Date: January 2025

Standard Business Report Model (SBRM)

V1.0 – beta 1

OMG Document Number: dtc/25-01-01

URI: https://www.omg.org/spec/SBRM/

This OMG document replaces the submission document (bmi/24-12-01). It is an OMG Adopted Beta Specification and is currently in the finalization phase. Comments on the content of this document are welcome and should be directed to issues@omg.org by June 30, 2025.

You may view the pending issues for this specification from the OMG revision issues web page https://issues.omg.org/issues/lists.

The FTF Recommendation and Report for this specification will be published in September 2025. If you are reading this after that date, please download the available specification from the OMG Specifications Catalog.

Copyright © 2021-24, Auditchain Labs AG,
Copyright © 2019-23, Charles Hoffman CPA
All rights in above assigned to Auditchain Labs, AG; use herein is by permission.
Copyright © 2019-24, Federated Knowledge, LLC
Copyright © 2023-24, EDM Council, Inc

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

The IPR Mode of this specification, as defined in OMG document ipr/12-09-02, is Non-Assert Mode.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 9C Medway Road, PMB 274, Milford, MA, 01757, U.S.A.

TRADEMARKS

CORBA®, CORBA logos®, FIBO®, Financial Industry Business Ontology®, FINANCIAL INSTRUMENT GLOBAL IDENTIFIER®, IIOP®, IMM®, Model Driven Architecture®, MDA®, Object Management Group®, OMG®, OMG Logo®, SoaML®, SOAML®, SysML®, UAF®, Unified Modeling Language®, UML®, UML Cube Logo®, VSIPL®, and XMI® are registered trademarks of the Object Management Group, Inc.

For a complete list of trademarks, see: https://www.omg.org/legal/tm_list.htm. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

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 Specifications, Report a Bug/Issue.

Table of Contents

1 Scope....1
2 Conformance....2
3 Normative References....3
4 Terms and Definitions....4
4.1 General Terms....4
4.2 Acronyms and Abbreviations....5
5 Symbols....7
Symbols 7
6 Additional Information....8
6.1 Changes to Adopted OMG Specifications....8
6.2 Acknowledgments....8
6.3 Intellectual Property Rights....8
6.4 Ontology Notation....9
7 Metamodel....11
7.1 Report Submission....11
7.2 Simple Report Design Approach....12
7.3 Financial Report Design Approach....14
7.4 Facts for Report Submissions....16
7.5 Classes....17
7.6 Report Definition - Types of Report Element....19
7.7 Top Level Structure....22
7.8 Report Structure....25
7.9 Concept Arrangement Patterns....28
7.9.1 Set Concept Arrangement Pattern....31
7.9.2 Roll Up Concept Arrangement Pattern....33
7.9.3 Roll Forward Concept Arrangement Pattern....35
7.9.4 Roll Forward Info Concept Arrangement Pattern....37
7.9.5 Adjustment Concept Arrangement Pattern....39
7.9.6 Variance Concept Arrangement Pattern....41
7.9.7 Arithmetic Expression Concept Arrangement Pattern....42
7.10 Hypercube Structure....45
7.11 Datatypes....47
8. Ontology....50
8.1 Namespace Definitions for External Ontologies....50
8.2 Namespace Definitions for the Ontologies Included in this Specification....51
8.3 Reporting Envelope Ontology....52
8.4 Report Structure Ontology....58
8.5 Financial Report Elements Ontology....59
Annex A. Not Used....61
Annex B. Document Components Ontology (Normative)....62
Annex C Standard Business Report Model Narrative Introduction (Informative)....68
C.1 Report Basics....68
C.2 Report Rendering....73
C.3 Report Structure....77
Annex D. Simple Narrative Example (Informative)....82

Preface

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 https://www.omg.org/.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from the OMG website at:
https://www.omg.org/spec

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. at:

OMG Headquarters
9C Medway Road, PMB 274
Milford, MA 01757 USA
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 https://www.iso.org

1 Scope

The Standard Business Reporting Model (SBRM) provides a set of modeling constructs for representing business and regulatory reporting that are reusable in different modeling and data deployment environments. It acts as a Platform Independent Model for report documents and data structures, complementing established platform-specific models such as eXtensible Business Reporting Language (XBRL).

Key features:

- Applies a business modeling approach to report definition, making reports easier to develop;
- Supports representation of reports ranging from simple structured text, through structured/formatted data, up to complete business and/or financial models;
- Enables business-level expression of rules in order to ensure quality and consistency;
- Facilitates reuse of information across multiple reports;
- Supports integration with industry- or enterprise-standard information models and ontologies;
- Supports direct linkage with enterprise information sources for report data, and provides complete data lineage and other metadata regarding the source for any content represented;
- Offers alternative formats for the same report content and structure, to allow access by a greater variety of analytical tooling;
- Separates the reporting envelope and structure of the report from the definition of the payload of that report (*i.e.*, the domain specific content);
- Enables modularizing and repurposing the content of any report, not only across reports but across data sources and business units;
- Enables auto-generation of reports from primarily structured content, including but not limited to the use of machine learning and generative artificial intelligence to do so;
- Supports saving and managng immutable versions of reports as submitted to any regulator or other authority (required for pharmaceutical and biomedical reporting);
- Supports both generating and reverse-engineering existing reports and report definitions in XBRL.

It provides a MOF metamodel and OWL ontologies.

2 Conformance

The aim of this specification is to define a logical system which can be executed to produce and validate business reports. Any conformant software should be consistent with the use of these components; however software does not need to be componentized in this manner and could, for example, be a single monolith supporting the same functionality and interchange capabilities.

Note that many triple stores include facilities for integration of additional capabilities, ranging from rule engines to calculation capabilities. Rules may be specified in languages such as Datalog, and need not be limited to SHACL.

The different aspects have been factored into several named conformance points: software may claim conformance to any or all of these. There are some dependencies noted below, *e.g.*, Structured data conformance is dependent on Report ontology conformance.

Table 2.1: Conformance

Report metamodel conformance	Generate report documents conformant with the SBRM metamodel
Report ontology conformance	Generate report documents conformant with the SBRM ontologies
Structured data conformance	Report ontology conformance, plus documents additionally conform with any ontologies referenced by the Facts
Numeric data conformance	Structured data conformance, plus documents additionally conform with any Concept Arrangement Patterns (see 7.9) referenced by the Facts
Report modeling	Interact with a user to create a SBRM Report Definition
Manual report generation	Interact with a user to obtain the values for Facts to create a SBRM Report Submission
Report ontology linking	Link a report definition with external ontology elements
Automated report generation	Create a SBRM report document by extracting the information from an IT system
XBRL taxonomy generation	Convert a SBRM report definition to an XBRL-compliant taxonomy
XBRL report generation	Convert a SBRM report submission to XBRL-compliant format
XBRL taxonomy ingestion	Convert a XBRL-compliant taxonomy to a SBRM report definition
XBRL report ingestion	Convert a XBRL-compliant report to a SBRM report

3 Normative References

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.

