counterparts for the EU-Rent guidance (business rules and ~~affirmations/admonitions~~ advices) presented in Annex E[8]. This restatement provides semantically equivalent expression that is more business friendly.

Comment: Many of the guidance statements in Annex E are supported by descriptions, which reflect EU-Rent users’ informal statements of the guidance, and by fact types and levels of enforcement. That material has been removed from here for the sake of brevity. Refer to Annex E for details.

F.3.1 Rule Set -- Rental Rules

It is necessary that each rental *has* exactly one requested car group.

A rental always *has* exactly one requested car group. Comment: “A” may be used in place of “each” with no change in meaning, as follows[9]. (This note will not be repeated subsequently.) Each rental always *specifies* exactly one car group

Guidance Type: structural business rule

It is necessary that each rental *has* exactly one rental period.

A rental always *has* exactly one rental period.

Guidance Type: structural business rule

7. [Ross2005], Clause 6, pp. 107-108.
8. As of the time of this writing.
9. [Ross2003]

It is necessary that each rental *has* exactly one return branch.

A rental always *has* exactly one return branch.

Guidance Type: structural business rule

It is necessary that the scheduled pick-up date/time *of* each advance rental *is after* the booking date/time *of* the rental booking that *establishes* the advance rental.

The scheduled pick-up date/time *of* an advance rental *is* always *after* the booking date/time *of* the rental booking that *establishes* the advance rental.

Guidance Type: structural business rule

F.3.2 Rule Set -- Charging / Billing / Payment Rules

Issue 9449 Revise text

It is permitted that a rental *is open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.

A rental may *be open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.

Guidance Type: operative business rule

It is necessary that the rental charge *of* each rental *is calculated in* the business currency *of* the rental.

The rental charge *of* a rental *is* always *calculated in* the business currency *of* the rental.

Guidance Type: structural business rule

If the renter of a rental requests a price conversion then it is obligatory that the rental charge of the rental is converted to the currency of the price conversion.

The rental charge of a rental must be converted to the currency of a price conversion requested by the renter of the rental. Comment: RuleSpeak does not recommend the "If …then…" syntax for operative business rules[10]. (This note will not be repeated subsequently.

Guidance Type: operative business rule

It is necessary that each cash rental honors its lowest rental price.

A cash rental always honors its lowest rental price.

Guidance Type: structural business rule

[From Annex E] "The structural business rule above can be elaborated as three detailed structural business rules:"

It is necessary that a cash rental price for a cash rental that is calculated because of EU-Rent price changes and that is less than the lowest rental price honored by the rental replaces the lowest rental price honored by the rental.

A cash rental price for a cash rental that is calculated because of EU-Rent price changes and that is less than the lowest rental price honored by the rental always replaces the lowest rental price honored by the rental.

It is necessary that a cash rental price for a cash rental that is calculated because of changes to the car group or rental duration of a rental replaces the lowest rental price honored by the rental.

A cash rental price for a cash rental that is calculated because of changes to the car group or rental duration of a rental always replaces the lowest rental price honored by the rental.

10. [Ross2003].

It is necessary that the lowest rental price *honored by* a rental *is* not replaced *after* the actual return date/time *of* the rental.

The lowest rental price *honored by* a rental *is* never replaced *after* the actual return date/time *of* the rental.

F.3.3 Rule Set -- Driver Rules

It is permitted that a rental *is open* only if each driver *of* the rental *is* not a barred driver.

A rental may *be open* only if each driver *of* the rental *is* not a barred driver.

Issue 9459 Change "Synonymous Form" to "Synonymous Statement"
Issue 9449 Revise text

Synonymous Statement: It is prohibited that a rental *is open* if a driver *of* the rental *is* a barred driver.

A rental must not *be open* if a driver *of* the rental *is* a barred driver.

Guidance Type: operative business rule

It is obligatory that each driver *of* a rental *is* qualified.

Each driver *of* a rental must *be* qualified.

Guidance Type: operative business rule

F.3.4 Rule Set -- Pick-up / Return Rules

Issue 9449 Revise text

It is obligatory that *at* the actual return date/time *of* each in-country rental and each international inward rental the local area *of* the return branch *of* the *rental owns* the rented car *of* the rental.

The local area *of* the return branch of an in-country or international inward rental must *own* the rented car *of* the rental *at* the actual return date/time *of* the rental. ~~NOTE: RuleSpeak treats this rule as structural, rather than operative, for the reasons given earlier.~~

Guidance Type: operative business rule

It is obligatory that the country *of* the return branch *of* each international inward rental *is* the country of registration *of* the rented car *of* the rental.

The country *of* the return branch *of* an international inward rental *is* always the country of registration *of* the rented car *of* the rental. NOTE: RuleSpeak treats this rule as structural, rather than operative, for the reasons given earlier.

Guidance Type: structural business rule

It is necessary that if a rental *is open* and the rental *is* not an international inward rental then the rented car *of* the rental *is owned by* the local area *of the* pick-up branch *of* the rental.

The rented car *of* a rental is always *owned by* the local area *of* the pick-up branch *of* the rental if the rental *is open* and the rental *is* not an international inward rental.

Guidance Type: structural business rule

If the actual return date/time *of* a rental *is after* the end date/time *of* the grace period *of* the rental then it is obligatory that the rental *incurs* a late return charge.

A rental must *incur* a late return charge if the actual return date/time *of* the rental *is after* the end date/time *of* the grace period *of* the rental.

Guidance Type: structural business rule

If the drop-off location *of* a rental *is* not the EU-Rent site that *is base for* the return branch *of* the rental then it is obligatory that the rental *incurs* a location penalty charge.

A rental must *incur* a location penalty charge if the drop-off location *of* the rental *is* not the EU-Rent site that *is base for* the return branch *of* the rental.

Guidance Type: operative business rule

If a rental *is* assigned then it is obligatory that the rental car that *is assigned to* the rental *is stored at* the pick-up branch *of* the rental.

The rental car *assigned to* a rental must *be stored at* the pick-up branch *of* the rental if the rental *is* assigned.

Guidance Type: operative business rule

At the actual pick-up date/time *of* each rental it is obligatory that the fuel level *of* the rented car *of* the rental *is* "full."

The fuel level *of* a rental car *assigned to* a rental must be "full" at the actual pick-up date/time *of* the rental.

Guidance Type: operative business rule

F.3.5 Rule Set -- Points Rental Rules

It is necessary that the booking date/time *of* a points rental *is at least 5 days* before the scheduled start date/time *of* the rental.

The booking date/time *of* a points rental is always *at least 5 days* before the scheduled start date/time *of* the rental.

Guidance Type: structural business rule

It is necessary that the renter *of* each points rental *is* a club member.

The renter *of* a points rental is always a club member.

Guidance Type: structural business rule

F.3.6 Rule Set -- Rental Period Rules

It is obligatory that the start date *of* each reserved rental *is in* the *future*.

The start date *of* a reserved rental must *be in* the *future*.

Guidance Type: operative business rule

It is prohibited that the start date *of* a reserved rental *is in* the *past*.

The start date *of* a reserved rental must not *be in* the *past*.

Guidance Type: operative business rule

Issue 9448 Change formatting
Issue 9449 Revise text

It is obligatory that the rental duration *of* a rental *is at most* 90 rental days.

The rental duration *of* a rental must *be at most* 90 rental days.

Guidance Type: operative business rule

If rental_1 *is* not rental_2 and the renter *of* rental_1 *is* the renter *of* rental_2 then it is obligatory that the rental period *of* rental_1 does not overlap the rental period *of* rental_2.

The rental period *of* rental_1 must not *overlap* the rental period *of* rental_2 if all the following are true:
- rental_1 *is* not rental_2
- the renter *of* rental_1 *is* the renter *of* rental_2.

Guidance Type: operative business rule

F.3.7 Rule Set -- Servicing Rules

It is obligatory that each rental car that *is* in need of service *has a* scheduled service.

A rental car in need of service must *have a* scheduled service.

Guidance Type: operative business rule

It is obligatory that the service reading *of* a rental car *is* at most 5500 miles.

The service reading *of* a rental car *is* always at most 5500 miles.

NOTE: RuleSpeak treats this rule as structural, rather than operative, for the reasons given earlier.

Guidance Type: structural business rule

If the rented car *of* an open rental *is in need of service* or *is in need of repair* then it is obligatory that the rental *incurs* a car exchange during rental.

An open rental must *incur* a car exchange during rental if the rented car *of* the rental *is in need of service* or *is in need of repair*.

Guidance Type: structural business rule

F.3.8 Rule Set -- Transfer Rules

At the transfer drop-off date/time *of* a car transfer it is obligatory that the transferred car *of* the car transfer *is owned by* the local area that *includes* the transfer drop-off branch *of* the car transfer.

The transferred car *of* a car transfer *is* always *owned at* the transfer drop-off date/time *of* a car transfer *by* the local area that *includes* the transfer drop-off branch *of* the car transfer. NOTE: RuleSpeak treats this rule as structural, rather than operative, for the reasons given earlier.

Guidance Type: structural business rule

It is obligatory that the country *of* the transfer drop-off branch *of* an international return *is* the country of registration *of* the transferred car *of* the international return.

The country *of* the transfer drop-off branch *of* an international return *is* always the country of registration *of* the transferred car *of* the international return. NOTE: RuleSpeak treats this rule as structural, rather than operative, for the reasons given earlier.

Guidance Type: structural business rule

Issue 9459 Change “Synonymous Form” to “Synonymous Statement”

Synonymous Statement: It is prohibited that the country *of* the transfer drop-off branch *of* an international return *is* not the country of registration *of* the transferred car *of* the international return.

NOTE: A RuleSpeak expression of the Synonymous Statement has been intentionally omitted. The form “prohibited … not” (or impossible … not”) is actually a double negative, which as noted earlier, RuleSpeak always discourages because it inevitably causes confusion.

- Prohibited that … not” is equivalent to “obligatory that not … not,” a double negative, which is shown more clearly in RuleSpeak, “must not … not.”
- Impossible that … not” is equivalent to “Necessary that not … not,” which would be more clearly a double negative in RuleSpeak, “never … not.”

At the drop-off date/time *of* an international return it is obligatory that the transferred car *of* the international return *is owned by* the local area that *includes* the transfer drop-off branch *of* the international return.

The transferred car *of* an international return *is* always *owned* at the drop-off date/time *of* an international return *by* the local area that *includes* the transfer drop-off branch *of* the international return. NOTE: RuleSpeak treats this rule as structural, rather than operative, for the reasons given earlier.

Guidance Type: structural business rule

Issue 9475/9945 Change heading

F.3.9 EU-Rent ~~Affirmations and Admonitions~~ Advices expressed in EU-Rent's English Vocabulary

It is possible that the notification date/time *of* a bad experience that *occurs during* a rental *is after* the actual return date/time *of* the rental.

The notification date/time *of* a bad experience that *occurs during* a rental *is* sometimes *after* the actual return date/time *of* the rental.

Guidance Type: ~~affirmation~~ advice of possibility

It is permitted that the rental car that *is moved by* a car transfer *is in need of service*.

The rental car *moved by* a car transfer may *be in need of service*.

Guidance Type: ~~admonition~~ advice of permission

It is permitted that a renter *has* more than one advance rental.

A renter may *have* more than one advance rental.

Guidance Type: ~~admonition~~ advice of permission

It is permitted that the renter *of* a rental *is* an additional driver *of* a rental.

The renter *of* a rental may *be* an additional driver *of* a rental.

Guidance Type: ~~admonition~~ advice of permission

It is permitted that the drop-off branch *of* a rental *is* not the return branch *of* the rental.

The drop-off branch *of* a rental need not *be* the return branch *of* the rental.

Guidance Type: ~~admonition~~ advice of permission

Annex G
(informative)

Concept Diagram Graphic Notation

A business vocabulary can be presented to a business audience using four simple main conventions, described in this Annex. These conventions have been purposely kept neutral of any particular modeling notations, and have been selected to be largely self-explanatory and visually intuitive. Note that a diagram using these conventions is only one view of a vocabulary and is intended to help in understanding some particular aspects of the vocabulary and the conceptual schema that underlies it.

Various graphic constructs are used to provide visual clarity (e.g., color, shading, font, font size, etc.). Unless explicitly stated, none of these carry any semantic or syntactic meaning. A diagram can be viewed in grayscale without losing information.

Issue 9952 Revise text

G.1 Boxes -- Concepts

A box of any size represents a core concept. The name in the box is the preferred ~~term (name)~~ name given to that concept. Refer to the Vocabulary for the precise meaning of each term. Because of the need to format within realistic bounds, some concepts re-appear in several diagrams.

For example, Figure G-1 depicts two concepts, termed ‘concept a’ and ‘concept s’.

Figure G.1 - concept a and concept s are core concepts within the model

G.2 Box-Within-A-Box -- Categories

G.2.1 Simple Categories

Straightforward categorization — where one element is a category of another element — is represented as a box within a box. Another way to think about this is that the inner box (the category) represents a specific kind or variation of the concept represented by the outer box (the more general concept).

There is no assumption in this graphic representation that box-within-a-box implies mutually exclusive categories, or represents an exhaustive or mandatory list of categories. When categories in the SBVR model are mutually exclusive, this constraint is documented in the Vocabulary. When the categories of a scheme are completely enumerated and required as shown, these constraints are documented in the Vocabulary.

For example, Figure G.2 depicts two concepts, termed ‘concept s’ and ‘concept t’ that are categories of a more general concept, termed ‘concept a’.

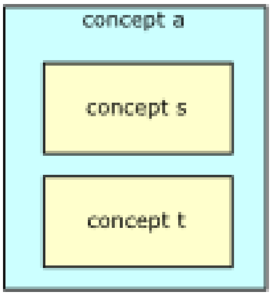


Figure G.2 - concept s and concept t are categories of concept a

G.2.2 Categorization Schemes and Segmentations

In some cases, categories form part of a designated categorization scheme. For these, a dashed-line box is used to depict the categorization scheme within the concept box. The scheme box surrounds the categories that make up the scheme. The categorization scheme's name is shown at the top of the scheme box that the scheme is for. Note that a category may appear in more than one scheme.

When the categorization scheme depicts a 'segmentation' -- a categorization scheme in which the set of categories are mutually exclusive and complete -- these constraints are documented in the Vocabulary. This may also be shown on the diagram as '[segmentation]' after the categorization scheme name.

For example, Figure G.3 depicts a categorization scheme, named 'Scheme X' that is for a concept (termed 'concept a'). Two concepts (termed 'concept s' and 'concept t') make up the scheme.

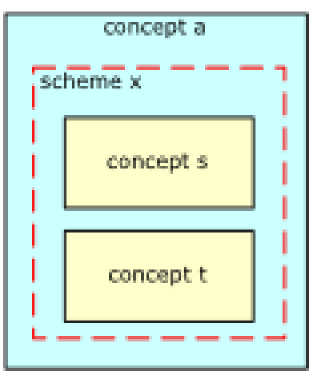


Figure G.3 - concept s and concept t are mutually exclusive categories of concept a within the categorization scheme 'scheme x'

G.3 Connections Between Boxes -- Fact Types

G.3.1 Binary Fact Types

A line connecting any two boxes (or the same box twice) indicates a connection between core concepts. Such a line represents a fact type. The labels adjacent to the lines are written as verbs or verb phrases so that the facts of the SBVR model can be read as simple sentences. These sentences convey the meaning of the connections in the context of the SBVR model; however, more explanation is given in the Vocabulary, along with the definitions for each of the terms involved.

The rules that apply to these constructs are also part of the SBVR model. However, these rules are not expressed in the model graphics. For example, the connection lines represent simple unconstrained facts (i.e., 'many-to-many' and 'optional' in both directions). While the diagram may suggest some rules, the final word on any rule is documented in the Vocabulary.

To avoid clutter, only one reading of a fact type is shown in the graphics. The fact type is read clockwise around the line, from participating concept, to verb phrase, to (other) participating concept. Additional readings, as useful, are provided in the Vocabulary.

Figure G.4 depicts two fact types, with one reading for each.

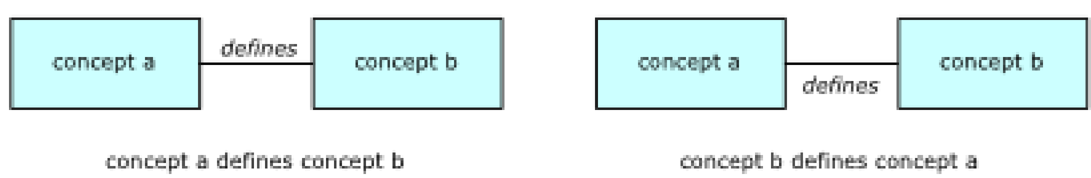


Figure G.4 - Reading two fact types, using 'defines' as a typical verb phrase

G.3.2 N-ary Fact Types

Where a connection involves more than two core concepts, a simple line cannot be used to represent the fact type. In this case, the fact type is shown as * with the fact type lines radiating from it to the participating concepts. The reading is placed adjacent to the * and no verbs are written on the lines.

Figure G.5 illustrates a ternary fact type and one of its readings.

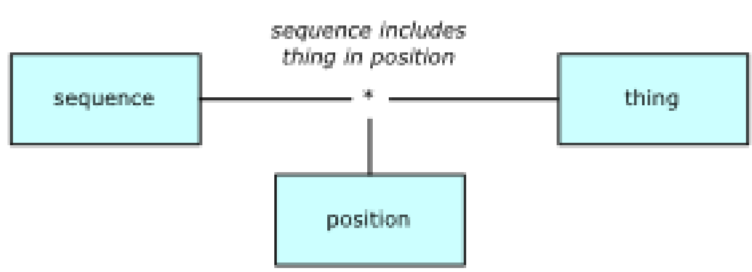


Figure G.5 - An n-ary fact type

G.3.3 Unary Fact Types

Unary fact types are shown using a similar * notation. A unary fact type is drawn as a line coming out of the concept box and ending with *. The fact type verb phrase is placed adjacent to the * symbol.

Figure G.6 illustrates a unary fact type.

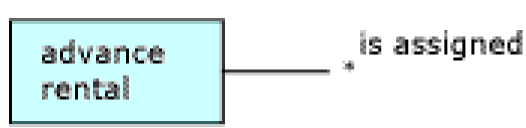


Figure G.6 - A unary fact type

G.3.4 'Objectified' Fact Types

When a noun concept is defined using objectification such that it is coextensive with a fact type it is shown as a box labeled with the primary term for the noun concept. The reading of the fact type is provided in a legend (or glossary). To aid in visually distinguishing these fact type-objectifying noun concepts from other concepts, the concept name is marked with * which provides the visual clue to look in the legend/glossary.

No verb phrase labels are written on the lines to the concepts that participate in the fact type. This permits the fact type itself to participate in other fact types without visual ambiguity.[1]

Figure G.7 depicts a fact type (rented car is rented from non-EU-Rent site to branch) and its objectification as the noun concept termed 'car recovery'.

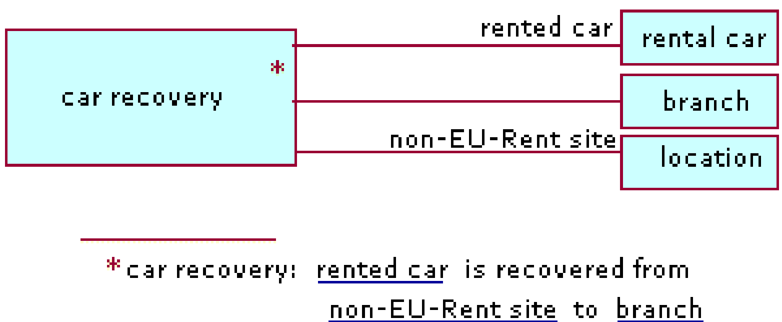


Figure G.7 - 'Objectified' fact type

Issue 9948 Revise text

G.4 Roles

A role name may be given to a concept's participation in a fact type. This is reflected as a term (the role name) adjacent to the box for the concept ~~playing the role~~ whose instances play in the fact type. There is no syntactic or semantic significance to the side of the line on which the role name is placed, other than careful placement to avoid confusion between the verb phrase and any role names. Figure G.8 depicts a role name 'part' given to the concept termed 'concept b' in this fact type.