[Commons] Commons Ontology Library https://www.omg.org/spec/Commons

[Dublin Core] Dublin Core™ Metadata Initiative Terms
https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

[ISO 11404] ISO/IEC 11404:2007, Information technology — General-Purpose Datatypes (GPD), https://www.iso.org/standard/39479.html

[JSON] JavaScript Object Notation (JSON) Data Interchange Format
https://tools.ietf.org/html/rfc8259

[MOF] Meta Object Facility
https://www.omg.org/spec/MOF

[OWL] OWL 2 Web Ontology Language Document Overview (Second Edition).
W3C Recommendation, 2012
https://www.w3.org/TR/owl2-overview

[RDF] Resource Description Framework (RDF) 1.1: Concepts and Abstract Syntax.
W3C Recommendation, 2014
https://www.w3.org/TR/rdf11-concepts

[SHACL] Shapes Constraint Language (SHACL)
W3C Recommendation 20 July 2017
https://www.w3.org/TR/shacl/

[SPAR] Semantic Publishing and Referencing Ontologies (SPAR). Shotton, D., Portwin, K., Klyne, G., Miles, A. (2009). Adventures in Semantic Publishing: Exemplar Semantic Enhancements of a Research Article. In PLoS Computational Biology, 5(4): e1000361. http://dx.doi.org/10.1371/journal.pcbi.1000361; Documentation is available at http://www.sparontologies.net/. The ontologies themselves are managed in GitHub at https://github.com/sparontologies.

[SPARQL] SPARQL 1.1 Query Language
W3C Recommendation 21 March 2013
https://www.w3.org/TR/sparql11-query/

[Turtle] Terse RDF Triple Language
W3C Recommendation 25 February 2014
https://www.w3.org/TR/turtle/

[UML] Unified Modeling Language
https://www.omg.org/spec/UML

[XBRL] XBRL version 2.1 Specification, https://specifications.xbrl.org/work-product-index-group-base-spec-base-spec.html

[XBRL Book] The XBRL Book, Ghislain Fourny, Sixth Edition, March 2023

[XMI] XMI (XML Metadata Interchange) Mapping Specification
https://www.omg.org/spec/XMI/

4 Terms and Definitions

4.1 General Terms

For the purposes of this specification, the following terms and definitions apply.

Aspect

A characteristic of a fact in addition to its value such as unit of measure, time or geographical scope; they can be used to organize or present a collection of facts in different ways.

Concept

The business meaning associated with each fact.

(Reporting) Entity

The organization which is the subject of a report.

Fact

A single data point, including metadata, for a concept in a report, equivalent to a cell in a spreadsheet, including its value. Often the fact value takes the form of a number, but it can also be textual information, or narrative/prose.

Fact provenance

The definition of the WHO, WHAT, WHEN and HOW aspects of a fact statement that make that statement uniquely identifiable across time and location.

Hypercube

A multi-dimensional data structure used to represent complex financial information. It allows the data to be analyzed and viewed from multiple perspectives simultaneously, enabling more dynamic and detailed reporting. Hypercubes are particularly useful for handling large datasets with various dimensions, such as time, geographic location, product lines, and financial metrics.

Pattern

A common way of relating different facts in a report such as the addition of a set of fact values to create a total (called a roll up).

Report

A structured file or document created or generated by an organization for intended delivery to one or more other organizations, typically a regulator.

Report component

An identified part of a report such as a section or table. They are typically arranged in a hierarchy, *e.g.*, a report contains sections which contain tables.

Report definition

The definition of the structure, semantics and values of a particular kind of business report stated in terms of the specification language of the Standard Business Report Model.

Report rule

A business-oriented constraint specifying how different facts in a report should relate to each other for the purposes of quality and consistency.

Reporting Scenario

A distinct condition or context used for a set of reported information—such as actual vs. budgeted, forecasted vs. prior periods, or different geographic regions—that are compared to calculate variance.

Submission

A Report as actually sent to a specific recipient.

4.2 Acronyms and Abbreviations

DL – Description Logics

FIBO – Financial Industry Business Ontology

IDMP – Identification of Medicinal Products

IRI – Internationalized (Uniform) Resource Identifier

JSON – Javascript Serialized Object Notation

ISO – International Organization for Standardization

LCC – Languages, Countries and Codes

MVF – Multiple Vocabulary Facility

OWL – Web Ontology Language

ODM – Ontology Definition Metamodel

RDF – Resource Definition Framework

SHACL – Shapes Constraint Language

TTL – Turtle (Terse RDF Triple Language)

UML – Unified Modeling Language

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

W3C – World Wide Web Consortium

XBRL – eXtensible Business Reporting Language

XMI – XML Metadata Interchange

XML – eXtensible Markup Language

5 Symbols

Symbols

See clause 6.4, Notation, for a description of the logic symbols used to describe the ontologies covered in this specification.

6 Additional Information

6.1 Changes to Adopted OMG Specifications

See Annex B which specifies a new ontology describing certain components of a document to be added to OMG's Commons Ontology Library specification.

6.2 Acknowledgments

The following individuals have supported and contributed to developing this specification :

Pete Rivett
Adaptive, Inc, Auditchain Labs AG, and Federated Knowledge, LLC
email: pete.rivett@federatedknowledge.com

Elisa Kendall
EDM Council, Inc.
email: ekendall@thematix.com

Charles Hoffman, CPA
Auditchain Labs AG
email: charles.hoffman@auditchain.com

Jason Meyers
Auditchain Labs AG
email: jm@auditchain.com

Dr. Heiner Oberkampf
ACCURIDS
email: heiner.oberkampf@accurids.com

Dr. Thomas Weber
OSTHUS GmbH – a PharmaLex Company
email: thomas.weber@pharmalex.com

Gerhard Noelken
Pistoia Alliance, Inc.
email: gerhard.noelken@pistoiaalliance.org

Sheila Elz, Master Data Manager
Bayer AG
Pharmaceuticals // R & D Data Science & AI
email: sheila.elz@bayer.com

6.3 Intellectual Property Rights

The ontologies included in this specification are open-source and available to anyone under the terms of **OMG's Non-Assertion Covenant**. The ontologies themselves are licensed under the **MIT open-source license** , available at https://opensource.org/licenses/MIT.

In addition, SPAR ontologies referenced by this specification are also open-source, licensed under the **Creative Commons Attribution 4.0 International License (CC BY 4.0) available at https://creativecommons.org/licenses/by/4.0/** .

6.4 Ontology Notation

The notation used to represent description logic expressions (*i.e.*, the expressions in the Parent columns in class tables containing ontology details) is consistent with the notation defined in the Description Logic Handbook [DL Handbook]. The notation used in this specification, representing a subset of OWL 2, is described in Table , below.

Table : Description Logic Expressions Notation

Construct	*Description*	*Notation*
Boolean Connectives and Enumeration		
intersection	The intersection of two classes consists of exactly those individuals which are instances of both classes.	$C \cap D$
union	The union of two classes contains every individual which is contained in at least one of these classes.	$C \cup D$
enumeration	An enumeration defines a class by enumerating all its instances.	oneOf (i_1, i_2, i_3, … i_n)
Property Restrictions		
universal quantification	Universal quantification is used to specify a class of individuals for which all related individuals must be instances of a given class (*i.e.*, allValuesFrom in OWL).	∀R.C, where R is the relation (property) and C is the class that constrains all values for related individuals
existential quantification	Existential quantification is used to specify a class as the set of all individuals that are connected via a particular property to at least one individual which is an instance of a certain class (*i.e.*, someValuesFrom in OWL).	∃R.C, where R is the relation (property) and C is the class that constrains some values of related individuals
individual value	Individual value restrictions are used to specify classes of individuals that are related to one particular individual (*i.e.*, hasValue in OWL).	∀R.I, where R is the relation (property) and I is the individual
exact cardinality	Cardinality (number) restrictions specify classes by restricting the cardinality on the sets of fillers for roles (relationships, or properties in OWL). Exact cardinality restrictions restrict the cardinality of possible fillers to exactly the number specified.	= n R (for unqualified restrictions) = n R.C (for qualified restrictions, i.e., including onClass or on DataRange)
maximum cardinality	Maximum cardinality restrictions restrict the cardinality of possible fillers to at most the number specified (inclusive).	≤ n R (for unqualified restrictions)

		≤ n R.C (for qualified restrictions)
minimum cardinality	Minimum cardinality restrictions restrict the cardinality of possible fillers to at least the number specified (inclusive).	≥ n R (for unqualified restrictions) ≥ n R.C (for qualified restrictions)
Class Axioms		
equivalent classes	Two classes are considered equivalent if they contain exactly the same individuals.	≡ C
disjoint classes	Disjointness means that membership in one class specifically excludes membership in another.	¬ C
Property Axioms		
complex role inclusions	Role inclusions allow [object] properties to be chained together in a sequence that is a subproperty of a higher-level property.	R ○ R

Note that in the case of complex restrictions, where there are nested elements in parentheses, the "dot notation" used as a separator between a property and the role filler is replaced with the embedded parenthetical filler definition. A "role" from a description logic perspective is essentially a property in OWL, and the role "filler" is the class or individual that provides the value for that role in a given axiom (*i.e.*, in a restriction or other logic expression).

7 Metamodel

For a narrative introduction to the subject matter and concepts please start with Annex C.

This section presents the SBRM Metamodel based on the Meta-Object Facility ([MOF]) standard. The metamodel is documented using [UML] class diagrams, which provide a structured and detailed representation of the core elements and their relationships. These diagrams serve as both a blueprint for understanding the metamodel and a visual representation of the concepts embedded within the ontologies.

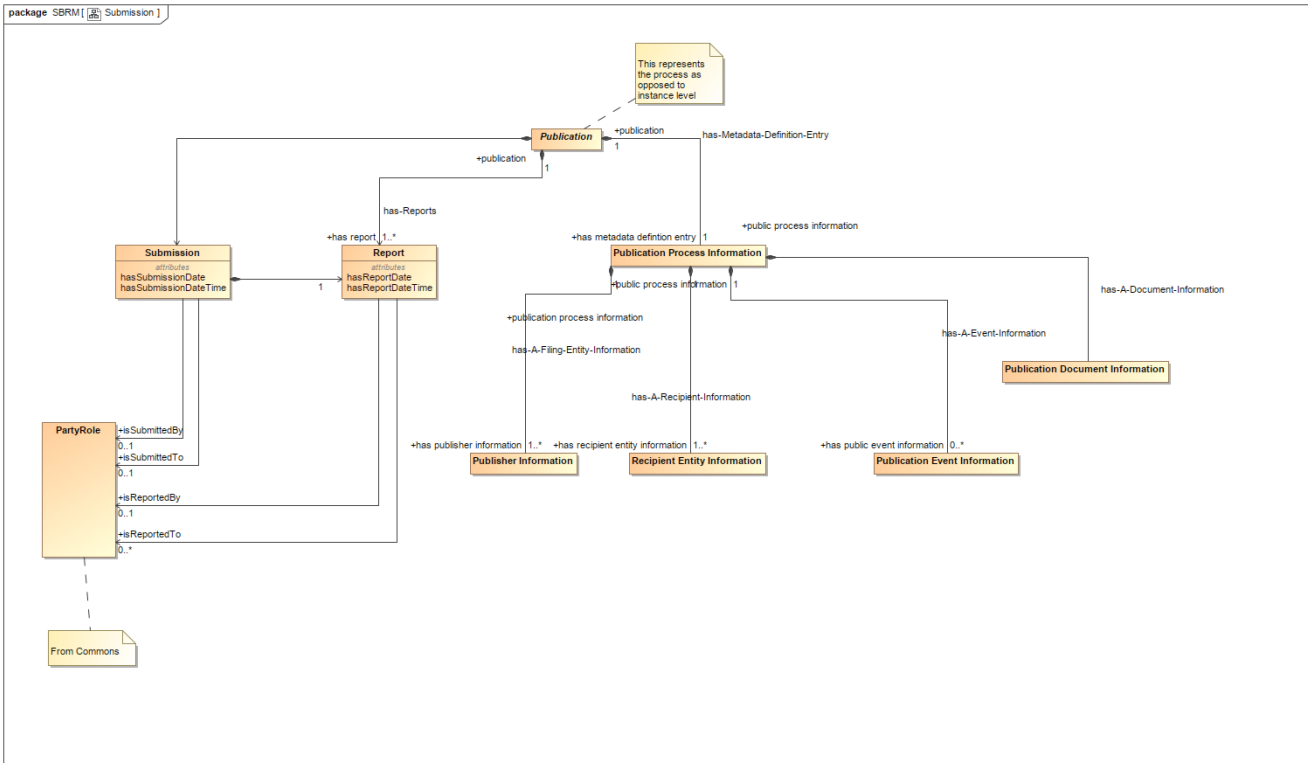
The UML diagrams included in this section depict the structural relationships between classes and align with the semantic constructs defined in the associated ontologies. This dual representation ensures consistency between the metamodel and the ontological framework, facilitating interoperability and enabling clear communication of the underlying design.

The diagrams also provide a visual representation of the concepts in the ontologies.

7.1 Report Submission

The Report Submission UML Diagram in Figure 1 models the process of submitting and publishing a report, capturing key entities, their attributes, and their relationships. At the core of the diagram is the ***Submission*** class, which represents the submission process itself, characterized by attributes like ***hasSubmissionDate*** and ***hasSubmissionDateTime***. Submissions are directly linked to one or more Reports, which include attributes such as ***hasReportDate*** and ***hasReportDateTime*** to specify the reporting timeline. The ***PartyRole*** class introduces flexibility in defining the roles associated with the submission, such as ***isSubmittedBy***, ***isReportedTo***, and ***isSubmittedTo***, enabling a detailed account of the parties involved.

The diagram further extends into the ***Publication*** process, which bridges the report and the broader publication lifecycle. This lifecycle is managed through the ***Publication Process Information*** class, which connects the publication to its key components, such as metadata, publisher information, recipient information, and any associated events or documents. The multiplicity in relationships (e.g., 1..* for publishers and recipients, 0..* for public events) ensures flexibility and completeness in capturing the publication details. Additionally, the ***Publication Process Information*** class linked to ***Publication*** provides a mechanism to document and manage the metadata associated with the process.