Figure G.8 - Role name

1. There is a potential for confusion if the objectified fact type then participates in another fact type that is objectified, but this case is so rare that these conventions have elected simplicity for the typical cases over excruciating precision and the associated complexity.

Annex H
(informative)

Use of UML Notation in a Business Context to Represent SBVR-Style Vocabularies

Issue 9960 Revise text
Issue 9930 Replace text
Issue 9955 Add text / replace text

~~The purpose of the UML diagrams in Clauses 8 through 12 and Annex E is to display a vocabulary graphically. This kind of UML model is commonly called a 'Business Object Model' (BOM). Note that these diagrams are not the MOF metamodel that is generated from a vocabulary, and these two uses of UML should not be confused.~~

A purpose of the UML diagrams in clauses 8 through 12 and Annex E is to display a vocabulary graphically. This kind of UML model is commonly called a 'Business Object Model' (BOM). Note that diagrams in clauses 8 through 12 also show SBVR's MOF-based metamodel using an interpretation explained in clause 13. The vocabulary interpretation described below and the MOF interpretation explained in clause 13 use the same diagrams, but the two interpretations should not be confused. The two interpretations are based on different profiles.

A BOM is commonly used to convey a business vocabulary (e.g., the SBVR vocabulary) so its use should be familiar. The diagrams do not show any special stereotypes as long as conventions are explained. This Annex provides that explanation.

H.1 General Concept (Noun Concept)

The primary term for a concept that is not a role, individual concept, or fact type is shown as a class (rectangle). The rectangle is labeled with the concept's primary term, written just as the entry term would appear in a presentation of the vocabulary.

If there are additional terms for the concept they can be added within the rectangle, labeled as such -- e.g., "*also*: is-category-of fact type" as depicted in Figure H.1.

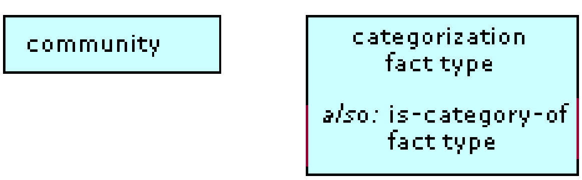


Figure H.1 - Two general concepts

Issue 9451 Add text to heading

H.2 Individual Concept (Noun Concept)

The name given to an individual concept is shown as an instance specification (rectangle). The name is followed by a colon and then by the term for its general concept. This text string is underlined within the rectangle.

While it is possible to have additional names for a given individual concept (i.e., names that are synonyms), the non-primary names of an individual concept are not typically reflected on the diagram. Figure H.2 depicts two individual concepts.

Figure H.2 - Two individual concepts

Alternatively, an individual concept can be depicted as an instance of its related general concept (noun concept), as in Figure H.3.

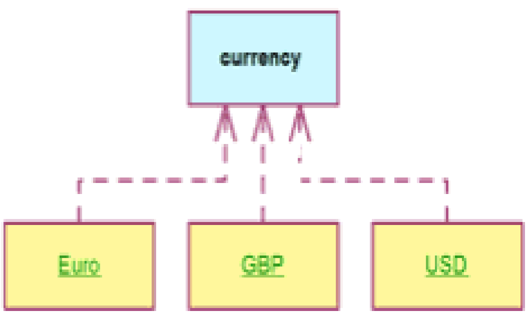


Figure H.3- Three individual concepts as instances of the related general concept

H.3 Fact Types

Use of the UML association notation works well for representing fact types in an SBVR-based vocabulary diagram. However, it is important to remember that an SBVR fact type is not an association. A fact type is a classifier that has particular semantics.

Issue 9958 Change form of expression with fact type form

H.3.1 Binary Fact Types

The fact type form ~~form of expression~~ of a binary fact type, other than one using 'has', is shown as an association (a line between rectangles). If there is another fact type form ~~form of expression~~ for the fact type that reads in the opposite direction, only the active form is needed if the other form is the normal passive form for the same verb.

Alternatively, both forms can be shown, one above the line and the other below. Either the 'clockwise reading rule' or a solid triangle as an arrow can be used to show the direction of reading. Figure H.4 illustrates three alternative presentations of a binary fact type.

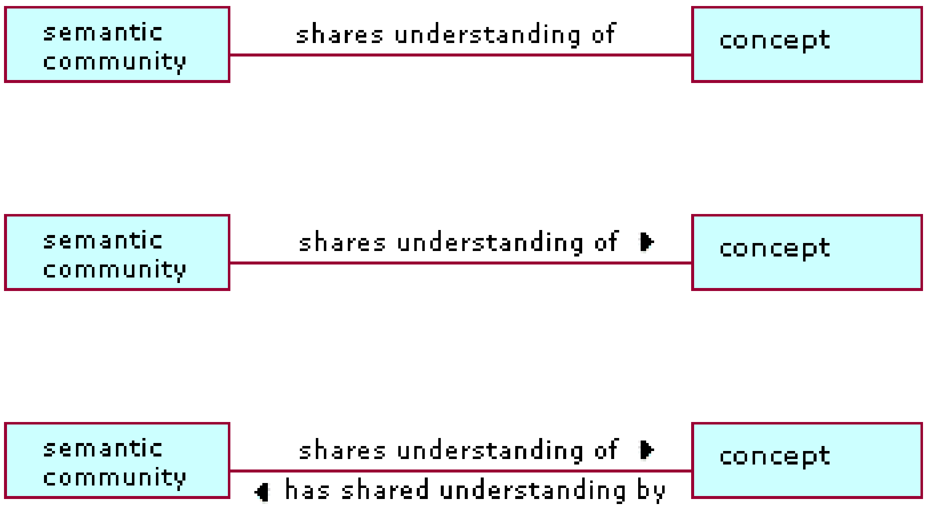


Figure H.4 - Three alternatives for presenting a binary fact type

---

Issue 9958 Change form of expression with fact type form
Issue 9948 Add text

H.3.2 Binary Fact Types using ‘has’

For each fact type form ~~form of expression~~ using ‘has’, the second role name is shown as an association end name. The verb ‘has’ is not shown on the diagram when giving an association end name. Each association end name in a diagram expresses a designation of a fact type role. An end name implies ‘has’ as shown in Figure H.5. Any verb phrase shown is assumed to be usable without the end name.

Figure H.5- Depicting the fact type ‘cash rental has lowest rental price’

When a binary fact type’s fact type form ~~form of expression~~ uses ‘has’ and there is no specialized role, the second role name is still reflected on the diagram in this consistent way (on the line adjacent to the rectangle) and ‘has’ is not displayed. This is illustrated in Figure H.6.

Figure H.6- Depicting the fact type ‘branch has country’

Issue 9958 Change form of expression with fact type form
Issue 9930 Change text, replace figure

H.3.3 Fact Types with Arity of 3 or more

For fact types with more than two roles, the UML association notation is used. The primary fact type form ~~form of expression~~ is shown, with the placeholders ~~included in square brackets~~ underlined as shown in Figure H.7.

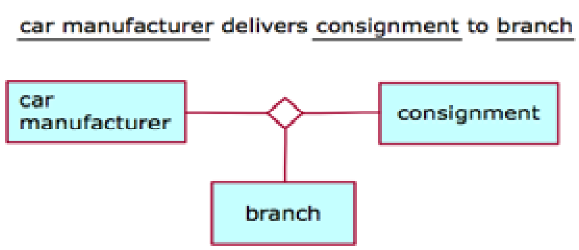


Figure H.7- Depicting a fact type with arity of three

H.3.4 Unary Fact Types

UML associations only apply to binary and higher-arity. Ordinarily a unary fact type is transformed into a UML Boolean attribute, as shown in Figure H.8.

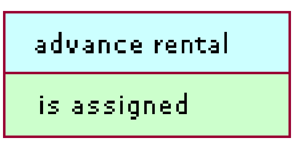


Figure H.8- Depicting the unary fact type 'advance rental is assigned' as a Boolean attribute

However, the SBVR unary fact type is more accurately modeled in UML using an alternative style, which applies the same conventions described in subclause F.3.3, adapted for the unary case shown in Figure H.9.

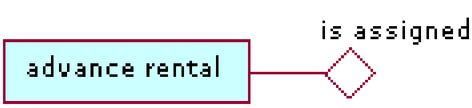


Figure H.9- Depicting the unary fact type 'advance rental is assigned' using association notation

H.4 Roles

Note that a 'role' in SBVR is a concept in its own right.

H.4.1 Role depicted as an Association End Name

A term for a role is typically shown as an association end name. Multiple appearances of the same role name coming into the same class imply a more general 'role' concept as well as the specific roles shown.

NOTE: Figure H.10 shows two fact type forms ~~forms of expression~~ for the same fact type (see also subclause H.3.2).

speech community *uses* vocabulary

vocabulary *has* audience

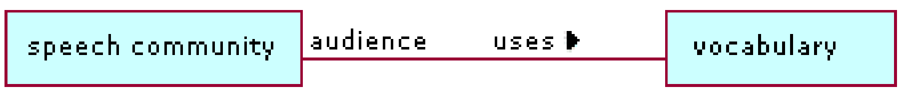


Figure H.10- Depicting a role as an association end name

Issue 9948 Revise text

H.4.2 Role depicted using UML Stereotyping

Since a 'role' in SBVR is a concept in its own right it can also be depicted as a class (rectangle), with UML stereotyping used to denote the ~~specialization~~ object type that it ranges over. As illustrated in Figure H.11, the stereotype <<role>> can be reflected for the class or the generalization line can use the stereotype <<is-role-of>>.

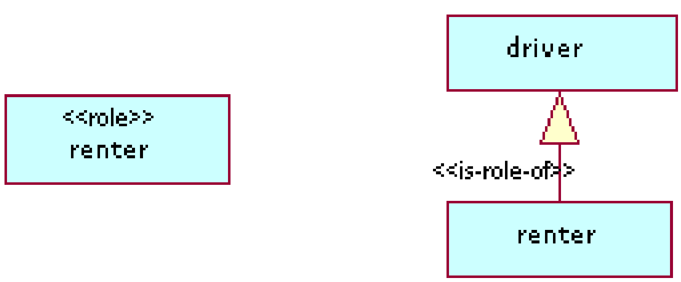


Figure H.11- Depicting a role as a class, with stereotyping

Issue 9958 Change form of expression with fact type form
Issue 10422 Revise text
Issue 9930 Revise text

H.4.3 Term for a Role in a Fact Type Form ~~Form of Expression~~

When a term for a role is used in a fact type form ~~form of expression~~, and that form is not an attributive form (e.g., "a has b"), then the term for the role needs to be shown. It is not shown as an association end because that would imply an attribute form (e.g., "has"). Instead, the term for the role is ~~shown in square brackets~~ underlined and shown, along with the verbal part of the fact type form. ~~form of expression as shown in Figure H.12~~.

Figure H.12 gives an example. In the fact type “rental incurs late return charge” (from EU-Rent), ‘late return charge’ is a term for a role -- the general concept is ‘penalty charge’. Rather than put “incurs” on the association line connecting “rental” to “penalty charge,” the text on the line incorporates the term for the role and reads, “incurs ~~[late return charge].”~~ late return charge.”

~~In the first example: for the fact type “concept incorporates characteristic”, ‘characteristic’ is a term for a role -- the general concept is unary fact type. Rather than put “incorporates” on the association line connecting “concept” to “unary fact type”, the text on the line incorporates the term for the role and reads, “incorporates [characteristic]”. A second example (from EU-Rent) is given.~~

Issue 10422 Replace figure
Issue 9958   Revise figure caption

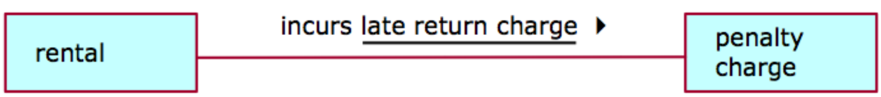


Figure H.12- Example of a term for a role in a ~~form of expression~~ fact type form

H.5 Generalizations

Generalizations are shown in the normal UML way as shown in Figure H.13.

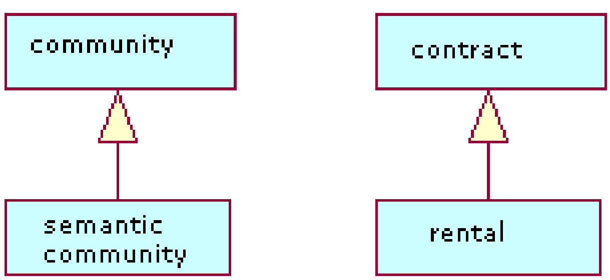


Figure H.13- Two examples of generalization

H.6 Categorization

H.6.1 Categories and Categorization Schemes

A set of mutually-exclusive categories can be depicted by bringing the generalization lines together, as shown on the left diagram in Figure H.14. Contrast that with the diagram on the right which reflects two independent specializations -- i.e., a community can be both a semantic community and a speech community.

Optionally, the name of a categorization scheme can be assigned to the set of categories, e.g., ‘Rentals by Payment Type’.

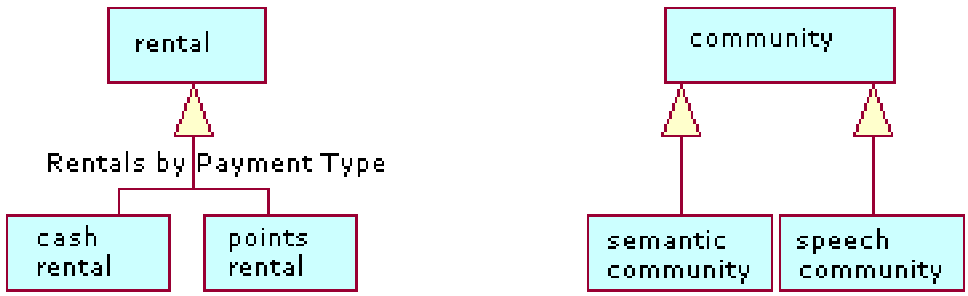


Figure H.14- Depicting mutually-exclusive categories vs. independent specializations

H.6.2 Categories and Categorization Types (Concept Types)

Use of UML powertype notation is not typical, but it can be used to show the categories specified by a categorization type (concept type). Note that the second diagram in Figure H.15 illustrates a named categorization scheme (‘Branches by Type’) which is related to the categorization type ‘branch type’.

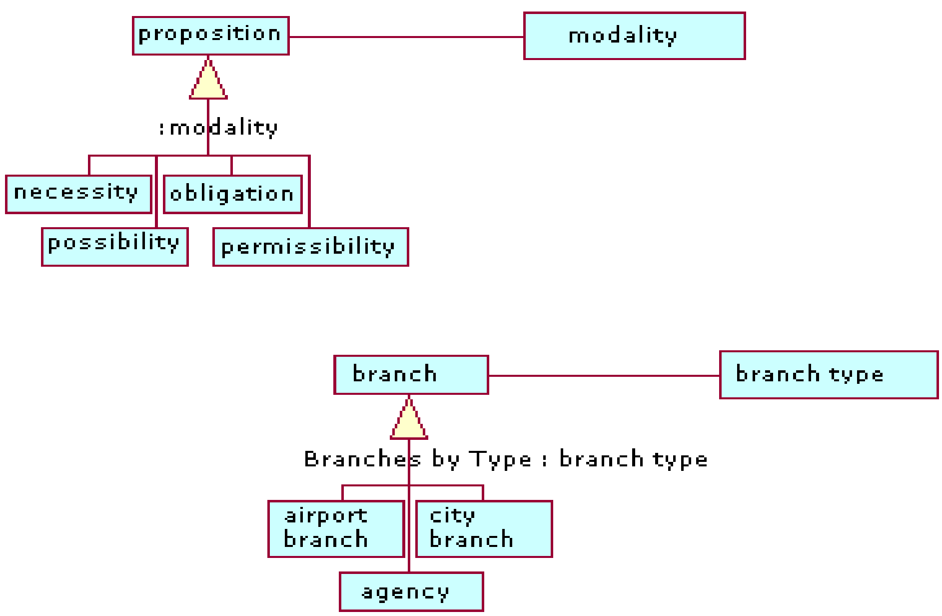


Figure H.15- Two examples of depicting the categories specified by a categorization type

H.7 Partitive Fact Type

UML aggregation notation is used to represent partitive fact types.

Issue 9958 Replace form(s) of expression with fact type form(s), replace Figure H.16
Issue 9941 Replace text and figure

The diagram on the left of Figure H.16 shows the fact type forms ~~forms of expression~~ ~~for the vocabulary-to-symbol and vocabulary-to-fact-type-form form-of-expression fact types~~ for the partitive fact types that 'body of shared meanings' is involved in.".

~~symbol *is included in* vocabulary~~

~~fact type form~~ ~~form of expression~~ ~~*is included in* vocabulary~~

body of shared meanings *includes* body of shared concepts

body of shared meanings *includes* body of shared guidance

~~The right diagram of Figure H.16 shows the fact type form form of expression for the partitive fact type between vocabulary and vocabulary, as follows:~~

~~vocabulary$_1$ *incorporates* vocabulary$_2$~~

The diagram on the left of Figure H.16 also illustrates the partitive fact type between 'body of shared meanings' and 'body of shared meanings', as follows:

body of shared meanings$_1$ *contains* body of shared meanings$_2$

Note that the subscripts in the fact type form ~~form of expression~~ are not reflected on the diagram.

As the diagrams of Figure H-16 illustrate, reflecting the verb phrase of a partitive fact type on the diagram is optional.

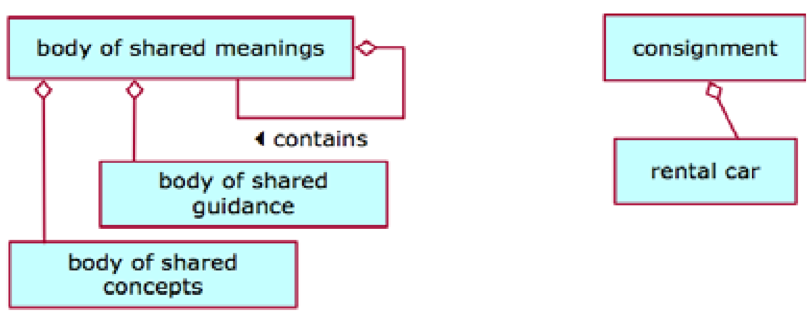


Figure H.16- Two examples of partitive fact type

H.8 Fact Type 'Objectification'

Where a noun concept is defined using objectification such that it is coextensive with a fact type, an association class is used to depict the noun concept, as shown in Figure H.17. A dashed line connects the association line for the fact type with the box for the noun concept. A binary fact type is shown in a similar fashion, with the dashed line connecting to the binary association line.