7.2 Simple Report Design Approach

Figure 2 represents the parts of the metamodel sufficient for non-financial reports that do not require complex calculations. In this approach, the report's content is represented as ***Facts*** and their corresponding ***Fact Values***, which are structured to ensure clarity and precision in the data presentation.

At the core is the ***Fact*** class, which serves as the fundamental unit of information within the report. ***Fact Sets***, which group related ***Facts*** for logical organization and ease of access. Each ***Fact Set*** is linked to the ***Report***, ensuring all relevant facts are properly categorized and associated with the report instance.

Key elements include:

1. ***Fact Value***:
    - Every ***Fact*** has a ***Fact Value***, representing the actual data point for that fact.
    - ***Fact Values*** can be further classified into:
        - ***Numeric Fact Value***: a number with associated ***Unit*** (for example representing length, weight) and optional ***rounding*** information.

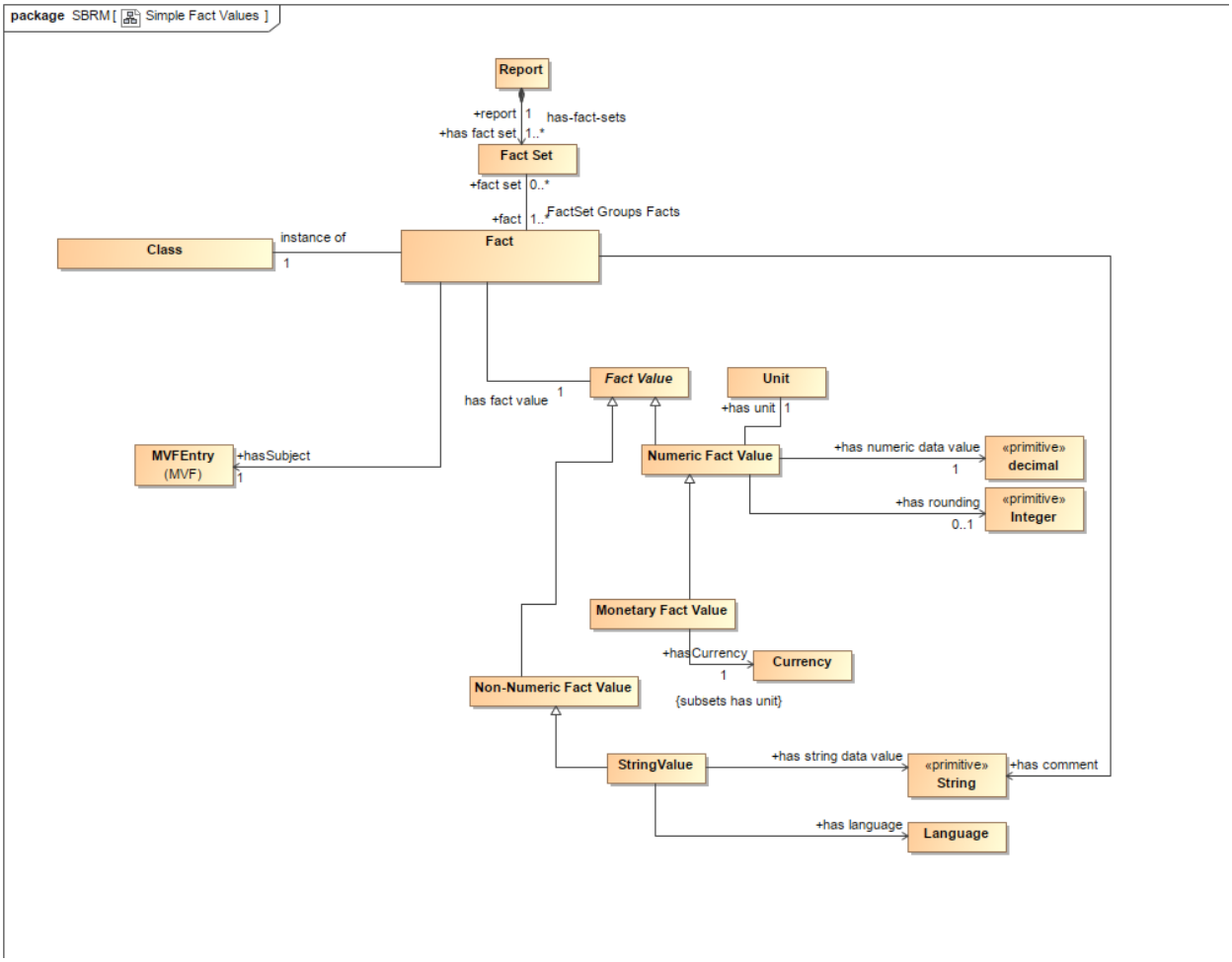
- ***Monetary Fact Value***: A specialization of ***Numeric Fact Value*** where the associated ***Unit*** is a specific ***Currency***.

- ***Non-Numeric Fact Value***: Includes text or string-based values, such as comments or descriptions, which have an associated ***Language***.

2. Relationships to Ontologies:

    - The subject of a fact is given by a ***MVFEntry*** (Multiple Vocabulary Facility Entry), a generic class from the [MVF] specification which represents a concept independent of specific ontologies or vocabularies used to represent it. This links ***Facts*** to subjects, enabling semantic alignment and ensuring that each fact can be mapped to relevant ontology concepts.

This approach provides a lightweight and flexible framework for representing non-financial report data, focusing on modularity and extensibility. By leveraging the simplicity of facts and values, it ensures broad applicability across various domains while maintaining compatibility with structured metadata and ontology frameworks.



7.3 Financial Report Design Approach

For ease of use in a modeling or ontology environment the approach is to represent each Concept (as used in XBRL) as a Class (metaclass) and use built-in rdf:type to associate the concept aspect with a Fact.
Debit/Credit and Instant/Period are represented as classes too.
Note that Class is used for the metamodel/ontology rather than Property despite it representing what is logically a property of the reporting entity. This is similar to XBRL's approach (all Facts are Elements)
SPARQL-based [SHACL] rules are used to represent arithmetic integrity/quality constraints e.g. for a Rollup the output = sum(input).
OWL restrictions are (somewhat redundantly) provided in addition to the SHACL for those working in a purely OWL environment.

The Financial Report Design Approach provides a robust framework for representing financial data in a modeling or ontology environment. It emphasizes compatibility with semantic standards (e.g., OWL, SHACL) and established practices like XBRL.

1. Representation of Concepts as Classes:

- Each Concept in the financial reporting domain, as defined in XBRL, is represented as a Class (metaclass) within the ontology. This allows for a structured and extensible representation of financial reporting elements.

- The rdf:type property associates these concepts with specific Facts, enabling semantic alignment between the data model and the reporting context.

- Additional financial constructs, such as Debit/Credit and Instant/Period, are also represented as classes, ensuring their inclusion in the semantic model.

2. Rationale for Using Classes:

    - The design leverages Classes rather than Properties for the metamodel and ontology representation. While these elements logically act as properties of the reporting entity, representing them as classes provides greater flexibility for modeling complex relationships and constraints.

    - This approach mirrors XBRL's treatment, where all Facts are treated as Elements, maintaining consistency with widely adopted financial reporting standards.

3. Arithmetic Integrity and Quality Constraints:

    - SPARQL-based SHACL rules define arithmetic integrity and quality constraints within the reporting framework. For example:

        - A Rollup operation is modeled such that the output = sum(input) relationship is explicitly enforced through SHACL.

        - These rules provide a formal mechanism to validate financial relationships, ensuring data consistency and accuracy.

4. OWL Restrictions for Compatibility:

    - While SHACL rules are the primary mechanism for representing constraints, OWL restrictions are also provided. Though somewhat redundant, these OWL-based definitions cater to users operating in OWL-only environments, ensuring compatibility across different modeling contexts.

5. Alignment with Ontology and XBRL Standards:

    - This design ensures broad applicability by aligning with XBRL's approach and incorporating both SHACL and OWL. It supports both traditional financial reporting systems and advanced ontology-driven modeling environments.

7.4 Facts for Report Submissions

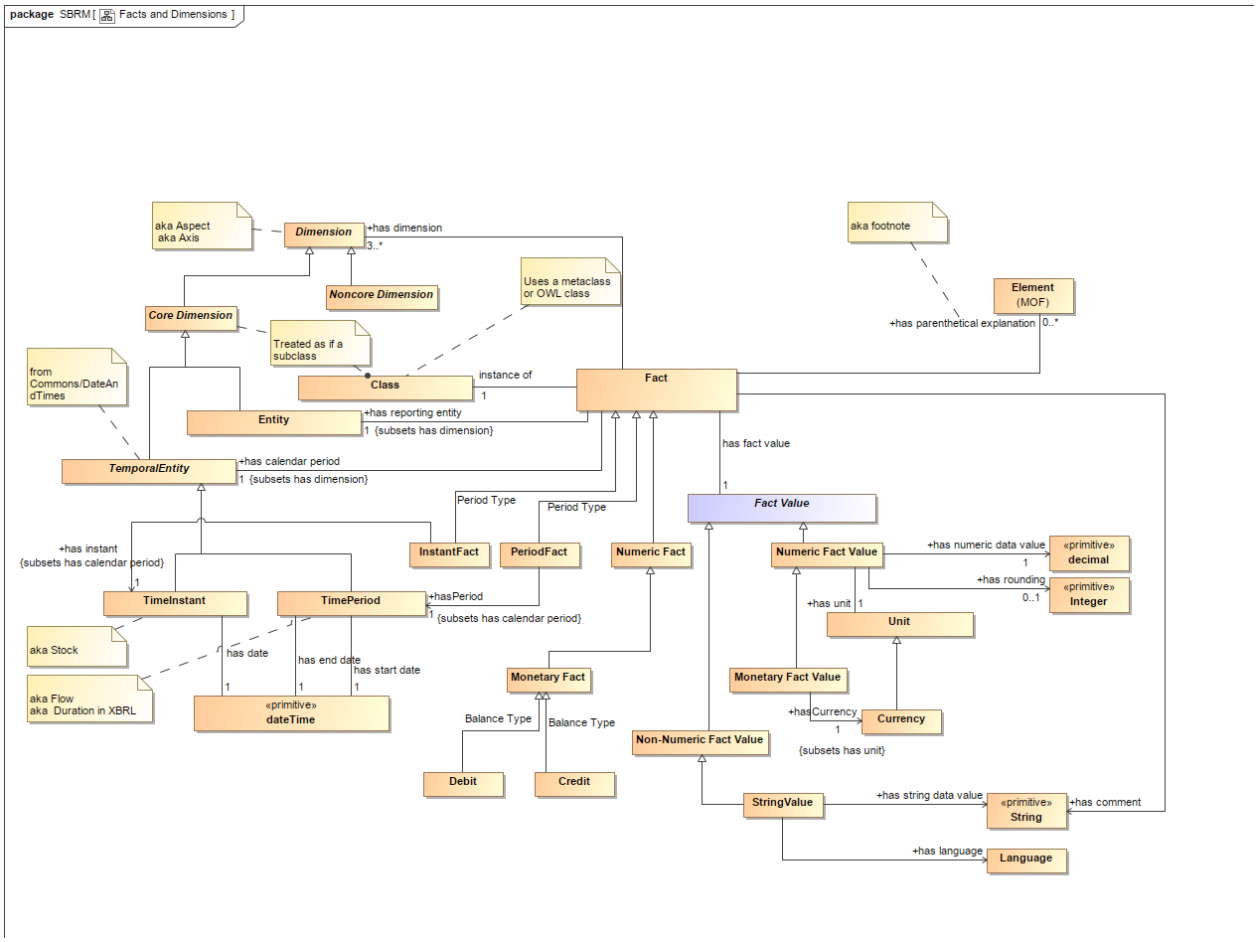
Compared to the Simple Approach in 7.2, Figure 3 captures an extended representation of ***Facts*** and their associated elements for more sophisticated report submissions where ***Facts*** have additional ***Dimensions*** in addition to their ***Values***. At the core again is the ***Fact*** class, which serves as the fundamental unit of data in a submission. Each ***Fact*** is characterized by its ***Class***, providing semantic meaning (covered in more detail in 7.5), and whether its value applies to a point in time (***InstantFact***) or a time period (***PeriodFact***); and for Monetary Facts whether they represent a ***Debit*** or a ***Credit***.

The ***Fact Value*** associated with each ***Fact*** is modeled as a distinct structure, which is covered in 7.2. This Figure 3 extended diagram also includes the relationships between Facts and their ***Dimensions***, optionally referred to as Aspects or Axes, which provide additional context. The approach of analyzing facts using Dimensions is common to data warehousing and data cubes. These ***Dimensions*** are divided into two categories:

- ***Core Dimensions***: Universal to all submissions, including the ***Entity*** (reporting organization to which the facts apply) and ***CalendarPeriod*** (time-based aspects like start and end dates).

- ***Noncore Dimensions***: User-defined dimensions that allow customization to meet specific reporting requirements. Examples of such Dimensions could include locations, products, organizational units, and they in turn could be hierarchical as detailed in 7.6.

.



7.5 Classes

Figure 4 elaborates the model for ***Class***, as introduced in 7.4. It is a fundamental building block for representing the semantics of ***Facts*** in the report submission framework that can also be seen as a core ***Dimension*** – one that provides the Concept or meaning for multiple ***Facts***.

1. Core Definition of a Class:

    - A ***Class*** encapsulates the semantic meaning of a ***Fact***, enabling precise representation and interpretation. Each ***Class*** is associated with one or more ***MVFEntries*** (Multiple Vocabulary Facility Entries), which link the ***Class*** to external vocabularies or ontologies, enriching its meaning and ensuring interoperability. Importantly, and somewhat counter-intuitively, a Class here will typically be linked via a MVFEntry to a Property in an external ontology, since it represents something that will have a specific value – for example PayrollTax, NumberOfEmployees.