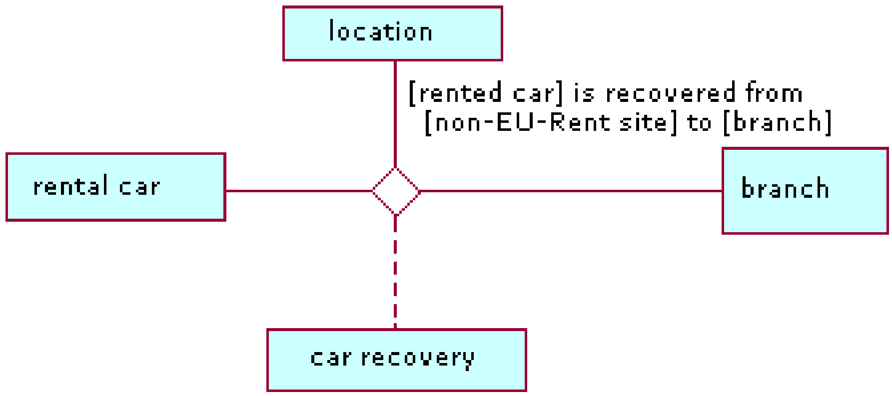


Figure H.17- Depicting fact type 'objectification'

H.9 Multiplicities

Multiplicities are typically not shown.

However, display of UML multiplicity is a diagram-level option. When UML multiplicity is used on a diagram (as a whole), this element is used to depict a formally-stated alethic necessity of a particular multiplicity. UML multiplicity is used for no other case. In a diagram that uses UML multiplicity, the default assumption for an unannotated association end is '*' (which is interpreted as '0 or more' -- i.e., unconstrained).

Annex I
(informative)

The ORM Notation for Verbalizing Facts and Business Rules

Annexes C and F discussed how to verbalize facts and business rules in the SBVR Structured English and RuleSpeak notations. This annex briefly presents a third approach to verbalization that is based on *Object-Role Modeling* (ORM) [Halp1998, Halp2001], a conceptual modeling approach that has been used productively in industry for over 30 years. While this approach has been localized to other languages (including Japanese, German and French), we restrict ourselves here to the English version.

Business rules may be specified in ORM using graphical and/or textual languages. We confine ourselves here to just part of ORM's textual language. We regard a *static business rule* to be a constraint or derivation rule that applies to each individual state of the business, taken one state at a time. This annex focuses on the *verbalization* of static business rules, ignoring dynamic rules relating to state transitions or workflows. In the interests of brevity, only a few of ORM's rule verbalization patterns are illustrated here, mainly using examples from the EU-Rent case study. A detailed discussion may be found in the references [Halp2003a, Halp2003b, Halp2003c, Halp2003d, Halp2004c, Halp2004d, Halp2004e, Halp2004f, Halp2004g, Halp2004b].

Issue 9948 Revise text

I.1 Criteria for Business Rule Verbalization in ORM

Static business rules are best applied to a fact model that identifies the fact types of interest to the business. Table I-1 shows some fact types with arities from 1 to 4. Each fact type role corresponds to an object placeholder (depicted here as an ellipsis "…") in the predicate. Here predicates are displayed in *mixfix* notation, allowing object terms to be placed in a sentence at any position. Higher arity predicates (quinary, etc.) are also possible.

Table I.1 - Examples of fact types of different arity

Fact Type	Predicate	Arity
Person smokes	… smokes	1 (Unary)
Person was born in Country	… was born in …	2 (Binary)
Person played Sport for Country	… played … for …	3 (Ternary)
Person introduced Person to Person on Date	… introduced … to … on …	4 (Quaternary)

The ORM textual language for verbalizing fact instances, fact types, and business rules is based on the following criteria:

- *expressibility* - the language is able to express a wide range of business rules
- *clarity* - the rules are understandable by non-technical domain experts
- *flexibility* - the language directly supports predicates of any arity
- *localizability* - the language constructs are expressible in different native languages

- *formality* - the rules are unambiguous, and should ideally be executable

Apart from its graphical language, ORM uses a textual language that is both formal and conceptual, so that it can serve for communication and validation with domain experts, as well as being executable. Relevant dimensions used in ORM for rule verbalization are listed in Table I.2, along with the choices available. For detailed discussion of these criteria, see the references.

Table I.2 - Classification schemes for rule verbalization

Dimension	Choices
Form	Positive Negative Default
Modality	Alethic Deontic
Style	Relational Attribute Mixed
Context	Local Global
Formality	Informal Semiformal Formal

ORM's verbalization language applies to mixfix predicates of any arity, with predefined patterns to cater for a very wide range of constraints found in business domains. Unlike some other approaches, ORM leaves the verbalization of the underlying fact model unchanged (e.g., no need to pluralize noun phrases and related verb phrases).

Every constraint has an associated *modality*, determined by the logical modal operator that functions explicitly or implicitly as its main operator. In practice, the modality is typically either *alethic* or *deontic* (see Table I.3). Logical negation may be used to obtain the usual equivalences (e.g., not necessary ≡ possible, not obligatory ≡ permitted, not permitted ≡ forbidden).

Table I.3 - Alethic and deontic modal operators

Alethic	Deontic
It is necessary that	It is obligatory that
It is possible that	It is permitted that
It is impossible that	It is forbidden that

The next two subclauses present some simple examples. Far more complex examples may be found in the references.

Issue 9948 Revise text

I.2 Some Basic Rule Examples in ORM

Simple uniqueness constraint:

Positive form:

Each rental car *is owned by* at most one branch.

In positive verbalizations, the modality is often assumed (as above), but may be explicitly prepended ("It is obligatory that" for deontic modality; "It is necessary that" for alethic modality).

Negative form, deontic modality:

It is forbidden that the same rental car *is owned by* more than one branch.

Negative form, alethic modality:

It is impossible that the same rental car *is owned by* more than one branch.

Composite uniqueness constraint:

Positive, deontic form of a uniqueness constraint over two fact type roles from the ternary fact type room *at* hour slot *is booked for* course.

It is obligatory that

given any room and hour slot

that room *at* that hour slot *is booked for* at most one course

Composite Exclusion constraint:

Relational style: No person *directed* and *reviewed* the same movie.

Attribute style: For each movie:

no director is a reviewer.

Join Subset constraint:

Each advisor who *serves in* a country

also *speaks* a language that *is used by* that country.

Derivation Rule:

Relational style: Define person_1 *is an uncle of* person_2 as

person_1 *is a brother of* person_3 who *is a parent of* person_2

Attribute style: For each person: uncle = brother of parent.

Issue 9449 Revise text

I.3 Some EU-Rent Rule Examples

~~This section provides restatements in ORM of some EU-Rent rules presented in earlier annexes. The ORM rewording is displayed after the original formulation, assuming that the fact types used in the verbalization are defined in the model.~~

~~It is obligatory that a rental have a car group.~~

It is obligatory that each rental *has* a car group.

~~It is obligatory that a rental car with a service reading greater than 5000 miles be scheduled for service..~~

It is obligatory that each rental car that *has* a service reading greater than 5000 miles *is scheduled for service.*

~~It is obligatory that in an international return the country of the local area of the receiving branch be the country of registration of the rental car.~~

It is obligatory that
if a rental car *is in* an international return that *is to* a receiving branch
that *is in* a local area that *is in* a country
then that rental car *is registered in* that country.

~~It is permitted that a renter books multiple rentals.~~

It is permitted that each renter *books* more than one rental.

I.3 EU-Rent Examples in ORM

This subclause provides restatements in ORM of the EU-Rent examples presented in SBVR Structured English (Annex E.1.4) and in RuleSpeak (Annex F). The ORM rewording is displayed after the SBVR Structured English formulation, assuming that the fact types used in the ORM verbalization are defined in the model.

Conventions used:

- Object types are bold and underlined.
- Verb phrases are bold.
- Components of constraints are in italics.

- Articles and referents are unadorned.
- The terms “may” and “must” indicate deontic modalities permission and obligation, respectively.
- The term “might” (as in #9) indicates alethic possibility; lack of any modal term (as in #1) indicates alethic necessity.
- The term “which” is used to provide proper English syntax to avoid ending with a preposition; the preposition immediately preceding “which” actually terminates a verb phrase in the model.

1	It is necessary that each rental *has exactly one* requested car group.
	Each **rental** **requests** *at least one* **car group**. *Each* **rental** **requests** *at most one* **car group**. - or, combined: *Each* **rental** **requests** *exactly one* **car group**.

Guidance Type: structural business rule

2	It is obligatory that the rental duration of *each* rental *is at most* *90 rental days*.
	It must be that each **rental** **lasts** *at most 90* **rental days**.

Guidance Type: operative business rule

3	It is obligatory that each driver of *a* rental *is qualified.*
	It must be that each **driver** **that drives** a **rental** **is qualified at** the **date/time** **at** which that **rental** **actually started**.

Guidance Type: operative business rule

4	It is obligatory that the rental *incurs a* location penalty charge if the drop-off location *of* a rental *is not* the EU-Rent site that *is base for* the return branch *of* the rental.
	It must be that a **rental** **incurs** a **location penalty charge** *if* the **rented car** **of that** **rental** **is dropped off at a location that is *different* from the** **EU-Rent site** **where the** **return branch** **of that** **rental** **is based.**

Guidance Type: operative business rule

Note not expressible using standard ORM constraint notation

5	It is obligatory that the rental charge *of* a rental *is calculated in* the business currency *of* the rental.
	It must be that a **rental charge** that **is incurred by** a **rental** **is calculated in a** **business currency** **that is used by that** **rental**.

Guidance Type: operative business rule

6	It is permitted that a rental *is open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.
	It may be that a **rental is open** *only if* **an estimated rental charge** that **is incurred by** that **rental is provisionally charged to** a **credit card** that **is held by the customer** that **acquires** that **rental.**

Guidance Type: operative business rule

Note: not expressible using standard ORM constraint notation.

7	It is obligatory that the local area that *includes* the return branch *of* an in-country rental or international inward rental *owns* the rented car *of* the rental *at* the actual return date/time *of* the rental.
	It must be that the **local area** that **includes** the **return branch** that **is** the **destination of an in-country rental** *or* an **international inward rental owns** the **rental car** that **is assigned to** that **rental at** the **date/time at which** that **rental is returned**.

Guidance Type: operative business rule

Note: not expressible using standard ORM constraint notation.

8	It is obligatory that *at* the actual pick-up date/time *of* each rental the fuel level *of* the rented car *of* the rental *is* full.
	It must be that the **rental car** that **is assigned** to a **rental has** a **fuel level** *equal* **to '***full***' at** the **date/time at** which that **rental actually started.**

Guidance Type: operative business rule

Note: not expressible using standard ORM constraint notation.

9	It is possible that the notification date/time *of* a bad experience that *occurs during* a rental *is after* the actual return date/time *of* the rental.
	It might be that the **notification of** a **bad experience** that **occurs during** a **rental is received at** a **date/time** that *is greater than* the **date/time at which** that **rental is actually returned**.

Guidance Type: advice of possibility

Note: not expressible using standard ORM constraint notation; however, possibilities are implied by the absence of other constraints - especially necessities - that preclude them.

10	It is permitted that the drop-off branch *of* a rental *is* not the return branch *of* the rental.
	It may be that a **rental is dropped off at** a **different branch** than the **branch to which** that **rental is to be returned**.

Guidance Type: advice of permission

Note: implied by the model, as is (no equality constraint is specified, therefore it is permitted).

Annex J
(informative)

ORM Examples Related to the Logical Foundations for SBVR

J.1 Introduction

This annex provides some detailed examples to illustrate how foundational concepts described in Clause 10.1.1 can be captured in an existing logic-based approach. The examples use Object-Role Modeling (ORM), which has a well-defined mapping to formal logic [Halp1989]. A basic introduction to ORM may be found in [Halp2000] and a detailed treatment in [Halp2001]. ORM takes a fact-based approach to modeling business scenarios that is compatible with the SBVR approach.

J.2 Simple Database Example

Figure J.1 shows an ORM schema for the simple Employee/Car database example discussed in Clause 10.1.1. In ORM, *objects* are either *entities* (non-lexical objects that are identified by definite descriptions, and that typically change state) or *values* (lexical constants that identify themselves, such as character strings). In ORM 2 (the latest version of ORM, used here), entity types and value types are depicted as named, soft rectangles with solid or dotted lines respectively (previous versions of ORM used ellipses instead of soft rectangles). Logical *predicates* are depicted as named sequences of role boxes, where each *role* is a part played in the relationship. For binary fact types, if forward and inverse predicate readings are displayed on the same side of the role boxes, they are separated by a slash "/". By default, predicates are read left-to-right and top-to-bottom.

A large dot on a role connector indicates that the attached *role is mandatory* (i.e., for each state of the fact model, each instance in the population of the object type must play that role). The object type's population in the fact model is not necessarily the same as the real world population in that state, and is typically far smaller than the extension of the object type (which covers all possible states). For example, each employee has an employee name, but it is optional whether an employee drives a car.

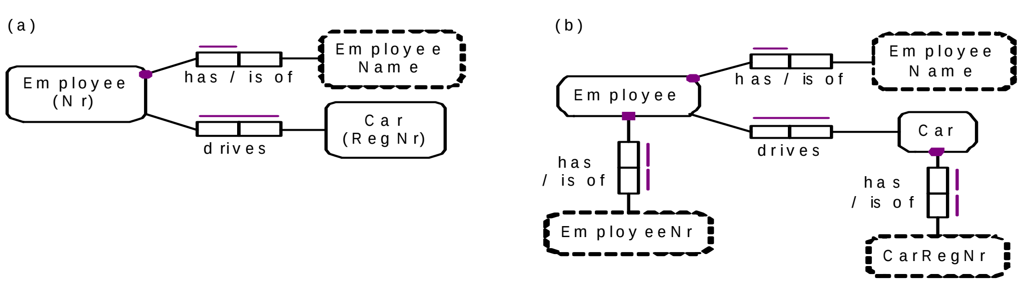


Figure J.1 - ORM schema for the simple Employee/Car database example

A bar beside a role box depicts a *uniqueness constraint*, indicating that for each state of the fact model each object that instantiates that role does so only once. For example, each employee has at most one employee name. A bar that spans two or more roles depicts a uniqueness constraint over that role combination, indicating that for each state of the fact model each object sequence that instantiates that role sequence does so only once. For example, the fact type Employee drives Car is many:many, and in each state any instance of this fact type appears at most once.

Figure J-1(b) displays simple injective (mandatory, 1:1 into) reference schemes explicitly as binary relationships. Employees are referenced by their employee numbers, and cars by their registration numbers. Figure J-1(a) displays these reference schemes compactly as parenthesized reference modes.

J.3 Open/Closed World

Consider the populated unary fact type in Figure J.2(a). For simplicity, we omit reference schemes, and assume people may be identified by their first names. We know that Fred smokes. If we use open world semantics, then it is unknown whether Sue or Tom smoke. If the ORM schema is mapped to a UML class, then the open world interpretation leads to an optional isSmoker attribute with only one possible value ('Y' for yes), as shown in Figure J.2(b). If we apply closed world semantics, then the absence of facts that Sue or Tom smoke entails that they don't smoke; this leads to a mandatory, Boolean isSmoker attribute, as shown in Figure J.2(c).

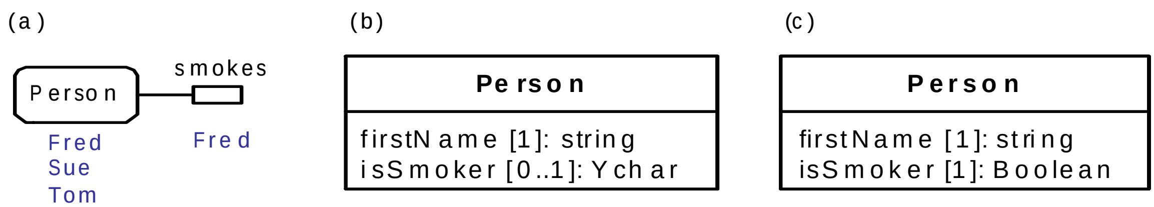


Figure J.2 - An ORM model (a), and UML classes based on (b) open world and (c) closed world semantics.

Currently most ORM tools adopt the closed world assumption for unaries. However for the next generation of ORM tools that are designed to interoperate with SBVR tools, it is anticipated that unaries will be treated as open by default.

For many fact types in a business domain, especially those without functional roles, it is impractical to include all the negative instances as base facts. For example, for the fact type Employee drives Car, there might be many thousands of cars, so one would normally not explicitly include negated facts such as Employee 1 does **not** drive Car 'AAA246'. In some cases however, especially with functional roles or when the population is small, it is practical to include negated facts as base facts.

Figure J.3 shows two ways to model a business domain where for each person in the population of the domain it is known whether that person smokes or not. In each case, negated facts are explicitly treated as base facts, and the predicates are given open world semantics. Semi-closure is implied because of the constraints. In Figure J.3(a) the xor constraint (circled mandatory dot overlaid by 'X" for exclusion) declares that each person referenced in the fact model population plays exactly one of the two roles (smoking or not smoking). In Figure J.3(b) the mandatory, uniqueness and value constraints collectively ensure the same thing. When either of the ORM schemas is mapped to a UML class, a mandatory Boolean isSmoker attribute results, as shown in Figure J.3(c).

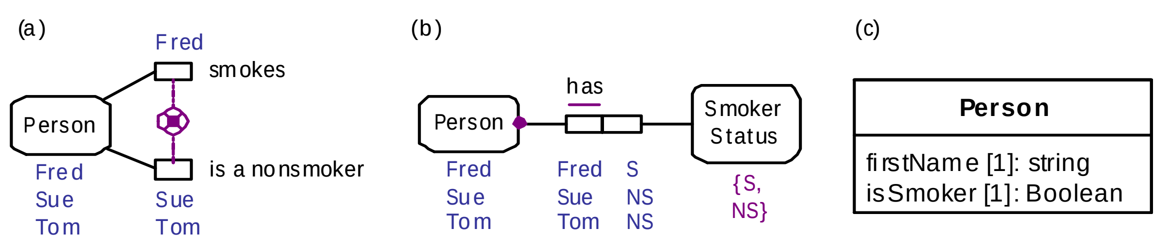


Figure J.3 - Open world semantics plus negated facts and constraints that ensure semi-closure.

Now consider a business domain where we know that Fred smokes, and that Sue doesn't smoke, but are unsure whether Tom smokes. To model this at all, we need open world semantics. Figure J.4 shows three ways to model this in ORM, as well as the equivalent UML class. Figure 4(a) uses an exclusion constraint, Figure J.4(b) uses an optional binary, and Figure J.4(c) uses a mandatory binary and a special value (here shown as "?") to indicate that the smoking status is unknown. We treat this special value like any other value, using 2-valued logic, rather than adopt a generic null based on 3-valued logic (as in SQL). The equivalent UML class notation is shown in Figure J.4(d).

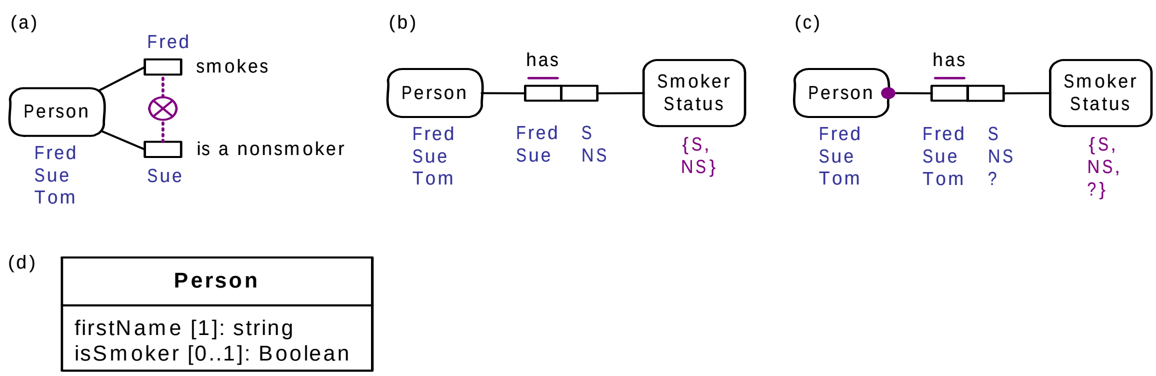


Figure J.4 - We may indicate whether a person smokes or not, or that this is unknown.