2. Relationships:

- ***MVFEntry***:
    - The ***MVFEntry*** establishes the connection between a ***Class*** and its conceptual meaning, supporting alignment with external standards or domain-specific terminologies.
    - Each ***Class*** can have multiple associated ***MVFEntries***, enabling broader applicability across diverse domains.
- ***ValueType***:
    - The ***ValueType*** specifies the allowable datatypes for values associated with the ***Class,*** bearing in mind that Classes are generally associated with external properties. This ensures that ***Facts*** using a given ***Class*** adhere to defined data formats.
- ***Datatype***:
    - A Class is linked to a ***Datatype***, represented by its unique id. The Datatype defines the structure or format of values (e.g., String, Integer), enforcing consistency in how data is captured and represented.

3. Key Attributes and Functions:

- Concept Flexibility:

  Classes are designed to be flexible and reusable across multiple ***Facts***, which differ in their other ***Dimensions*** (for example the Class *TotalSales* would apply to Facts for different regions and different products). A single ***Class*** can serve as the foundation for multiple reporting elements, enhancing modularity and reducing redundancy.

- Value Constraint:

  By associating Classes with specific ValueTypes and Datatypes, the model ensures that values conform to the expected format, supporting validation and quality assurance.

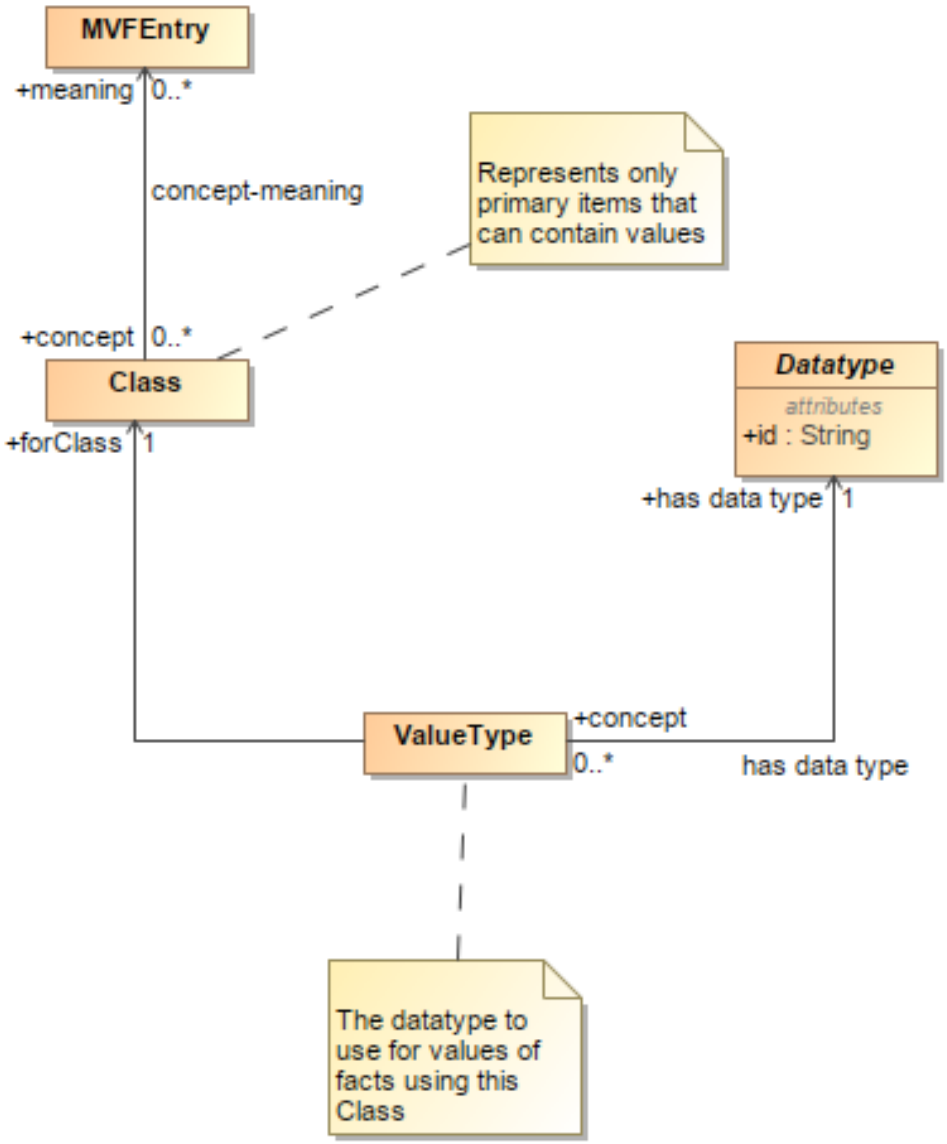


Figure 4: Classes

7.6 Report Definition - Types of Report Element

Figure 5 moves from Report Instances to Report Definitions and introduces the fundamental building blocks, or Report Elements, that comprise a report. These elements define the structure, organization, and semantics of the report, ensuring clarity and consistency in data representation. The diagram provides an overview of the types of elements within a report definition (which are expanded on in later sections) and how they're arranged hierarchically.

Core Report Elements

- ***Report Element***:
    - The most general entity in the hierarchy represents any element that can appear in a report. It is also referred to as an Item.
    - Report Elements can take various forms, depending on their role in the definition of the report.
- ***Structure Element***:
    - A subtype of Report Element represents elements that define a report's structural organization.
    - Examples include Sections, Line Items, and Hypercubes, which collectively form the report's framework.
- ***Structure***:
    - A category of Structure Element used to organize data in hierarchical or grouped forms, such as tables or line items.

Specific Types of Structure Elements

- ***Class***:
    - This was covered in 7.5 and is the most atomic type of Report Element, linked directly to Facts with Values. A Concept provides semantic meaning to individual pieces of data and is essential for aligning Facts with their definitions in a report.
    - In essence, a Class is the only Report Element that has a direct relationship with data values.
- ***Dimension***:
    - This was covered in 7.4 and represents aspects or axes that describe data, such as time, location, or other contextual details.
    - Dimensions help classify and organize data, providing additional context for the facts represented in a report.
- ***Section***:

    - Represents a logical grouping of Report Elements, such as chapters, headings, or specific divisions in the report.

- ***Line Items***:

    - A subset of Structure Elements representing rows in a table or lists of items under a specific heading.

- ***Hypercube***:

    - A specialized structure that defines a multidimensional space for organizing data. This is equivalent to the cube structure often seen in traditional data warehouses.

    - Attributes:

        - ***isClosed***: A Boolean value indicating whether the Hypercube's ***Dimensions*** are complete (e.g., all combinations of members are present).

    - Hypercubes are critical for handling complex data arrangements, such as financial reports with multiple dimensions.

- ***Member***:

    - Represents an individual element within a ***Dimension***, covered in more detail in 7.8.

    - Attributes:

        - ***isHierarchyAdditive***: A Boolean value indicating whether ***Members*** can be summed hierarchically (for example this is normally the case in a regional hierarchy of district, county, state, country).

- ***Domain***:

    - Represents a specific type of ***Member***, often called All X, to indicate all possible members within a specific category or grouping.

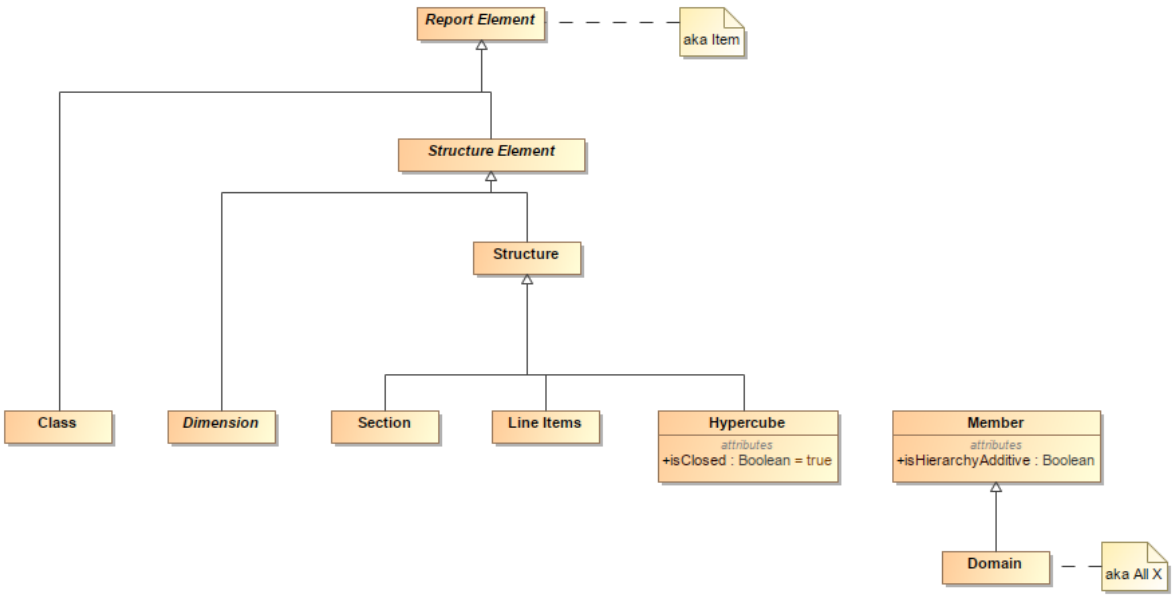


Figure 5: Types of Report Element

7.7 Top Level Structure

Figure 6 captures the hierarchical framework for organizing and categorizing elements within a ***Publication***. This structure emphasizes the relationships between the overarching ***Publication***, its constituent ***Reports***, and additional organizational elements such as ***Topics*** and ***Fragments***. ***Publication*** and ***Report*** were already covered in 7.1.

- ***Publication***
    - A ***Publication*** represents the top-level entity in this framework, encompassing everything that might be submitted as part of a reporting process.
    - Each ***Publication*** can include multiple Reports, serving as a container for the individual information units.
- ***Reports***
    - A ***Report*** is a core entity within a ***Publication***, defined by attributes such as ***hasReportDate*** and ***hasReportDateTime***
    - ***Reports*** may have additional metadata linked to ***Publication Process Information*** to capture submission details.

- ***Report Level Rules***:
    - Reports are associated with ***Report Level Rules***, which define validation or structural constraints at the report level (e.g., arithmetic rules, structural requirements).
- ***Topics***
    - ***Topics*** are optional organizational elements used for:
        - Categorizing different parts of a report.
        - Organizing information into logical sections.
    - A ***Topic*** can have:
        - ***SubTopics***: Representing a hierarchical organization of content.
        - ***Parent Topics***: Allowing for nested or related topics to be defined within the structure.
- ***Fragments***
    - ***Fragments*** provide further modularization within the structure. They allow for breaking down a report into smaller, reusable parts.
    - ***Fragments*** can:
        - Reference other Fragments (using ***refers-Fragments***).
        - Contain ***Fact Sets*** to group related data (already covered in 7.2).
        - Include Arrangement Patterns, such as:
        - ***Concept Arrangement Patterns***: To define the logical relationships between ***Facts***. See 7.9.
        - ***Member Arrangement Patterns***: To structure relationships within ***Dimensions***. See 7.10.
- ***Fact Sets***
    - ***Fact Sets*** are groupings of individual ***Facts*** within a ***Fragment***. These groupings:
        - Enable the logical organization of related data points.

- Are tied to the ***Concept Arrangement Patterns*** and ***Member Arrangement Patterns***, which define the semantics of the relationships within the group.