J.4 Deontic Constraints

In the ORM schema shown in Figure J.5, the fact type Person is a husband of Person is declared to be many to many, as shown by the alethic, spanning uniqueness constraint over the top of the predicate. In addition a deontic uniqueness constraint has been added (depicted by a bar starting with an "o" for "obligatory") to each role to indicate that the fact type *ought* to be 1:1. The leftmost deontic constraint verbalizes as: **It is obligatory that each** Person is a husband of **at most one** Person. The other deontic constraint (each wife should have at most one husband) may be handled in a similar way.

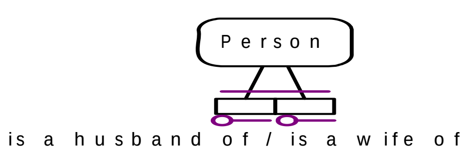
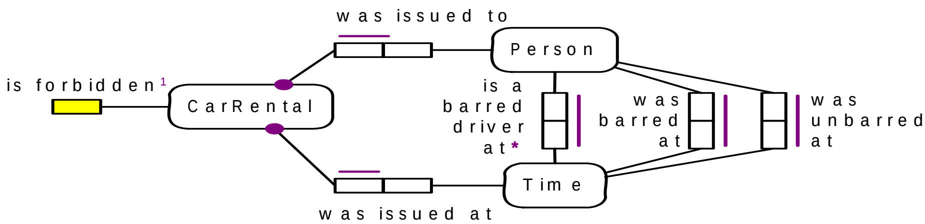


Figure J.5 - Deontic constraints obligate the marriage relationship to be 1:1.

The deontic constraint "Car rentals ought not be issued to people who are barred drivers at the time the rental was issued." may be captured by the textual constraint on the domain fact type CarRental is forbidden, as shown in the ORM schema in Figure J.6. The fact type Person is a barred driver at Time is derived from other base fact types (Person was barred at Time, Person was unbarred at Time) using the derivation rule shown.

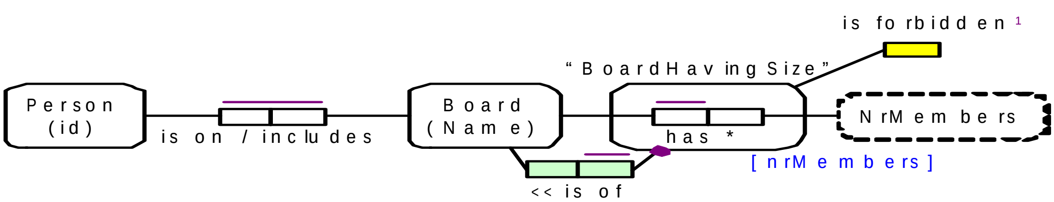


*Person is a barred driver at Time$_1$ **iff**
Person was barred at **a** Time$_2$ <= Time$_1$ **and**
Person was **not** unbarred at **a** Time$_3$ **between** Time$_2$ **and** Time$_1$

[1] CarRental is forbidden **if**
CarRental was issued at Time **and**
CarRental was issued to Person **and**
Person is a barred driver at Time

Figure J.6 - Specifying a deontic constraint forbidding rentals to barred drivers using a domain level predicate.

The deontic constraint “It is forbidden that more than three people are on the EU-Rent Board.” is captured by the textual constraint on the derived fact type BoardHavingSize is forbidden in the ORM schema shown in Figure J.7. The derivation rule is stated in attribute style, but its underlying relational style is used in invoking the derivation rule within the body of the deontic constraint.

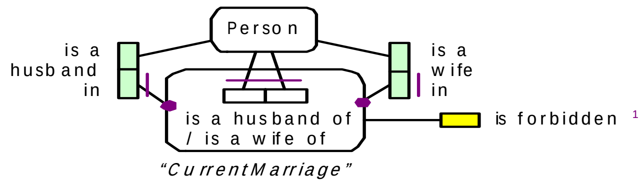


* nrMembers **of** Board = **count each** Person **who** is on Board

[1] BoardHavingSize is forbidden **if**
BoardHavingSize is of **a** Board
that has BoardName ‘EU-Rent Board’
and has NrMembers > 3

Figure J.7 - Specifying a deontic constraint on the size of the EU-Rent board using a domain level predicate.

The deontic constraints that require each person to have at most one spouse may be formulated as textual constraints on the fact type CurrentMarriage is forbidden, as shown in Figure J.8.

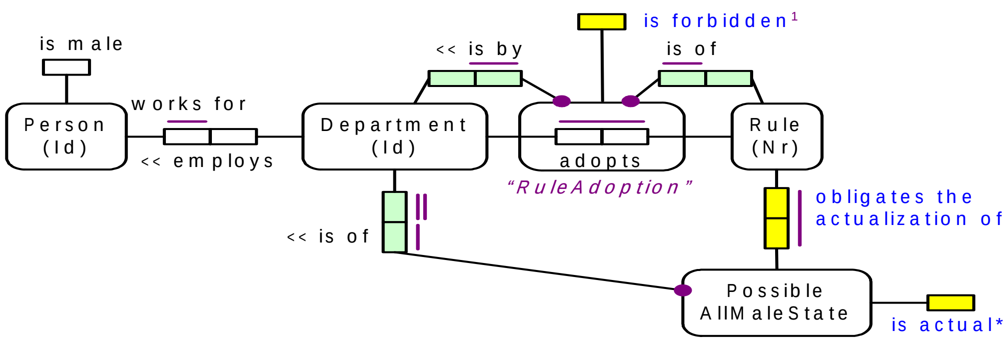


[1] CurrentMarriage is forbidden **if**
a Person$_1$ **who** is a husband in CurrentMarriage
is a husband of **more than one** Person$_2$.
CurrentMarriage is forbidden **if**
a Person$_1$ **who** is a wife in CurrentMarriage
is a wife of **more than one** Person$_2$.

Figure J.8 - An alternative way to capture the deontic constraints in Figure 5.

The ORM schema in Figure J.9 relates to the following deontic constraint: “It is not permitted that some department adopts a rule that says it is obligatory that each employee of that department is male.” This example includes the mention (rather than use) of an open proposition in the scope of an embedded deontic operator. The schema uses the special predicates “obligates the actualization of” and “is actual,” as well as an object type “PossibleAllMaleState” which includes all conceivable all-male-states of departments, whether actual or not.

The formalization of the deontic constraint works, because the relevant instance of PossibleAllMaleState exists, regardless of whether or not the relevant depart actually is all male. The “obligates the actualization of” and “is actual” predicates embed a lot of semantics, which is left implicit. While the connection between these predicates is left informal, the derivation rule for PossibleAllMaleState is actual provides enough semantics to enable human readers to understand the intent.



* PossibleAllMaleState is actual **iff**
PossibleAllMaleState is of **a** Department **and**
each Person **who** works for **that** Department is male

[1] RuleAdoption is forbidden **if**
RuleAdoption is by **a** Department
and is of **a** Rule
that obligates the actualization of **a** PossibleAllMaleState
that is of **the same** Department

* $\forall x$:PossibleAllMaleState
[x is actual $\equiv \exists y$:Department (x is of y & $\forall z$:Person (z works for y $\supset$ z is male))]

[1] $\forall x$:RuleAdoption
[$\exists y$:Department $\exists z$:Rule $\exists w$:PossibleAllMaleState
(x is by y & x is of z & z obligates the actualization of w & w is of y)
$\supset$ x is forbidden]

aliter:

[1] $\forall x$:RuleAdoption $\forall y$:Department $\forall z$:Rule $\forall w$:PossibleAllMaleState
[(x is by y & x is of z & z obligates the actualization of w & w is of y) $\supset$ x is forbidden]

Figure J.9 - A complex case involving embedded mention of propositions

Annex K
(informative)

Mappings and Relationships to Other Initiatives

K.1 Mapping to Other Standards and Metamodels

K.1.1 For Rule Representation

Issue 9721 Replace text
Issue 9955 Add text

There are several existing metamodels for representing rules. Only OWL has become a standard. The scopes of these metamodels differ from SBVR. The discussion in this subclause gives an overview of the most well-known, their use and characteristics.

- No standard is yet widely used on a commercial basis.
- With respect to rules, these metamodels focus on representation. SBVR focuses on unique, discrete meanings independent of form or representation.
- SBVR includes a Formal Model theory and semantic formulations. There is only partial coverage in other metamodels.
- Uniquely, SBVR provides necessity and obligation ~~claims~~ formulations, which are critical to the formal representation of business rules.
- SBVR places special emphasis on obligation ~~claims~~ formulations. In the real world of business activity, people can break such operative business rules, a crucial fact other metamodels do not address.

It is possible to create transformations from SBVR to any of the metamodels or vice versa. Any of the transformations, especially those moving from information systems specifications back to SBVR may require manual input to provide missing semantics or to transform decisions not automatable.

Development of transformations should consider the following points:

Transformation from SBVR to the other metamodels

- A decision should be made how to treat necessity and obligation ~~claims~~ formulations. One option is to translate these to predicates.
- Some of the non-SBVR representations do not have an equivalent operator for the 'whether or not' and 'equivalence' operators.
- Some of the non-SBVR representations do not have equivalent operators for quantifiers like 'each', 'some', 'at least one', etc. In that case might be possible to create special predicates or functions to deal with this semantics.

Transformation from other metamodels to SBVR

- The non-SBVR representations can have primitive types or primitive functions that do not exist in SBVR. By extending the SBVR Vocabularies with an additional vocabulary, one can create a mapping from another metamodel to the extended SBVR. SBVR is self-extensible.

Metamodels at the Business Level Used to Talk about Real Business Things – Optimally Conceptualized to be 'Business Friendly' for Business People

Name	Type	Developed by	Used by	Form	Reference
N458 Topic Map Constraint Language	Proposed Standard	ISO/IEC	Topic Maps	document	http://www.isotopicmaps.org/tmcl/tmcl-2005-02-12.html

Topic Map Constraint Language is designed to allow users to constrain any aspect of the topic map data model. TMCL adopts TMQL [Topic Map Query Language] as a means to express both the topic map constructs to be constrained and topic map structures that must exist in order for the constraint to be met.

Development of transformations should consider the following points:

Transformation between SBVR and Topic Map Constraint Language

- The only transformation required, in addition to the generally applicable ones mentioned above, would be where semantics conceptualized into SBVR metamodel constructs differently from the way it is conceptualized into metamodel constructs in Topic Maps as they both talk about real business things in business friendly terms.

Metamodels that can be Used at the Business Level Used to Talk about Real Business Things – Optimally Conceptualized for Logicians and/or Machine Processing Efficiency

Name	Type	Developed by	Used by	Form	Reference
24707 Common Logic	Proposed Standard	ISO	KIF, CGIF, XCL, PSL	Document	www.iso.org
OWL	Standard	W3C	Semantic Web	DTD or XML schema	www.w3c.com

ISO Common Logic is a first order logic language for knowledge interchange. It provides a core semantic framework for logic and the basis for a set of syntactic forms (dialects) all sharing a common semantics. **ISO Common Logic** can also be used at the Information System Specification level to talk about information and information system components as it is a general-purpose first-order predicate logic standard.

OWL is a Web Ontology language. Where earlier languages have been used to develop tools and ontologies for specific user communities (particularly in the sciences and in company-specific e-commerce applications), they were not defined to be compatible with the architecture of the World Wide Web in general, and the Semantic Web in particular.

OWL uses both URIs for naming and the description framework for the Web provided by RDF to add the following capabilities to ontologies:

- Ability to be distributed across many systems
- Scalability to Web needs

- Compatibility with Web standards for accessibility and internationalization
- Openness and extensibility

Issue 9955 Replace text

OWL builds on RDF and RDF Schema and adds more ~~vocabulary for describing~~ ways to describe properties and classes: among others, relations between classes (e.g., disjointness), cardinality (e.g., “exactly one”), equality, richer typing of properties, characteristics of properties (e.g., symmetry), and enumerated classes.

Development of transformations should consider the following points:

Transformation from SBVR to the above standards

- In general, formal logic-based entries in SBVR-based conceptual schemas and models will be transformable into ISO Common Logic or into OWL.

Transformation from the above standards to SBVR

- Any ISO Common Logic sentences and Owl entries that can be expressed in ISO Common Logics that
  - talk about real business things (and not data about real business things or information system buckets that hold such data), and
  - are limited to the SBVR ‘restricted higher order logic’

  can be transformed into SBVR if the semantic equivalences of different representations and different semantic formulations are provided by the transformation as these are not kept track of in ISO Common Logic.
- Some contents of SBVR-based conceptual schemas and models which do not have counterparts in OWL or ISO Common Logics might need to be provided manually.

Metamodels that Specify Information Systems at the PIM/PSM Levels

Name	**Type**	**Developed by**	**Used by**	**Form**	**Reference**
13211 Prolog	Standard	ISO		document	www.iso.org
Production Rules Representation	Proposed Specification	OMG		XMI	www.omg.org
RuleML	Metamodel	Consortium (see reference)	Mandarax, the website contains a list of 40 participants (mostly academics)	DTD	www.ruleml.org
SWRL	Metamodel	DAML		XML Schema	www.daml.org

Proprietary Metamodels					
Name	**Type**	**Developed by**	**Used by**	**Form**	**Reference**
RBML	Metamodel	LibRT	VALENS, Artis, Power	XML Schema	www.librt.com

SRML	Metamodel	Ilog	Ilog Jrules	DTD	www.ilog.com
SRL	Metamodel	Fair Isaac	Blaze Advisor	DTD	www.fairisaac.com
BRML	Metamodel	IBM	IBM CommonRules	XML Schema	www.ibm.com

Development of transformations should consider the following points:

Transformation from SBVR to the above metamodels

- Alignment of SBVR with the above metamodels requires a transform from SBVR whose entries talk about real things in the business to specifications of data about the real business things, and the design specifications for the buckets used to store that data within various components of the information system.

Transformation from the above metamodels to SBVR

- Requires the (re-)introduction, probably manually, of whatever business semantics (or pointers to them) are not within the information systems specifications.

K.1.1.1 For Vocabulary Representation

Today there are several standards and models for representing a vocabulary. It must be noted, however, that none of these provides an adequate extension to formal logics to fully support business rules. The following list gives an overview of the most well-known, their use, and characteristics:

Metamodels at the Business Level Used to Talk about Real Business Things – Optimally Conceptualized to be 'Business Friendly' for Business People

Name	Type	Developed by	Used by	Form	Reference
1087-1, 704-2000, 10241, & 12620 Terminology	Standard	ISO		document	www.iso.org
17115 Health Informatics -- Vocabulary for Terminological Systems	Standard	ISO/DIS		document	www.iso.org
2788 & 5964 Thesaurus	Standard	ISO		document	www.iso.org
12620 & 13250-2 Topic Maps	Standard	ISO		document	www.iso.org
Public Domain De Facto Industry Standard					
ORM	Metamodel	Terry Halpin, et al	Microsoft, Case talk, Infagon		www.orm.net www.demo.nl www.mattic.com/ Infagon.html

- SBVR is based on the **ISO standards 1087-1 and 704-2000** for terminology and information science. These standards describe a methodology but do not provide a product metamodel that can be used to store and interchange business vocabularies.

- **Health Informatics -- Vocabulary for Terminological Systems** supplements the ISO 1087-1 and 704-2000 standards to provide a more formal structuring of terminology. From the standard: "The purpose of this International Standard is to define a set of basic concepts required to describe formal concept representation systems, especially for health sciences, and to describe representation of concepts and characteristics, for use especially in formal computer based concept representation systems. A main motivation is to make it possible to precisely describe content models described in other International Standards."

- **ISO 2788 & 5964 Documentation - Guidelines for the establishment and development of monolingual/ multilingual thesauri** is about creating indexes for books and other documents by identifying the subjects or topics (concepts) discussed in the document. From the standard: "The recommendations set out in this International Standard are intended to ensure consistent practice within a single indexing agency, or between different agencies (for example members of a network)."
- **ISO/IEC 13250 Topic Maps** is about Topics (Concepts) and connections between them (Facts). From the standard: "This International Standard provides a standardized notation for interchangeably representing information about the structure of information resources used to define topics, and the relationships between topics. A set of one or more interrelated documents that employs the notation defined by this International Standard is called a topic map."
- **ORM** is a modeling method originally intended for database design. SBVR is highly influenced by the way ORM defines and verbalizes fact types and facts. Transformations between a vocabulary of SBVR and ORM tools can be established although not all SBVR concepts have an equivalent in ORM.

Development of transformations should consider the following points:

Transformation between SBVR to the above standards

- With the following exceptions the only transformation required, in addition to the generally applicable ones mentioned above, would be where the semantics in SBVR was conceptualized into metamodel constructs differently from the way it was conceptualized into metamodel constructs in Topic Maps as they both talk about real business things in business friendly terms.
  - (except for ORM) none of the above standards are based on formal logics so there will need to be some manual decisions going from them to SBVR. The other direction should be automatic except for constructs not in the SBVR metamodel.
  - Constructs in SBVR not in those standards would be lost
- ORM is very similar to SBVR so that two-way transformations are minimal. However, SBVR is more comprehensive than ORM so some semantics would be lost going from SBVR and have to be provided manually going to SBVR.

Metamodels at the Business Level Used to Talk about Real Business Things – Optimally Conceptualized for Machine Processing Efficiency

Name	Type	Developed by	Used by	Form	Reference
RDF(S)	Standard	W3C		DTD or XML schema	www.w3c.com
OWL	Standard	W3C		DTD or XML schema	www.w3c.com

- **RDF(S)** not only talks about real business things but also contains pointers (URLs) to the storage locations where information about those business things is kept. Thus RDF(S) also includes mappings across the transformation from the Business Level to the Information System specification levels. RDF(S) does provide a metamodel that can be used to store and interchange business vocabularies. It is expected that lossless bidirectional transformations between SBVR and RDF(S) can be established.
- *OWL Web Ontology Language* is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL has more facilities for expressing

meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

Development of transformations should consider the following points:

Transformation from SBVR to the above standards

- All and only formal logic-based entries in SBVR will be able to be transformed into RDF(S) and/or OWL because both are also mapped to ISO Common Logic.

Transformation from the above standards to SBVR

- Any RDF(S) and Owl entries that can be expressed in ISO Common Logics that
  - talk about real business things (and not data about real business things or information system buckets that hold such data), and
  - are limited to the SBVR 'restricted higher order logic'

  can be transformed into SBVR if the semantic equivalences of different representations and different semantic formulations are provided manually as these are not kept track of in ISO Common Logic.
- All of the vocabulary related entries not part of the SBVR formal logics subclause will have to be provided manually.

Issue 9955 Add text

K.1.1.2 Standards for database and system modeling

Today there are several standards and models for representing a database or a systems object model. Most well known and widely used are UML for Object Oriented models and Entity Relationship diagram for relational databases.

A vocabulary that is developed using SBVR may contain representations of concepts and fact types that should also be represented in a database or object model. For those concepts a transformation to UML or Entity Relationship diagrams can be created. Be aware that the SBVR model and a PIM level UML model or ER diagram have a different perspective. That is the reason that not all elements of SBVR may be relevant in a PIM perspective and the PIM model may need to be augmented after a transformation from SBVR.

Metamodels that Specify Information Systems at the PIM/PSM Levels

Name	Type	Developed by	Used by	Form	Reference
11179 Metadata Registry	Standard	ISO/IEC		document	www.iso.org
UML	Specification	OMG		document	www.omg.org
Entity Relationship (CWM)	Specification	OMG		document	www.omg.org

[See Subclause 1.9.3.3.1 for description of SBVR mapping to PIM standards, specifications, and models.]

- **ISO/IEC 11179 - Metadata Registries (MDR)**, addresses the semantics of data, the representation of data, and the registration of the descriptions of that data. It is through these descriptions that an accurate understanding of the semantics and a useful depiction of the data are found. The purposes of ISO/IEC 11179 are to promote the following:
  - Standard description of data
  - Common understanding of data across organizational elements and between organizations
  - Re-use and standardization of data over time, space, and applications

- Harmonization and standardization of data within an organization and across organizations
- Management of the components of data
- Re-use of the components of data

- **The Unified Modeling Language™ - UML** is an OMG (Object Management Group) specification for modeling application structure, behavior, and architecture.
- **Entity Relationship – CWM (Common Warehouse Metamodel)** The Common Warehouse Metamodel (CWM™) is a specification that describes metadata interchange among data warehousing, business intelligence, knowledge management, and portal technologies. Entity Relationship (ER) models are used frequently as a means of describing business processes and the data on which they operate. Because of its importance as a design and tool model, the CWM includes a foundational ER model from which individual tool models may derive their specific extensions. Doing so will improve the extent to which ER models can be interchanged between various tooling environments.

Development of transformations should consider the following points:

Transformation from SBVR to the above standards

- Inputs to this transformations are:
  - An extract from an SBVR model that fits the scope of the application software to be designed
  - Additional Business requirements for the application software
- The transformation is effectively the design process of the class-of-platform independent PIM model. It includes, among others, such design choices as:
  - Design of generalized data storage structures e.g., Hierarchies, Data-driven generalizations
  - Class / Attribute / Association / Association Class decisions
  - One Concept of a Business Thing implemented in two Attributes
  - Store vs. derive decisions
  - Design of time constructs
  - 'State' implementation design decisions
  - Surrogate Keys design choices

Transformation from the above standards to SBVR

- This is a reverse engineering transformation which is made possible only by adding back in, as part of the transformation process, any SBVR business semantics that were not stored with the model when it was created, and maintained since then.

Issue 9955 Replace text

K.1.1.3 Standards for Business Modeling Vocabularies + Rules

There are a number of standards that provide vocabularies and rules for subjects commonly used to specify in business models the way businesses are to be operated. These standards can be imported into the SBVR metamodel to become general-purpose SBVR Business Vocabularies+Rules.

- Country names and codes (ISO/IEC 3166)

- Dates and times (ISO/IEC 8601)
- Currency codes (ISO/IEC 4217)
- Addresses (ISO/IEC 11180)
- Information and documentation (ISO/IEC 5127)
- Business Agreement Semantic Descriptive Techniques (ISO/IEC 15944)
- Process Specification Language (ISO 18629 series of standards)
- …many others from ISO and other standards bodies

In turn these general-purpose SBVR ~~can be adopted by business-specific~~ can be incorporated into business-specific vocabularies.

K.1.2 Use of UML Notation in a Business Context to Represent SBVR Vocabulary Concepts

UML Notation can be used to represent SBVR-based vocabularies. Details of the mapping of SBVR concepts to UML Notation are provided in Annex H.

K.1.3 Reuse of other OMG Standards

This SBVR specification reuses the MOF 2.0, XMI 2.1, and the UML 2 Infrastructure for its model repository and for interchange of SBVR Vocabularies and rules.

K.1.4 Relationship of SBVR with other OMG RFPs

Issue 9955 Add text

K.1.4.1 SBVR and Business Modeling

SBVR is only one of several BEIDTF initiatives in the business modeling arena. Others include:

Business Process Definition Metamodel (BPDM)

The revised submission deadline for RFP responses is in August 2005.

SBVR and BPDM are complementary. SBVR specifies the meaning and representation of Business Vocabulary and Rules. BPDM specifies the use of Business Vocabulary and Rules by various BPDM model elements.

The primary relationship of SBVR and BPDM is the roles Business Rules play in a BPDM. The definition of the relationship between Business Concepts, Business Facts and Business Rules in SBVR and the various model elements in BPDM is scheduled to be called for in a separate RFP, the adoption of whose response will integrate both SBVR and BPDM.

Secondarily, SBVR can be used to provide formal logics-based definitions for all the model elements in BPDM (see subclause 1.9.4.2).

Organization Structure Metamodel (OSM)

The initial submission deadline for RFP responses is in November 2005.

Business Rules Management (BRM)

The RFP is being drafted.

SBVR is about the meaning and representation of Business Vocabulary and Rules and only that. BRM focuses on all other information about Business Vocabulary and Rules needed to effectively manage and use them to operate the business and as part of information system requirements. SBVR provides the Business Vocabulary and Rules that are managed by BRM. BRM manages the contents of SBVR.

Business Motivation Model (BMM)

The Business Rules Group (BRG) has been encouraged to submit its Business Motivation Model: *Business Governance in a Volatile World* [BMM][1] to the BEIDTF under the OMG's Request for Comment (RFC) process. This model addresses business goals, strategies, and policies.

SBVR and BMM are complementary. SBVR adopts the BMM definition of Business Policy, and BMM adopts the SBVR definition of Business Rule.

SBVR and Need for Integration among Business Modeling Specifications

These BEIDTF developments are related. For example, BPDM and SBVR have strongly related central concepts:

- From the BPDM perspective, Business Rules deliver 'factored out', flexible detail to support Business Processes.
- From the SBVR perspective, Business Processes provide the specific contexts in which Business Rules need to be evaluated. (In a PIM view, this might mean 'fired' or 'triggered', for example.)

Whether the BRG's Business Motivation Model is accepted or not, the BEIDTF will need a metamodel for its domain. Business processes are better defined when a business knows where it wants to go (its goals and objectives), and what it needs to do to get there (its strategies, tactics, and policies). Business processes realize the strategies and tactics. Business rules realize the business policies, and both support and constrain the business processes.

Business processes, supported by business rules, are associated with organization roles and structure. Some business rules apply to organization structure and roles, independently of processes.

There is clearly a need for integration. This has been recognized. For example, the BPDM submission included 'hooks' for business rules and organization roles.

Need for a Common Vocabulary

An important first step towards integration is to ensure a common vocabulary. Within a business, 'customer' and 'product' should mean the same everywhere they are intended to be the same, no matter what aspects of the business people are discussing or defining -- processes, rules, organizational responsibilities, locations, etc.

It is suggested that the BEIDTF consider *integration by adoption*, a loose coupling of metamodels by adoption of concepts and terms. This would mean:

- Shared concepts would be defined once in an 'owner' standard, and adopted by other standards as 'users'
- Benefits would be consistency across standards and reduction of replication
- The implication would be that when an 'owner' standard is revised, all the 'users' have to be considered (note: this would be a good thing!)

1. The BRG released version 1.0 in 2000, entitled *Organizing Business Plans*.

Concepts could also be adopted from outside the OMG; for example, this specification for SBVR adopts from ISO, standard dictionaries, and other authoritative sources.

What is important for OMG Business Modeling Standards is to ensure a shared body of meanings, largely by use of accepted vocabularies and diligent examination of the similarities and differences in the vocabularies of BEI and ADPTF standards. By definition, there are different communities and contexts involved, and the ~~term-concept~~ signifier-concept relationships may be different. Synonyms and homonyms need to be recognized, and definitions brought up to a formal logics quality.

For the Future – A Common Vocabulary Model?

This specification for SBVR incorporates a well-developed approach to vocabulary development. The SBVR view is that the concepts should be consistent across the business, and the terms used for them should be unambiguously understood. This includes management of synonyms, homonyms, and resolution of ambiguity by providing contexts.

This is important for practical application of SBVR to real businesses. People in different operational areas, in different geographical locations and in parts of businesses that have been merged or acquired, will use their familiar terminology. They can be encouraged into standard terminology, but they cannot be forced. Major customers, partner organizations, outsourcers, and trade groups will also share concepts, even if they use different words.

This need to support this is not specific to business rules. It is relevant to all types of business description, from mission statements to scripts for help lines.

A next step from *integration by adoption* across OMG business modeling standards would be to create a common metamodel for business vocabularies. If the BEIDTF decided to do this, it would be reasonable to propose a subset of the SBVR model as a candidate. The part of the model that supports concepts, fact types, and vocabularies has been separated from the business rules part, and can be reused to support other aspects of business modeling.

K.1.4.2 SBVR and Platform Independent Modeling

As discussed above, the SBVR standard should be integrated with other OMG standards for business modeling. This would help ensure that coherent business models are developed and supported consistently with tools and methodologies based on these standards.

Such business models (or substantial parts of them) will be used as bases for specification of information system models. In MDA, this would require mappings and transformations from a business model to a Platform Independent Model (PIM).

Mapping to a PIM

The current MDA practice is for a PIM to be defined using UML models. Two kinds of transformation will be used:

- From business concepts (including fact types) to a UML class model. Some concepts will map to classes, others to attributes. Some fact types will map to associations in the class model. Some structural business rules will map to constraints on cardinality, optionality, and mutual exclusion.
- From business rules to operations and constraints in the UML models formed from business concepts. There are several possible approaches for this, and further investigation would be needed.

The transformation of business rules would provide only part of a PIM, which would also support transformed content from other business model aspects, including business process, user interfaces and workflow. This reinforces the case made above for a common business vocabulary model.

See Subclause M.1.1.2 for adopted PIM non-OMG standards and OMG Specifications.

Other submissions for SBVR

Other submissions for SBVR have presented PIM-oriented metamodels that would support a rule-based approach more directly than the general mapping to PIM described above.

They are based on extensions of UML such that many types of business rule (as described in this specification) could be expressed in OCL. Two kinds of transformation would be required:

- From a subset of the business vocabulary to a UML class model, as described above.
- From a subset of business rules to OCL, using the vocabulary of the class model. Additional guidance would be needed for types of business rules that would not map directly.

This is an important piece of the architectural jigsaw, especially with regard to transformation to a Platform-Specific Model (PSM), and the BEIDTF might consider issuing another RFP to address it.

Production Rule Representation

The BEIDTF has issued an RFP for Production Rule Representation. The RFP requests a model and XML interchange format for rules executed in an inference engine. Initial responses were submitted in August 2004, and the proposers have since agreed to collaborate on a joint proposal.

Production rules have the general form “if condition, do action,” and would use the vocabulary of a PIM’s class model. They may be grouped into rule sets that can be invoked en bloc.

Business rules in this SBVR specification have the declarative form “the following proposition should/must always be true,” and use a business vocabulary.

As with other approaches for mapping to PIM, a UML class model consistent with the business vocabulary is assumed. Some transformations from declarative business rules to production rule form are already well-understood at the level of individual rules, but substantial work will be required to develop a full mapping that includes making all conditions and actions explicit, and grouping rules into rule sets.

Ontology Definition Metamodel

As well as BEIDTF initiatives, SBVR is also related to the Ontology Definition Metamodel (ODM), which is being developed in response to an RFP issued by the OMG Analysis and Design Task Force.

The OMG Ontology Definition Metamodel (ODM) intends to provide an integrated family of metamodels for a variety of knowledge representation techniques, to assist in defining and interchanging ontologies, with a key objective of supporting semantic technologies. Most of the metamodels in this family reflect the abstract syntax of an existing standard, rather than inventing a new representation paradigm. The term “ontology” refers to a machine-processable representation of knowledge, particularly for automated inferencing. In general, the audience for the ODM is the developers of inference engines, tools that capture and prepare ontologies for inference engines from other declarative forms, such as UML models and structural business rules, and tools that convert ontologies into other forms of implementation model. A key concept in ontologies is that knowledge is “monotonic”: Over time we can add to our knowledge, but we won’t learn anything that contradicts something we already know for sure, so that knowledge from multiple sources can be combined.

The ODM is being developed concurrently with SBVR. The draft proposed ODM includes metamodels of several popular knowledge representation languages, with mappings between them.

The draft proposed ODM as of October 2004 includes proposed MOF metamodels for:

- Resource Description Framework Schema (RDFS – W3C Recommendation),
- Web Ontology Language (OWL – W3C Recommendation),

- ISO Common Logic (CL, defined in ISO 24707),
- Topic Maps (TM – ISO 13250),
- Unified Modeling Language (UML), and
- Description Logic (DL).

For business rules, monotonic logic is only applicable to a small fraction of the concerns. In many areas, the business is not interested in what is true for all time, but rather in what is true now and may change in the next hours or days. And in some cases, it is the objective of certain business rules to change currently true but unfavorable situations into future favorable situations. So there is a significant difference in the purposes of these standards. And this gives rise to significant differences in the interpretation of the logic models. (In the Semantic Web work, the distinction is made between class-based reasoning, which matches a subset of SBVR structural rules capabilities and is monotonic, and instance-based reasoning, which deals with actual facts about specific objects and may not be "safe" for monotonic reasoning.)

To handle operative business rules, which involve obligations and permissions, SBVR supports logic elements beyond those of the ODM languages, including CL. Lossless bi-directional transformations between the SBVR rules metamodel and the ODM metamodels are not guaranteed. A partial mapping between SBVR and the ODM metamodels could be developed. With proper care, SBVR could be used in ontology development.

ISO is considering extending CL to include modal and other logics and is planning a natural language surface syntax for CL. Both of these ISO initiatives may be important to SBVR and ODM in the future, but they are out of scope for the current ODM and SBVR work.

Annex L
(informative)

A Conceptual Overview of SBVR and the NIAM2007 Procedure to Specify a Conceptual Schema

L.1 Introduction

The acceptance of SBVR is a breakthrough in productivity in requirements and knowledge management. It is fundamentally a fact oriented approach, which makes it comprehensible to many people. It so happens that experience with this approach started in Europe in the seventies, and a mature business practice has been developed during the last 35 years. The current version of this practice is called NIAM2007 [Nijs1977, Nijs1978, Nijs1980, Nijs1986, Nijs2006].

In this annex we will primarily concentrate on describing the coherence of the essential concepts of SBVR, using the NIAM2007 methodology, and thus providing the reader with an easy to grasp framework for SBVR. NIAM2007 uses fact type diagrams that combine the advantages of diagrams and natural language statements, by integrating diagrammatic and natural language aspects.

A small part of the EU-Rent example of Annex E will be used to build up, step by step, an understanding of a well-selected subset of the SBVR core concepts and how they interrelate (i.e., their coherence).

For communication purposes we start with a concrete example, which, in the framework of Figure L.1 is at the level called ‘Fact population.’ From there we move systematically via the domain-specific component of the conceptual schema to the generic component. This is another useful direction compared to clause 8, 9 and 10 and especially appreciated by people new to the subject.

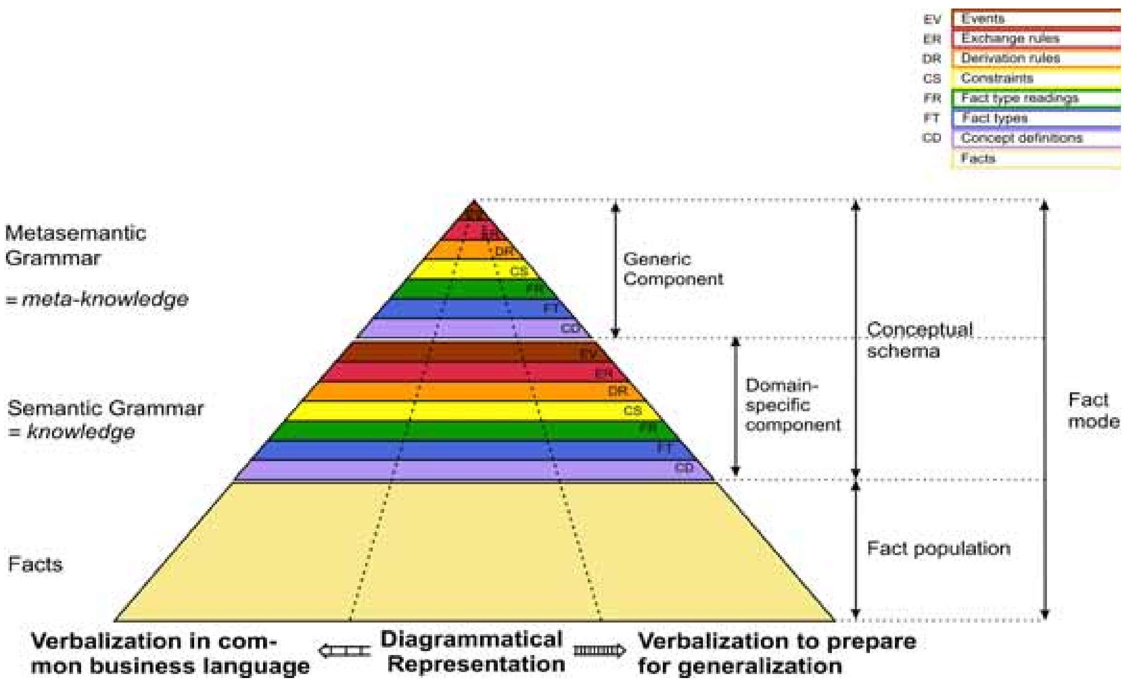


Figure L.1 - Knowledge triangle

The knowledge triangle of Figure L.1 represents the core concepts described in sub clause 10.1.1.2 and a few other concepts, specific for NIAM2007, in diagrammatical coherence.

The knowledge triangle is divided into three vertical lanes. The middle lane represents diagrammatical representations of structured knowledge. The left lane represents verbalizations for business people, which represent the same knowledge in a way familiar to business people. The right lane represents the NIAM2007 'verbalizations to prepare for generalization,' which represents the same knowledge as perceived from the perspective to derive the next level. The latter form of verbalization allows for generalization, which is a step in the procedure of deriving the next level. This will be shown in detail in the following sections.

The knowledge triangle is divided into three levels of knowledge or facts:

I. Facts: facts without a grammatical function, called ground facts in 10.1.1.2, e.g., 'The operating country Germany uses the business currency Euro,' 'The operating country France uses the business currency Euro,' and 'The operating country USA uses the business currency USD.'

II. Semantic Grammars: facts with a domain-specific grammar function, called the domain-specific component of the conceptual schema in 10.1.1.2, e.g., 'For each country that is recorded, its currency must also be recorded' and 'Each country name recorded in this fact type has to be unique (i.e., the only occurrence of that name).'

III. Metasemantic Grammar: facts with a generic or meta grammar function, called the generic component of the conceptual schema in 10.1.1.2, e.g., 'Each fact type must have a role or a sequence of roles for which uniqueness is required.'

In the following sub clauses of this annex, understanding of these concepts will be built up step by step. It will be shown that level II contains the rules and concept definitions for ground facts, and that level III contains meta-rules i.e., rules for rules, including the meta-rules themselves as well as the relevant concept definitions. Thus, level III describes itself. Therefore, these three levels suffice for describing knowledge.

The triangle was chosen as the form to represent structured knowledge to show that there are always more ground facts than rules for them and more level II (domain-specific) rules than meta-rules. This is the intent of defining rules: rules about knowledge are made to make working with knowledge more productive.

In the knowledge triangle the domain-specific as well as the generic component of the conceptual schema are divided in seven related knowledge classes:

- Concept definitions
- Fact types
- Fact type readings (also known as sentential forms)
- Constraints
- Derivation rules
- Exchange rules
- Events

These knowledge classes are part of SBVR as well as NIAM2007, except for exchange rules and events, which are not part of SBVR. Why are they in the knowledge triangle? To facilitate respectful discussions with other communities, such as UML. In the following sub clauses of this annex, all of these knowledge classes will be explained, except exchange rules and events, which fall outside of the scope of this annex. Of course, the concept of 'fact' will also be thoroughly explained.

SBVR is a major step forward towards widespread application of semantics in business and education. SBVR is the first specification in business computing where concept definitions are first class citizens. The concept definitions form the bridge between the formal and the informal world, hence are vital for business communication. One of the 7 knowledge classes at the domain-specific and the generic level, Concept Definitions, form the basis for each of the conceptual schemas, the domain-specific component and the generic component. As various annexes put the major emphasis on rules, this annex puts major emphasis on concept definitions, fact types and a useful variant of verbalization.

L.2 Use Case EU-Rent 1.1

A substantially reduced version of the EU-Rent Use case presented in Annex E is given below:

EU-Rent Use case 1.1

1. EU-Rent rents cars to its customers. Customers may be individuals or companies. It is obligatory
2. that the rental charge of a rental is calculated in the business currency of the rental.
3. this is a currency in which EU-Rent undertakes financial transactions. A rental has a business
4. currency, if and only if the business currency is the currency of the operating country of the
5. operating company that includes the local area that includes the pick-up branch of the rental.
6. The used business currencies are Euro (EUR), GBP (British Pound) and USD (United States
7. Dollar). Every country only uses one business currency.
8.
9. In each country in which it does business EU-Rent has an Operating Company. EU-Rent's current
10. operating countries are Canada, USA, France, UK, Ireland, Germany, Italy and Switzerland.

Regarding this use case, we first wish to focus on the domain-specific component of the conceptual schema and from there on a core part of the generic component of the conceptual schema to illustrate main concepts of SBVR. First of all, a sample graphical report has to be made regarding the different operating countries of EU-Rent and their respective currencies.

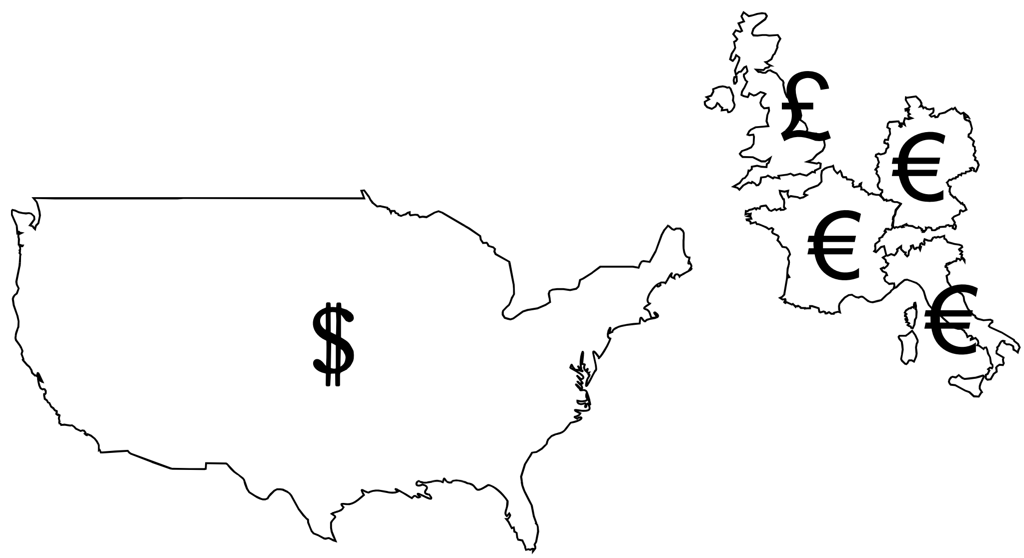


Figure L.2 - Graphical representation of operating countries and their business currency

Figure L.2 is a graphical representation of facts illustrating the use case described above. It is a sample report and could be called a data use case. Since the diagram represents actual facts satisfying the data use case, it is possible to verbalize the contents as if the business professional is talking to a colleague over the phone and writes down the textual representation of the represented facts. This is what is called “to verbalize a graphical representation.” Extensive experience with fact orientation during four decades has shown that starting with a concrete example of a data use case represented in the preferred notation of the business user is *the most productive* way to start requirements engineering and illustrate business processes. It is also recommended for knowledge elicitation.

If we ask a subject matter expert or business person to verbalize the given information of Figure L.2, the following sentences or facts will result:

Table L.1 - The result of verbalization by the domain expert

Operating country	Germany	uses the business currency	Euro
Operating country	UK	uses the business currency	GBP
Operating country	France	uses the business currency	Euro
Operating country	USA	uses the business currency	USD
Operating country	Italy	uses the business currency	Euro

As a first step towards a structured specification or conceptual schema, for every fact or sentence it is indicated where the variable and where the constant sentence elements are located. The result of this operation is presented in Table L.2.

Table L.2 - The result of assigning constant and variable parts

Operating country	Germany	uses the business currency	Euro
” ”	UK	” ” ” ”	GBP
” ”	France	” ” ” ”	Euro
” ”	USA	” ” ” ”	USD
” ”	Italy	” ” ” ”	Euro

As can be seen from the table above, there are two constant elements in each sentence, i.e., elements that are the same in each sentence (in this case “Operating country” and “uses the business currency,” respectively). There are two variable elements in each sentence, i.e., elements that have potentially different counterparts in the other sentences. The first variable element of each sentence is an example of an operating country and the second variable element of each sentence is an example of a business currency.

Like professionals in many other professions, extensive use is made of pattern recognition in the NIAM2007 methodology. From which fact type reading are the five listed facts an instance, an instantiation or a realization? We can conclude that we are dealing with sentences or facts that can be generated from the same fact type reading by filling in a value in two places, while the other elements consist of the same information for every sentence. The places in the fact type reading where the variable elements are to be filled in to form sentences, are called the ‘placeholders’ of the fact type reading in SBVR. By formulating a fact type reading it becomes possible to communicate the contents of a diagrammatical or report representation in a manner suited to a specific audience. The fact type reading which can be formulated based on the sentences listed in Tables L.1 and L.2, is given below and is assigned the number 1.

1: Operating country <Country> uses the business currency <Currency>.

This fact type reading has been derived by generalization of five example sentences, or facts. By filling in the placeholder <Country> with the name of an actual operating country (e.g., “Germany”), and the placeholder <Currency> with the name of a business currency (e.g., “Euro”), we obtain a concrete sentence or fact, in this case one of the sentences we started with.

Each placeholder has a counterpart in a fact type, and this counterpart is called ‘role’ in SBVR. This counterpart is shown in Figure L.3 in a diagrammatical form, using a NIAM2007 representation. In the diagrammatical representation of a fact type i.e., the fact type diagram, a role is represented by a rectangle containing the name of the role. This diagram also contains the fact type reading. In such diagrams, it is advised to include a sample population. In this case, five different pairs of variable elements are filled into the pair of roles, as population of the fact type.

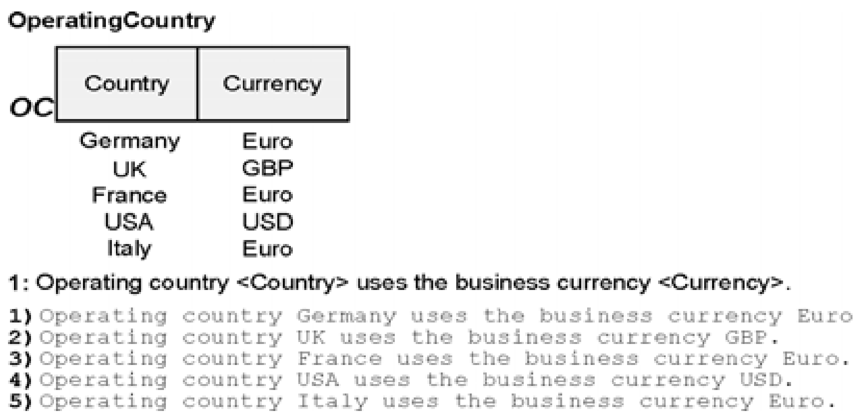


Figure L.3 - Provisional fact type diagram with population

Every fact instance verbalization is given a unique number followed by the symbol ‘)’. The fact instance verbalizations are generated, based on the values in the fact population of the fact type diagram OperatingCountry in Figure L.3.

Every fact type reading is given a unique number or code, in this case the number 1, within the domain-specific component of the conceptual schema. Every fact type is given a name, in this case OperatingCountry as well as a shorter code (here: OC) to facilitate communication.

The fact to be generated from the first record below the roles in the fact type diagram (which is the first record of the population) and the fact type reading can be read as follows: Operating country Germany uses the business currency Euro.

Regarding the structural understanding of the world (or the semantics) of these kinds of fact examples, at least the following terms have to be defined as concept definitions in this domain-specific conceptual schema:

Business currency
{Business currency} is a monetary entity in which EU-Rent undertakes financial transactions.

Operating country
{Operating country} is a country in which EU-Rent does business.

Above, we used a concrete graphical example in which the relevant facts are represented in a diagrammatical manner. We verbalized these diagrammatical representations of facts to get textual representations of the same facts. We made a start in transforming each textual representation into a domain-specific conceptual schema. Until now this transformation has resulted in:

1. two concept definitions;
2. one fact type diagram, as a possible representation of the fact type;
3. one fact type reading.

These three knowledge classes are only a part, although a very important basis, of the desired domain-specific conceptual schema. We therefore have to continue specifying the additional parts of the conceptual schema. We proceed in a structured way to the next part of the conceptual schema, the so-called constraints, a class of business rules.

What is a constraint? A constraint is a rule that limits the populations of the fact types and its population transitions, allowing only populations and transitions considered useful. According to NIAM2007, the most important constraint is the uniqueness constraint, which is illustrated in the following section. A uniqueness constraint corresponds to the set of independent variables of a function, a major concept in mathematics.

L.3 Uniqueness Constraint

To derive constraints, it is advised to use a precise process for systematic specification. As uniqueness constraints are the major constraints, we first derive these. The precise process ensures that all questions that need to be posed to a business domain expert are systematically composed and expressed in the familiar jargon of the business professional. The result of those processes leads to the following question to the subject matter expert in a language readily understood by the business domain expert:

Is it possible that the following two sentences can exist at the same time in the fact population?

Operating country Germany uses the business currency Euro.

and

Operating country Germany uses the business currency USD.

Or, as recommended by NIAM2007, are the contents of Figure L.4 acceptable to you?

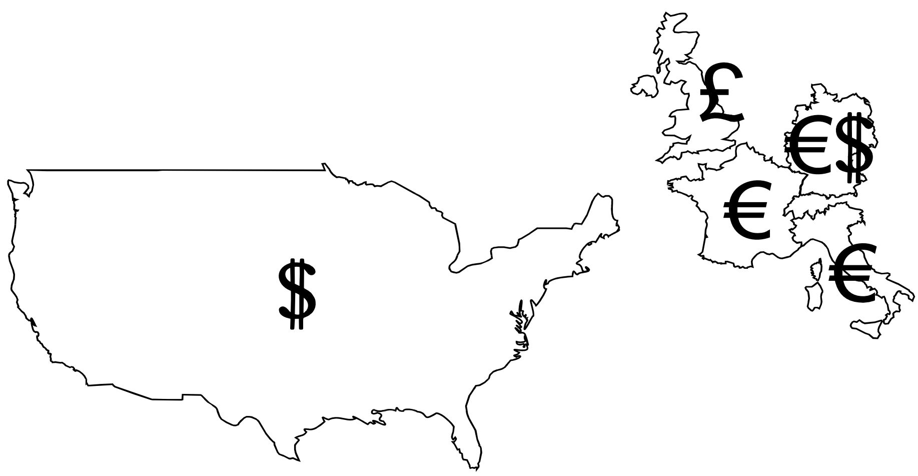


Figure L.4 - Concrete not permitted business example

The business domain expert will clearly say “No.” It is not allowed for an operating country to use two or more different business currencies as specified in line 7 of EU-Rent Use case 1.1. This answer is shown diagrammatically in the fact type diagram in Figure L.5 below.

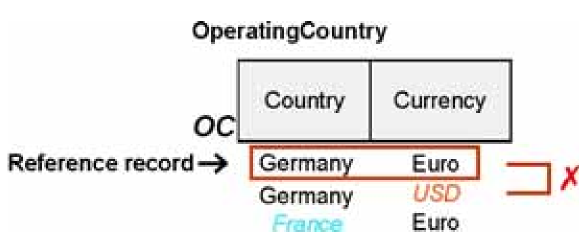


Figure L.5 - Matrix method for uniqueness: forbidden combination of records

Based on this particular answer from the business domain expert it is possible to conclude that the name of an operating country can only appear once in the fact population below the role Country of fact type diagram OperatingCountry. This results in the situation that the name of a country is unique within this fact population. In a fact type diagram in NIAM2007 a uniqueness constraint is indicated by a solid line with an arrow at both ends. In Figure L.6 below, the uniqueness constraint, arbitrarily named pk23 as an indication of primary key, is added to the fact type diagram OperatingCountry.

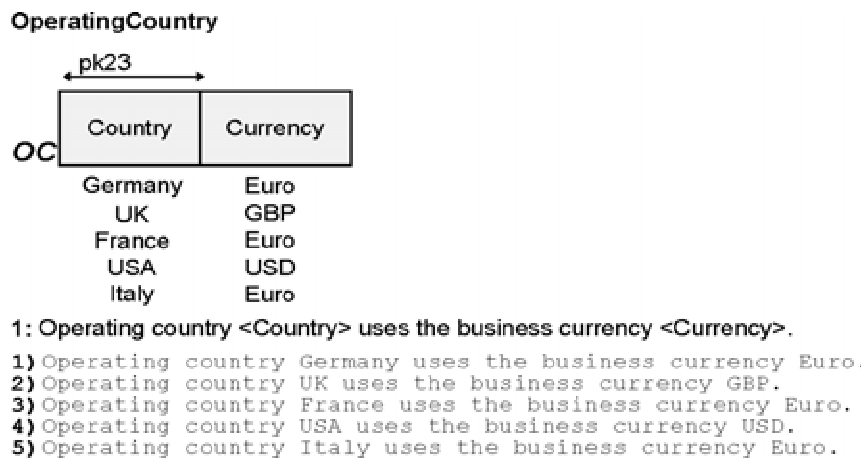


Figure L.6 - Fact type diagram OperatingCountry, after addition of uniqueness constraint pk23

What are the operational semantics of a uniqueness constraint? Every uniqueness constraint arrow means: below me in the fact population no duplicate values or signifiers can occur.

To know if a uniqueness constraint holds for the second role named “Currency,” one has to ask the business domain expert whether or not the following two facts can appear at the same time in the fact population:

Operating country Germany uses the business currency Euro.

and

Operating country France uses the business currency Euro.

The business domain expert will say “Yes, this was already clear in Figure L.2.” It is indeed possible that France as well as Germany use the same business currency; please note that this was represented in the data use case of Figure L.2. So the use of a specific business currency is not unique in this fact population. This implies that the values under the role “Currency” are not unique in this fact population and therefore no uniqueness constraint applies to this particular role of the fact type.

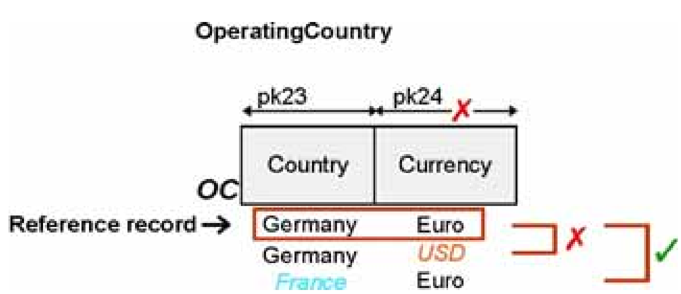


Figure L.7 - Intermediate result: no uniqueness constraint on Currency role

In Table L.3 an overview diagram of the procedure mentioned above is given.

Table L.3 - The register of answers given by the domain-specific expert

Country	Currency	Business domain expert’s answer regarding the simultaneous existence with the first record	Explanation
Germany	Euro		Record 1 (Reference record)
Germany	***USD***	No	It is not allowed that an operating country uses more than one different business currency.
France	Euro	Yes	It is possible that France as well as Germany both use the same business currency.

The use of easily recognizable symbols, like traffic-signs, makes communication about a conceptual schema more productive. In addition to the uniqueness constraint symbol introduced above, Figure L.8 below introduces the following symbols: the value rule symbol (curly brackets {} in a circle), the data type symbol (a character or character combination in the left upper corner of a role’s rectangle) and the non-empty rule symbol (a black rectangle in the right lower corner of a role’s rectangle). A value rule limits the values that can be used to fill in a particular role to a given list of possible values (listed at the bottom of the fact type diagram for representational purposes). In our use case we know that there is a limited number of operating countries for EU-Rent: Canada, USA, France, UK, Ireland, Germany, Italy and Switzerland. Others do not exist, thus should not be recorded, which is prescribed by value rule val5003. A data type limits the possible values of a particular role to values of a specified type. In the OperatingCountry fact type in Figure L.8, all values are allowed by the data type ‘c’ which is an abbreviation for ‘character’. A non-empty rule forbids a role to be left empty in a record. In the fact type OperatingCountry this is true for both roles, as it is not allowed to record an operating country without recording the corresponding business currency and vice versa. The latter is implied by the uniqueness constraint pk23 and the former is indicated by the non-empty rule symbol in the Currency role.

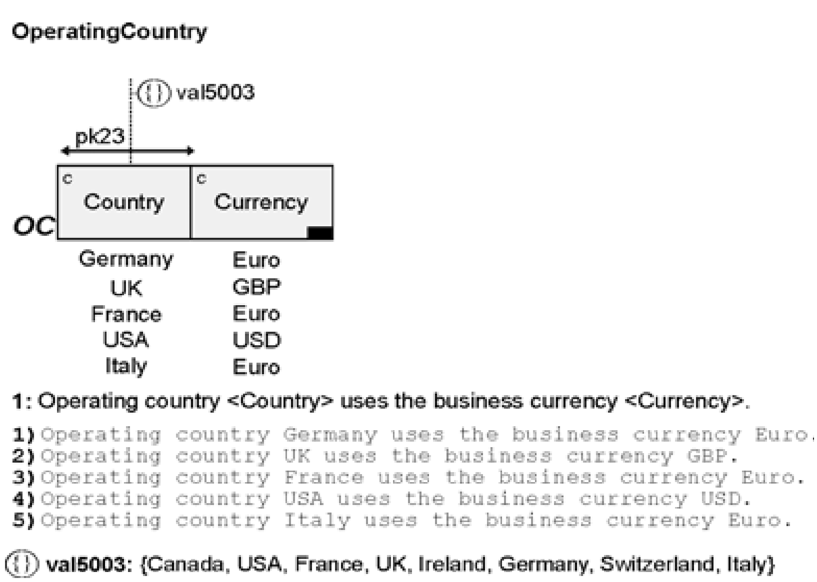


Figure L.8 - Complete fact type diagram OperatingCountry

L.4 LA Route to Time Invariant Knowledge

The previous sub clauses illustrated that we are able to verbalize the facts depicted in Figure L.2. From these verbalizations it was possible to derive fact type readings, placeholders, fact types, roles and some constraints (based on the questions systematically posed to the business domain expert). The business domain expert is proficient in answering questions in his own language or business vocabulary (“language that is readily understood by the business domain experts”, sub clause 10.1.1.2) in terms of permitted or not permitted concrete examples, a familiar world to the user. In other words, this was a trip from:

1. the level of ground facts (middle part of level I in Figure L.15), using verbalization to arrive at
2. the textual representation of the facts (right part of level I), then applying generalization to arrive at
3. the level II (middle part) of the domain-specific component of the conceptual schema in 10.1.1.2.

The next interesting question is: is it possible to verbalize these resulting diagrams of the conceptual schema with the aim to arrive at the next level? Let us apply verbalization to the fact type reading. Hence we treat the fact type reading in the middle part of level II in the same way we treated the middle part of level I. For a fruitful discussion we first provide a concept definition of ‘position.’

Position

A {Position} in a fact type reading may either consist of a constant piece of text until the first placeholder, or a constant piece of text between two placeholders, or a constant piece of text at the end of the fact type reading behind the last placeholder, or a position in a fact type reading is taken by an individual placeholder.

If we use this concept definition to analyze fact type reading 1, we get four positions, which are indicated below in the fact type reading by representing every character in the consecutive positions by the corresponding position number within the fact type reading:

```
1: Operating country   <Country>   uses the business currency   <Currency>

   11111111111111111   222222222   33333333333333333333333333   4444444444
```

Based on this we are able to give the verbalizations of a part (in this case a fact type reading) of the diagrammatical representation of the domain-specific component of the conceptual schema, see table L.4.

Table L.4 - Verbalization of a part of the domain-specific conceptual schema

Fact type reading	1	has in position	1	A	constant	with contents	Operating country
Fact type reading	1	has in position	2	A	variable	with contents	Country
Fact type reading	1	has in position	3	A	constant	with contents	uses the business currency
Fact type reading	1	has in position	4	A	variable	with contents	Currency

When we replace the constant sentence parts in every sentence after the first sentence in table L.4 by quotation marks, the result is as shown in table L.5:

Table L.5 - The result of assigning constant and variable parts

Fact type reading	1	has in position	1	a	constant	with contents	Operating country
" " "	1	" " "	2	"	variable	" "	Country
" " "	1	" " "	3	"	constant	" "	uses the business currency
" " "	1	" " "	4	"	variable	" "	Currency

From the example above it is straightforward to derive the fact type reading. It is decided to assign number 1000 to this new fact type reading:

1000: Fact type reading <FactTypeReading> has in position <Position> a <ConstOrVar> with contents <Contents>.

The resulting fact type diagram is given in Figure L.9.

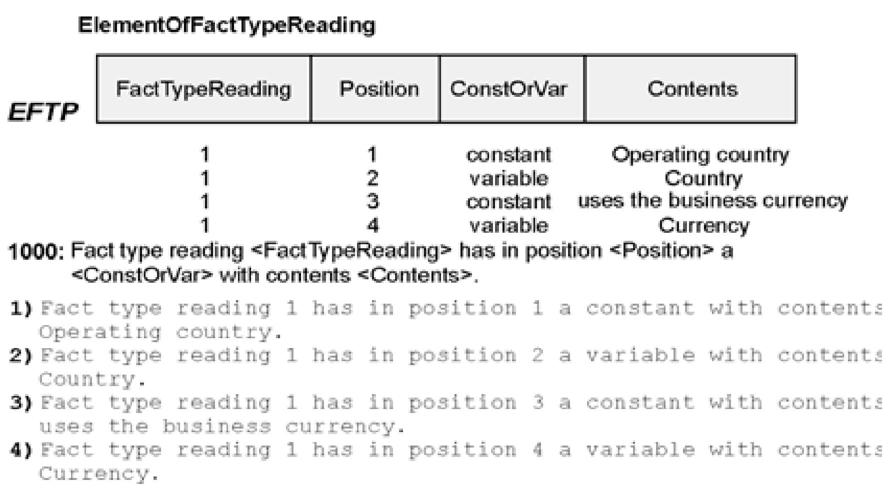


Figure L.9 - Meta fact type with population of domain-specific schema elements

Fact type reading 1000 is a rule which applies to all fact type readings, irrespective of the domain specifics. In other words, we have now arrived at the next level, the generic component of the conceptual schema. In NIAM2007 this is called the conceptual meta schema or Metasemantic Grammar (see Figure L.1). This component is topic independent.

Regarding the structural understanding of the world of these kinds of examples, at least the following concept definitions have to be provided:

ConstOrVar

{ConstOrVar} indicates whether or not an element in a [Fact type] or [Fact type reading] is variable or constant. In other words, whether or not an element in a [Fact type] or [Fact type reading] remains fixed or can be used to indicate a [Role] or [Placeholder] respectively.

Fact (Facts)

A {Fact} is a proposition that is taken as true.

Examples of facts are:

Within the class of all Member States of the United Nations the name	Austria	identifies a specific member state.
Within the class of all Member States of the United Nations the name	Belgium	identifies a specific member state.
Within the class of all Member States of the United Nations the name	Canada	identifies a specific member state.
Within the class of all Member States of the United Nations the name	The Netherlands	identifies a specific member state.
Within the class of all Member States of the United Nations the name	United States of America	identifies a specific member state.

These are five examples of unary, existential facts.

Existential facts need not be unary.

An example of a ternary existential fact is:

Within the class of all telephones in the EU the combination of country code 31, area code 45 and local number 5600222 identifies a specific telephone.

An example of an existential {Fact} as used in sub clause 10.1.1.2 is as follows:
There is a country that has the country code 'US'

Based upon a long experience in industrial fact oriented modelling we recommend to use the formulation:

Within the class of all <ConceptPlural> the <Signifier> identifies a specific <ConceptSingular>

An example of a unary non-existential fact is:
Bill Clinton smokes

Five examples of binary facts are:

Amsterdam	is the capital of	The Netherlands
Brussels	is the capital of	Belgium
Ottawa	is the capital of	Canada
Washington	is the capital of	United States of America
Vienna	is the capital of	Austria

Five examples of ternary facts are:

The Netherlands	entered the	EU	in	1957
The Netherlands	entered the	NATO	in	1949
United States of America	entered the	NATO	in	1949
United States of America	entered the	NAFTA	in	1989
Canada	entered the	NAFTA	in	1989

Fact type (Fact types)

A {Fact type} is a structure that enables recording of [Variable elements] of [Facts] that can be verbalized within a subject.

Fact type reading

A {Fact type reading} is a mould that belongs to a [Fact type] that consists of constant parts of a [Fact] and [Placeholders], with which the [Population] of a [Fact type] can be displayed in understandable sentences.

Placeholder (Placeholders)

A {Placeholder} is a part of a [Fact type]; each {Placeholder} in a [Fact type reading] has a corresponding [Role] in a [Fact type].

Population

A {Population} is a set of all [Variable elements] of a [Fact] that are being recorded in the [Role] of a [Fact type].

Position

Position is the place of a [Variable element] or constant in a [Fact type] or [Fact type reading].

Role (Roles)

A {Role} is part of a [Fact type]. It facilitates recording of one specific [Variable element] of those [Facts], for which all [Variable elements] are being recorded by means of this [Fact type]. [Fact types] always contain one or more {Roles}.

Variable element (Variable elements)

A {Role} is part of

{Variable elements} are the varying parts within a set of distinct [Facts], where these [Facts] must have the same kind of meaning and use the same kind of phrasing.

Above, the required concept definitions are given. In Figure L.9 we have defined an intermediate fact type diagram. We will now proceed to derive the uniqueness constraint for the fact type.

The following questions have to be asked:

a. Is it possible that the following two sentences can exist at the same time in the fact population?

   Fact type reading 1 has in position 1 a constant with contents ***Operating country*** and
   Fact type reading 1 has in position 1 a constant with contents **Operating company**

b. Is it possible that the following two sentences can exist at the same time in the fact population?

*Fact type reading 1 has in position 1 a **constant** with contents Operating country* and
Fact type reading 1 has in position 1 a **variable** with contents Operating country

c. Is it possible that the following two sentences can exist at the same time in the fact population?

*Fact type reading 1 has in position **1** a constant with contents Operating country* and
Fact type reading 1 has in position **2** a constant with contents Operating country

d. Is it possible that the following two sentences can exist at the same time in the fact population?

*Fact type reading **1** has in position 1 a constant with contents Operating country* and
Fact type reading **2** has in position 1 a constant with contents Operating country

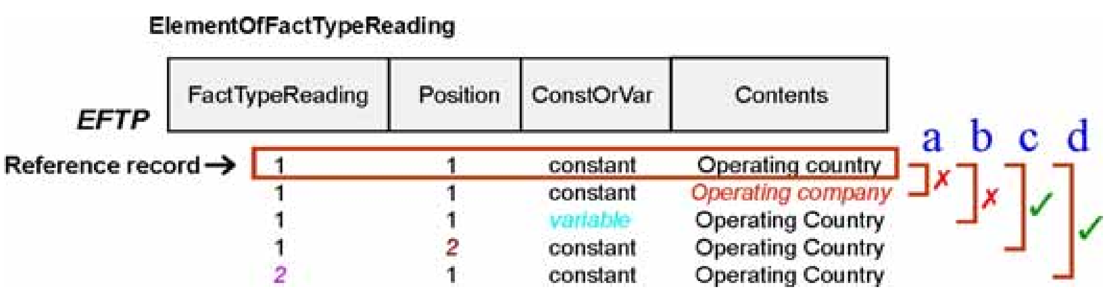


Figure L.10 - The register of answers given by the generic expert

Based on the above mentioned procedure, the following results are recorded:

Table L.6 - The register of answers and explanations given by the generic expert

Fact Type Reading	Position	ConstOrVar	Contents	Answer regarding the simultaneous existence with the first record	Explanation and notes
1	1	constant	operating country		Record 1 (Reference record)
1	1	constant	operating company	No	There can be only one kind of content in any given position.
1	1	variable	operating country	No	Every position in a fact type reading is either a constant or a variable. Additionally, we have to mention that we deal with a value rule for the role “ConstOrVar” which prescribes that only the values “constant” or “variable” can be used, see Figure L.12.

Table L.6 - The register of answers and explanations given by the generic expert

1	2	constant	operating country	Yes	An additional constraint is: two adjacent positions cannot both be of type constant.
2	1	constant	operating country	Yes	Of course

Based on these answers the analyst is able to derive the uniqueness constraint and adds it to the fact type diagram as shown in Figure L.11.

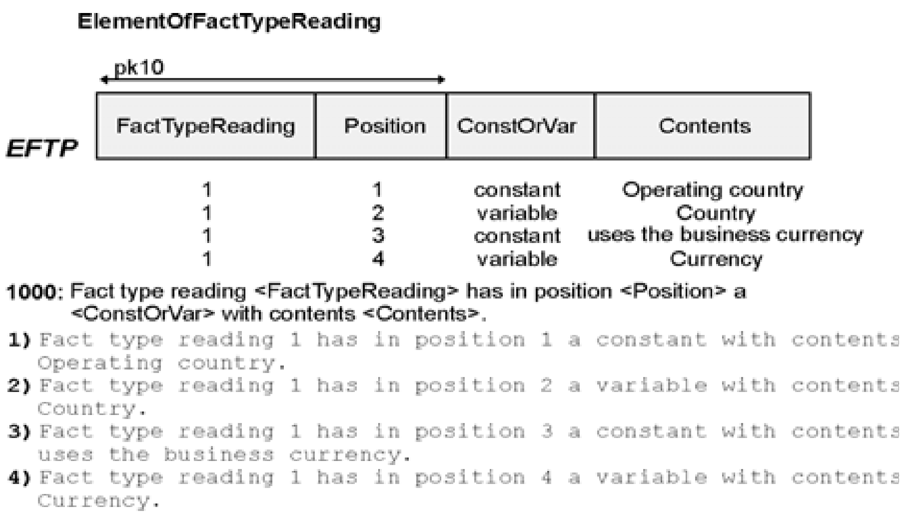


Figure L.11 - Meta fact type with population of domain-specific conceptual schema elements

Since we know that the role “ConstOrVar” can only contain the values “constant” or “variable,” a value constraint is applied to limit the possible values to the ones mentioned, as can be seen in Figure L.12. In addition, the necessary mandatory roles and data types are added (‘n’ is an abbreviation for ‘numeric’).

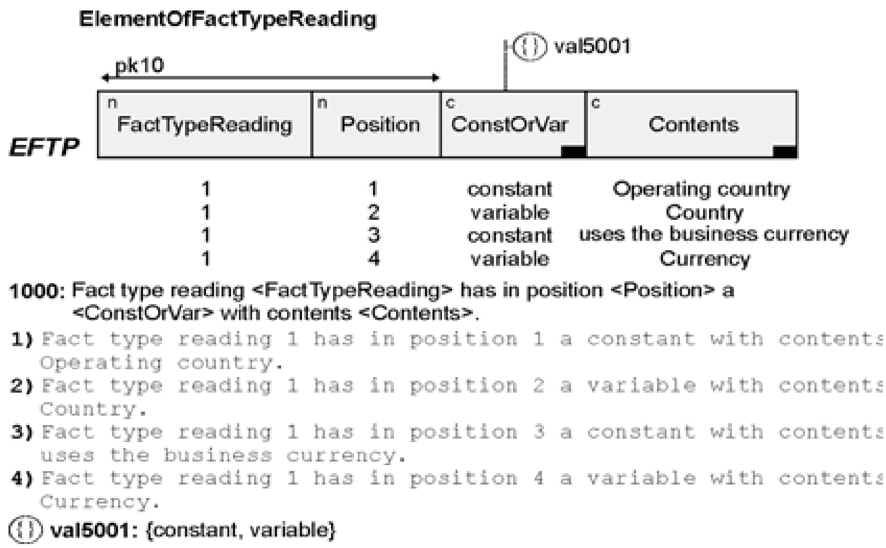


Figure L.12 - Meta fact type with population of domain-specific conceptual schema elements, mandatory role, value rule and data types added

Since we are now at the middle part of the generic component level in the knowledge triangle (i.e., level III), the question can be asked whether it is possible to verbalize (a part of) this fact type diagram. Using the methodology we used at levels I and II, we come to the result as shown in table L.7, when we verbalize the fact type reading of fact type diagram “ElementOfFactTypeReading.”

Table L.7 - Verbalization of a part of the generic conceptual schema

Fact type reading	1000	has in position	1	a	constant	with contents	Fact type reading
Fact type reading	1000	has in position	2	a	variable	with contents	FactTypeReading
Fact type reading	1000	has in position	3	a	constant	with contents	has in position
Fact type reading	1000	has in position	4	a	variable	with contents	Position
Fact type reading	1000	has in position	5	a	constant	with contents	a
Fact type reading	1000	has in position	6	a	variable	with contents	ConstOrVar

Table L.7 - Verbalization of a part of the generic conceptual schema

Fact type reading	1000	has in position	7	a	constant	with contents	with contents
Fact type reading	1000	has in position	8	a	variable	with contents	Contents

When the constant sentence parts are replaced by quotation marks, we come to the result shown in table L.8.

Table L.8 - The result of assigning constant and variable parts

Fact type reading	1000	has in position	1	a	constant	with contents	Fact type reading
" " "	1000	" " "	2	"	variable	" "	FactTypeReading
" " "	1000	" " "	3	"	constant	" "	has in position
" " "	1000	" " "	4	"	variable	" "	Position
" " "	1000	" " "	5	"	constant	" "	a
" " "	1000	" " "	6	"	variable	" "	ConstOrVar
" " "	1000	" " "	7	"	constant	" "	with contents
" " "	1000	" " "	8	"	variable	" "	Contents

From this it is evident that there is no higher level than the generic component level and that this level actually is self-describing. Since we previously concluded that the fact type diagram ElementOfFactTypeReading is a grammar rule for all fact type readings, as a consequence, it is possible to add the information from fact type reading 1000 to this diagram as well. Hence the rules of the generic component can be presented as fact population of the generic component itself, see Figure L.13.

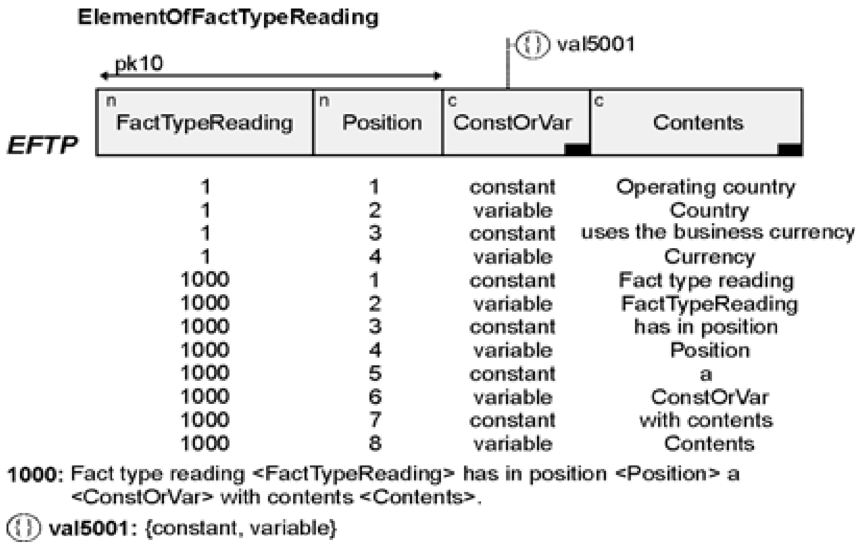


Figure L.13 - Meta fact type with population of both domain-specific and generic conceptual schema elements

We have now followed certain paths of the methodology in the knowledge triangle. In Figure L.14 a more elaborate knowledge triangle enhanced with processes is given. In this figure, the arrows (1b), (4b) and (7b) differ from the similar ones (1a), (4a) and (7a), because their function is not to use verbalization in order to get to a higher level, but to verbalize in a language business people are familiar with. Thus, no arrows of the kind of 2, 5 and 8 appear on the left side in the knowledge triangle.

L.5 Overview of structured and structuring knowledge

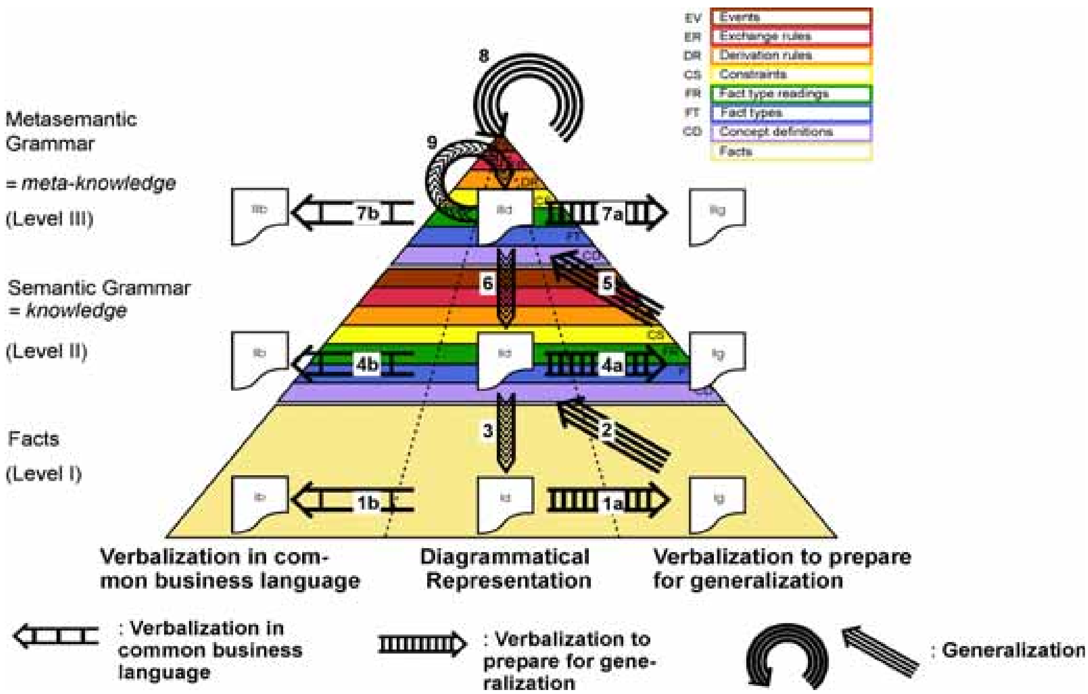


Figure L.14 - Knowledge triangle with process aspects

In conclusion, we started from the level of ground facts (level I in Figure L.14). At this level the facts have no grammatical function. By applying verbalization (1a and 1b) to the diagrammatic representation at the ground fact level (Id, i.e. level I diagrammatic representation), the results are the facts in a textual format (Ig and Ib). By applying generalization (2) to textual representation Ig at the ground facts level, the core of the domain-specific component of the conceptual schema was obtained in a diagrammatic format (IId).

This diagrammatic format of the domain-specific component of the conceptual schema (IId) - a semantic grammar - describes the meaning of terms at the ground fact level (Id) and it specifies the rules for fact populations (Id), fact population transitions (Id) and it contains the fact generation rules (IId). Hence IId determines (3) Id and describes its meaning.

Next, by applying verbalization with the aim to arrive at a higher level (4a) to the diagrammatic representation at the level of the domain-specific component of the conceptual schema (IId), we obtain a textual format of the domain-specific component (IIg).

Continuing this process, by using generalization (5) at level II, the result is a diagrammatic representation of a core part of the generic component of the conceptual schema (IIId). This diagrammatic format of the generic component of the conceptual schema - the metasemantic grammar - stipulates (6) the semantics and rules for the domain-specific component of the conceptual schema (IId). Again, by applying verbalization with the aim to arrive at the next level (7a) to the diagrammatic format of the generic component (IIId), we obtain a textual representation of a core part of the generic component of the conceptual schema (IIIg).

As was illustrated previously, by applying generalization (8) at level III, the result was the identical representation of the metasemantic grammar, i.e. there is no higher conceptual level than level III.

The beauty of (IIId), the generic component of the conceptual schema, is that in effect it stipulates itself (9)! The route we followed regarding the creation of time invariant knowledge is also illustrated in Figure L.15, with concrete examples.

The result of (4b) and (7b) could be SBVR Structured English. The aim of (IIb) and (IIIb) is to be understandable to persons who do not know the diagrammatical representation, but do of course know well-expressed English sentences.

L.6 Summary and recommendation

SBVR is a major step forward for the business and education community. The era of sufficient attention to semantics has begun in earnest. SBVR covers many aspects which cannot all be discussed in one annex as the annex would become a textbook. Various useful concepts of SBVR have not been discussed in this annex as there was a limit to the number of concepts to be illustrated in this annex. E.g. the concepts of necessity, obligation, permissibility, and possibility have intentionally not been discussed in this annex. It aims to be a useful add-on to the other annexes and has therefore concentrated on

a. a diagrammatic overview of some core concepts

b. concept definitions

c. a diagrammatic representation of fact types with the longest experience in industry

d. verbalization of fact instances, to be distinguished from rule verbalization as illustrated in Annexes C, F and I and

e. a small part of a long standing methodology which shows the power of SBVR.

My expectation is that SBVR 101, SBVR 102 and SBVR 103 will start in 2007. Sooner or later it will be taught in nearly all business oriented faculties.

Our recommendation to experienced fact oriented experts is to promote widespread use of SBVR.

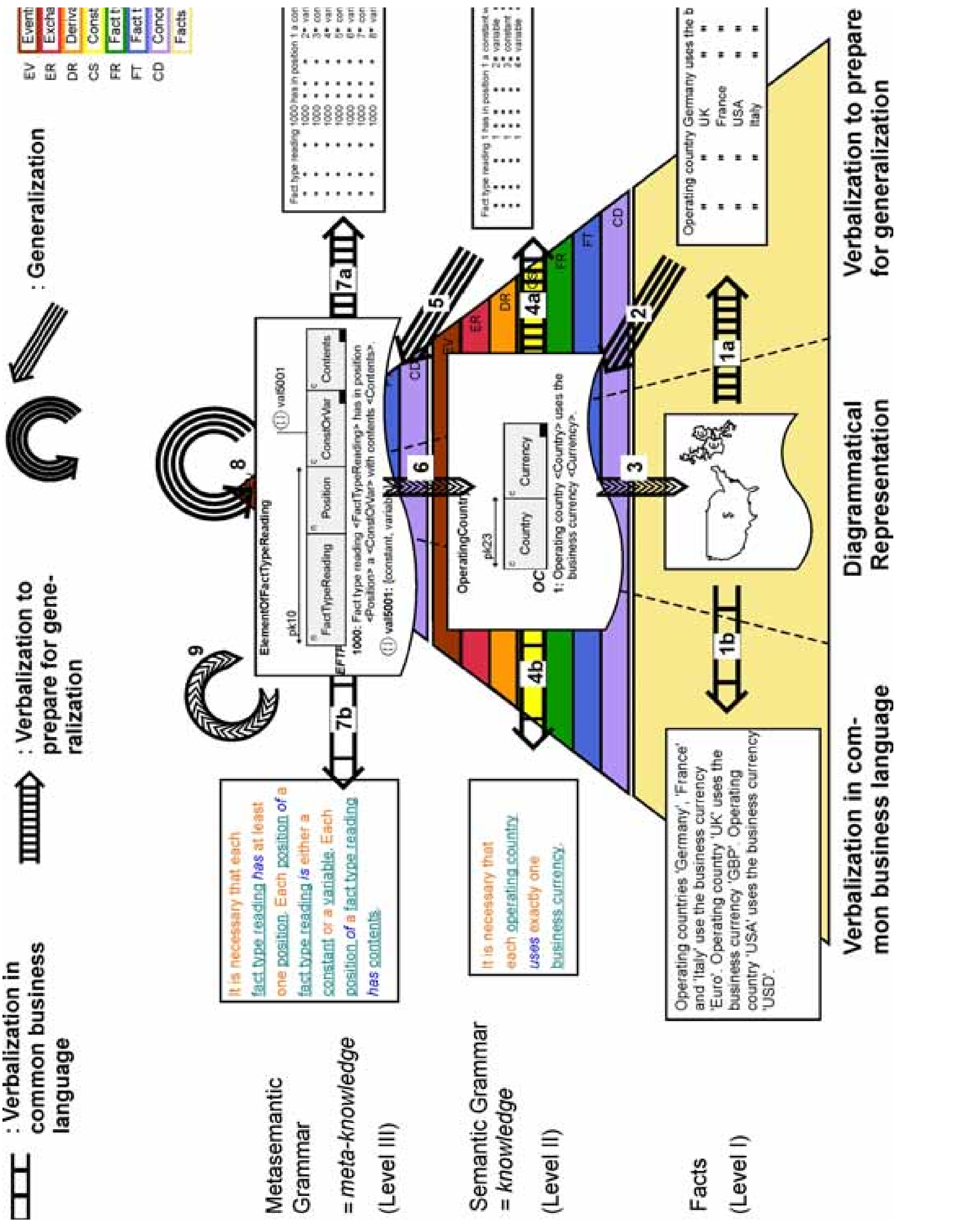


Figure L.15 - Knowledge triangle, with concrete input and output of processes

Annex M
(informative)

Additional References

M.1 Bibliography / Normative References

[AH] *American Heritage Dictionary.*

[Anto2001] Antonelli, A. *Non-Monotonic Logic*, Stanford Encyclopedia of Philosophy, 2001. Available from http://plato.stanford.edu/entries/logic-nonmonotonic/

[Bloe1996] Bloesch, A.C., and Terry A. Halpin. "ConQuer: a Conceptual Query Language." In *Proc. ER'96: 15th Int. Conf. on Conceptual Modeling*, 121-133: Springer LNCS, 1996. Available from http://www.orm.net/pdf/ER96.pdf

[Bloe1997] ________. "Conceptual Queries using ConQuer-II." In *Proc. ER'97: 16th Int. Conf. on Conceptual Modeling*, 113-126: Springer LNCS, 1997. Available from http://www.orm.net/pdf/ER97-final.pdf

[BMM] Business Rules Group. *The Business Motivation Model ~ Business Governance in a Volatile World*. 1.2 ed., Sept. 2005. Originally published as *Organizing Business Plans ~ The Standard Model for Business Rule Motivation*, Nov. 2000. Available from http://www.BusinessRulesGroup.org

[BRM] Business Rules Group. Ronald G. Ross, ed. *Business Rules Manifesto ~ The Principles of Rule Independence*. 1.2 ed. The Business Rules Group, 2003. Updated Jan. 8, 2003. PDF. Available from http://www.BusinessRulesGroup.org/brmanifesto.htm

[BRG2002] Business Rules Group. *Defining Business Rules ~ What Are They Really?* 4th ed., July 2002. Originally published as *GUIDE Business Rules Project Report*, 1995. Available from http://www.BusinessRulesGroup.org

[BRJ2005] Editors of BRCommunity.com. "A Brief History of the Business Rule Approach." *The Business Rules Journal* 6, no. 1 (2005). Available from http://www.BRCommunity.com/a2005/b216.html

[CDP] *The Cambridge Dictionary of Philosophy.* 2nd ed.: Cambridge University Press, 1999.

[CSILL] *Cognitive Science Initiative: Language Lexicon*. University of Houston. Available from http://www.hfac.uh.edu/COGSCI/lang/Entries/

[Dean1997] Dean, Neville. *The Essence of Discrete Mathematics*, The Essence of Computing Series: Prentice-Hall, 1997.

[Fitt2002 (or TTGG)] Fitting, Melvin. *Types, Tableaus, and Gödel's God*, Trends in Logic, Studia Logica Library. Dordrecht, the Netherlands: Kluwer Academic Publishers, 2002.

[Girl2000 (or MLP)] Girle, Rod A. *Modal Logics and Philosophy*: McGill-Queen's University Press, 2000.

[Halp1989 (or HALT89)] Halpin, Terry A. "A Logical Analysis of Information Systems: Static Aspects of the Data-oriented Perspective." PhD thesis, Department of Computer Science, University of Queensland, 1989.

[Halp1998] ________. "Object-Role Modeling (ORM/NIAM)." In *Handbook on Architectures of Information Systems*. Heidelberg: Springer, 1998.

[Halp2000] ________. *Object-Role Modeling: An Overview.* San Francisco: Springer, 2000. Available from http://www.orm.net/pdf/springer.pdf

[Halp2001 (or IMRD)] ________. *Information Modeling and Relational Databases*. San Francisco: Morgan Kaufmann, 2001.

[Halp2003a] ________. "Verbalizing Business Rules: Part 1." *The Business Rules Journal* 4, no. 4 (2003). Available from http://www.BRCommunity.com/a2003/b138.html

[Halp2003b] ________. "Verbalizing Business Rules: Part 2." *The Business Rules Journal* 4, no. 6 (2003). Available from http://www.BRCommunity.com/a2003/b152.html

[Halp2003c] ________. "Verbalizing Business Rules: Part 3." *The Business Rules Journal* 4, no. 8 (2003). Available from http://www.BRCommunity.com/a2003/b163.html

[Halp2003d] ________. "Verbalizing Business Rules: Part 4." *The Business Rules Journal* 4, no. 10 (2003). Available from http://www.BRCommunity.com/a2003/b172.html

[Halp2004 (or HALT2004)] ________. "Information Modeling and Higher-Order Types." In *Proc. CAiSE'04 Workshops*, eds. J. Grundspenkis and M. Kirkova, 1, 233-248: Riga Tech. University, 2004. Available from http://www.orm.net/pdf/EMMSAD2004.pdf

[Halp2004b] ________. "Business Rule Verbalization." In *Lecture Notes in Informatics*, eds. A. Doroshenko, Terry A. Halpin and S. Liddle, P-48, 39-52. Salt Lake City: Proc. ISTA-2004, 2004.

[Halp2004c] ________. "Verbalizing Business Rules: Part 5." *The Business Rules Journal* 5, no. 2 (2004). Available from http://www.BRCommunity.com/a2004/b179.html

[Halp2004d] ________. "Verbalizing Business Rules: Part 6." *The Business Rules Journal* 5, no. 4 (2004). Available from http://www.BRCommunity.com/a2004/b183.html

[Halp2004e] ________. "Verbalizing Business Rules: Part 7." *The Business Rules Journal* 5, no. 7 (2004). Available from http://www.BRCommunity.com/a2004/b198.html

[Halp2004f] ________. "Verbalizing Business Rules: Part 8." *The Business Rules Journal* 5, no. 9 (2004). Available from http://www.BRCommunity.com/a2004/b205.html

[Halp2004g] ________. "Verbalizing Business Rules: Part 9." *The Business Rules Journal* 5, no. 12 (2004). Available from http://www.BRCommunity.com/a2004/b215.html

[Halp2005a] ________. "Verbalizing Business Rules: Part 10." *The Business Rules Journal* 6, no. 4 (2005). Available from http://www.BRCommunity.com/a2005/b229.html

[Halp2005b] ________. "Verbalizing Business Rules: Part 11." *The Business Rules Journal* 6, no. 6 (2005). Available from http://www.BRCommunity.com/a2005/b238.html

[Halp2005c] ________. "Verbalizing Business Rules: Part 12." *The Business Rules Journal* 6, no. 10 (2005). Available from http://www.BRCommunity.com/a2005/b252.html

[Halp2005d] ________. "Verbalizing Business Rules: Part 13." *The Business Rules Journal* 6, no. 12 (2005). Available from http://www.BRCommunity.com/a2005/b261.html

[Halp1981 (or DL)] Halpin, Terry A., and Rod A. Girle. *Deductive Logic*. 2nd ed. Brisbane: Logiqpress, 1981.

[Hunt1971 (or META)] Hunter, Geoffrey. *An Introduction to the Metatheory of Standard First Order Logic*: University of California Press, 1971.

[IETF RFC 2396] Berners-Lee, Tim, R. Fielding, and L. Masinter. *Uniform Resource Identifiers (URI): Generic Syntax*. The Internet Society, 1998. Updated August 1998. Available from http://www.ietf.org/rfc/rfc2396.txt

Issue 10568 Add text

[ISO6093] International Organization for Standardization (ISO). *Information processing - Representation of numerical values in character strings for information interchange*. ISO, 1985.

[ISO704] ________. *Terminology work - Principles and Methods*. English ed.: ISO, 2000.

[ISO1087-1] ________. *Terminology work - Vocabulary - Part 1: Theory and Application*. English/French ed.: ISO, 2000.

[ISO860] ________. *Terminology work - Harmonization of Concepts and Terms*. ISO, 1996.

[ISO639-2] ________. *Codes for the Representation of Names of Languages-- Part 2: Alpha-3 Code*. Library of Congress, 2002. Available from http://www.loc.gov/standards/iso639-2/langcodes.html

[ISO/IEC CD 24707] ________. *Information technology -- Common Logic (CL) -- A Framework for a Family of Logic-Based Languages*: ISO, 2005. Available from http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39175

[Levi1983 (or LEVS)] Levinson, Stephen C. *Pragmatics*, Cambridge Textbooks in Linguistics: Cambridge University Press, 1983.

[MATH] *PlanetMath.org*. Available from http://planetmath.org/encyclopedia

[Mend1997 (or MEN97)] Mendelson, Elliott. *Introduction to Mathematical Logic*. 4th ed.: Chapman & Hall, 1997.

[MWCD] *Merriam-Webster Collegiate Dictionary.*

[MWDS] *Merriam-Webster Dictionary of Synonyms*.

[MWU] *Merriam-Webster Unabridged*.

Issue 11263 Add text

[Nijs1977] Nijssen, Sjir. “On the Gross Architecture for the Next Generation Database Management Systems.” In: *Proc. IFIP’77, 1977 IFIP Working Conf. on Modelling in Data Base Management Systems*, ed. B. Gilchrist, 327-335: North Holland Publishing Company, 1977.

[Nijs1978] ________. “A Framework for Discussion.” In: *ISO/TC97/SC5/WG3* and comments on *78.04/01* and *78.05/03*, 1-144.

[Nijs1980] ________. “A Framework for Advanced Mass Storage Applications.” In: *Proc. IFIP MEDINFO’80, 3rd World Conference on Medical Informatics*: North Holland Publishing Company, 1980.

[Nijs1986] ________. “On Experience with Large-scale Teaching and Use of Fact-Based Conceptual Schemas in Industry and University.” In: *Proc. DS-1’85: IFIP WG 2.6 Working Conference on Data Semantics*, eds. T.B. Steel and R. Meersman, 189-204: North Holland Publishing Company, 1986.

[Nijs2006] Nijssen, Sjir, and R. Bijlsma. “A Conceptual Structure of Knowledge as a Basis for Instructional Designs.” In: *Proc. ICALT’06, IEEE: 6th Int. Conf. on Advanced Learning Technologies*, eds. R. Kinshuk, P. Koper, P. Kommers, D. Kirschner, G. Sampson, and W.E. Didderen, 7-9: IEEE, 2006.

[NODE] *The New Oxford Dictionary of English*.

[Nolt1998 (or LSO)] Nolt, John, Dennis Rohatyn, and Achille Varzi. *Logic*. 2nd ed., Schaum's Outlines. New York: McGraw-Hill, 1998.

[ODE] *Oxford Dictionary of English*.

[OSM] *Organizational Structure Metamodel*: OMG, 2005.

[Peik (or PEIL)] Peikoff, Leonard. "The Analytic-Synthetic Dichotomy." In *Rand1990*, 88-121.

[Rand1990 (or RANA90)] Rand, Ayn. *Introduction to Objectivist Epistemology.* expanded 2nd ed. New York: Meridian, 1990.

[Ross1997] Ross, Ronald G. *The Business Rule Book -- Classifying, Defining and Modeling Rules*. 2nd ed. Houston, TX: Business Rule Solutions, Inc., 1997. Originally published as *The Business Rule Book (1st Ed.),* 1994. Available from http://www.BRSolutions.com

[Ross2003] ________. *Principles of the Business Rule Approach*. Boston, MA: Addison-Wesley, 2003. Available from http://www.BRSolutions.com

[Ross2005] ________. *Business Rule Concepts: Getting to the Point of Knowledge*. 2nd ed.: Business Rule Solutions, LLC, 2005. Available from http://www.BRSolutions.com

[RuleSpeak] Business Rule Solutions. *BRS RuleSpeak® Practitioner's Kit*. Business Rule Solutions, LLC, 2001-2004. PDF. Available from http://BRSolutions.com/p_rulespeak.php

[SEP] *Stanford Encyclopedia of Philosophy*. Edward N. Zalta, ed. The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University. Available from http://plato.stanford.edu/

[SOED] *Shorter Oxford Dictionary of English*.

[SubePLTS (or PLTS)] Suber, Peter. *Propositional Logic Terms and Symbols*. Philosophy Department, Earlham College, 1997. Available from http://www.earlham.edu/~peters/courses/log/terms2.htm

[SubeGFOL (or GFOL)] ________. *Glossary of First-Order Logic*. Philosophy Department, Earlham College, 1999-2002. Available from http://www.earlham.edu/~peters/courses/logsys/glossary.htm

[UML2infr] Object Management Group (OMG). *Unified Modeling Language: Infrastructure*. Ver. 2.0: OMG.

[Unicode4] "The Unicode Standard, Version 4.0.0." In *The Unicode Standard, Version 4.0*. Boston, MA: Addison-Wesley, 2003. Available from http://www.unicode.org/versions/Unicode4.0.0/b1.pdf

[USG] The Unicode Consortium. *Glossary of Unicode Terms*. 1991-205. Updated Nov. 17 2004. Available from http://www.unicode.org/glossary/

[W3ID] *Webster's 3rd New International Dictionary.*

[WD] *Webster's Dictionary.*

[XMI2.1] *XML Metadata Interchange (XMI)*. Ver. 2.1: OMG.