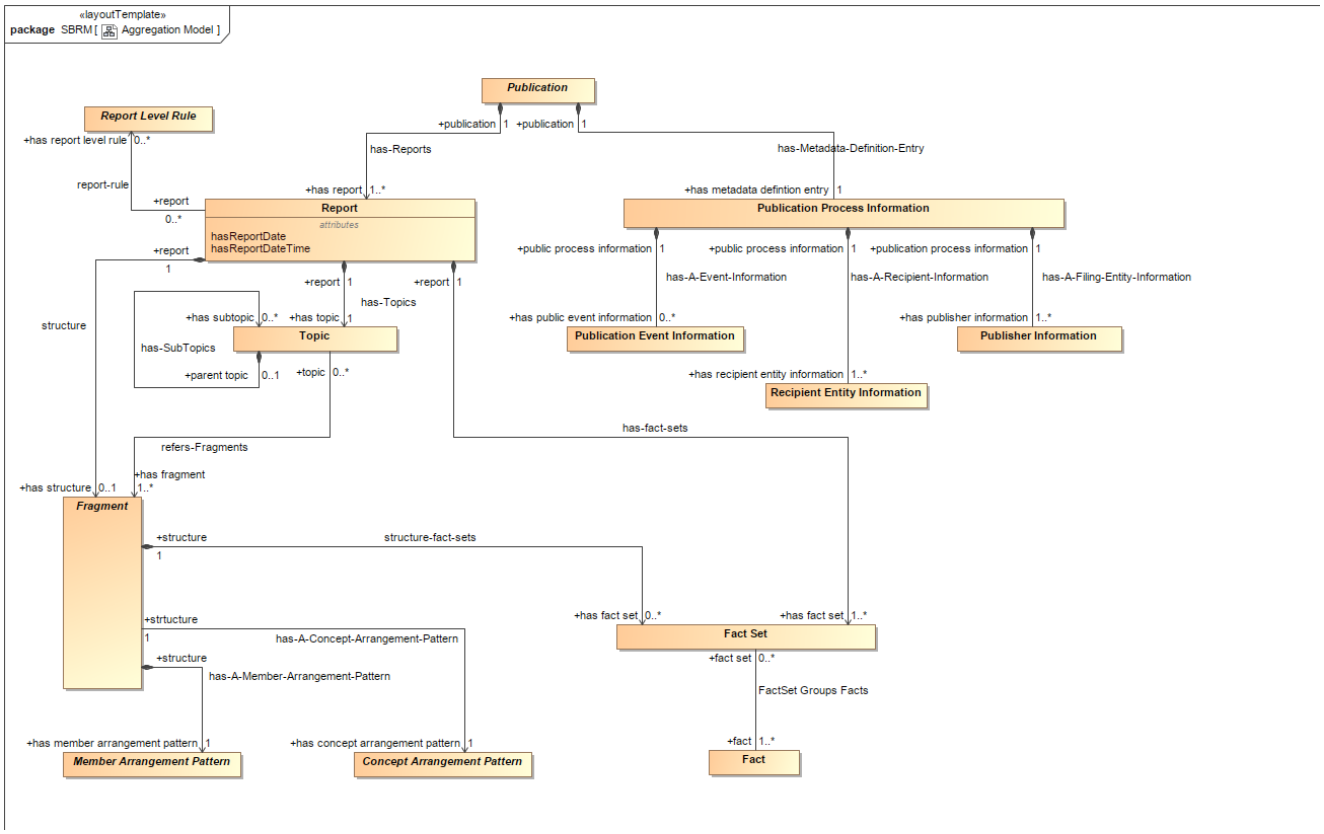
- Supporting Entities

  - ***Publication Process Information***: This section links the ***Publication*** to the metadata, recipient information, and events related to the submission process.

  - ***Publication Event Information***: Provides when the report is produced (e.g., quarterly, after a type of corporate action), ensuring clarity in the submission process.

  - ***Recipient Entity Information***: Provides details about who the report is intended for, ensuring clarity in the submission process.

  - ***Publisher Information***: Provides details about who publishes the report and how (e.g., the SEC on the EDGAR site).



7.8 Report Structure

Figure 7 captures the hierarchical organization of elements within a report definition, emphasizing the whole-part relationships that define how the different structural components interact. This diagram illustrates these relationships and explains how each element contributes to the report's framework.

- ***Report Element***
    - The ***Report Element*** is the top-level abstraction for structural components in a report. It can represent any individual item that forms part of the report's hierarchy.
    - It serves as a foundation for all subordinate Structure Elements.
- ***Structure Element***
    - A **Structure Element** represents any intermediate or final component in the report's framework, such as ***Sections***, ***Tables***, or ***Line Items***. See also 7.6 which has a further breakdown, including ***Dimension***.
    - These elements organize and group content within the report, ensuring modularity and reusability.
    - A ***Structure Element*** can be a ***Structure*** that provides the report hierarchy.
- ***Structure***
    - The ***Structure*** class defines how the ***Structure Elements*** are organized or linked to one another.
    - It includes:
        - ***Structure Associations***: Represent relationships between structure elements (e.g., "has part" or "is part of").
        - These associations support attributes such as sequence, which determines the structure's order or hierarchy of elements.
- ***Structure Association***
    - A ***Structure Association*** defines the relationships between structural components.
    - Example: Linking sections to subsections or tables to their respective line items.

- Attributes:
    - ***sequence***: Specifies the order of elements within a structure, allowing for sequential organization or aggregation.

- ***Member Arrangement Pattern***
    - This class represents patterns for arranging Members within a structure and is covered in detail in 7.9.

- ***Member***
    - A Member represents an individual element within a Member Arrangement Pattern.
        - Example: A specific data field, column, or grouping within a table.
        - Attributes:
            - ***isHierarchyAdditive***: A Boolean attribute indicating whether the member's values can be summed hierarchically.

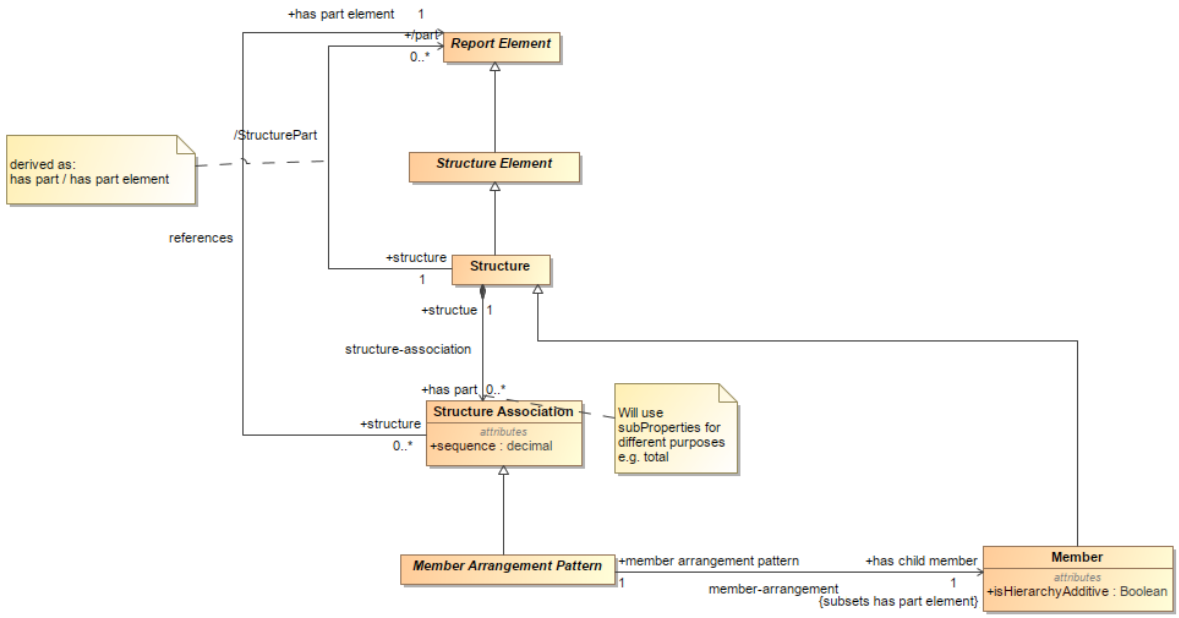


Figure 7: Report Structure

7.9 Concept Arrangement Patterns

Figure 8 provides an overview of predefined patterns that overlay simple calculation structures with additional whole-part relationships and integrity rules. Each of the subclasses is described in its own diagram in the following sections, except for TextBlock which can be elaborated using the Document Components and SPAR ontologies.

These patterns are essential for organizing financial reports in a structured, consistent, and semantically meaningful way. They also enforce rules for combining specific elements, such as time periods and balance types, to ensure data quality and validity.

Focus on Financial Reports

These patterns are tailored to support common financial reporting structures and adhere to best practices outlined in the [XBRL book], rather than supporting unrestricted or arbitrary arrangements. By following widely accepted standards, this approach ensures:

- Alignment with Industry Practices: The patterns facilitate reporting consistency across organizations and jurisdictions.
- Interoperability: Reports structured using these patterns can be more easily integrated into reporting systems and analytical tools.
- Data Integrity: The constraints imposed by the patterns enhance the logical consistency of report components.

Pattern Overview

These patterns enhance the basic calculation structure by introducing rules and relationships that govern:

1. How data is arranged within a report: Each pattern specifies the structure and organization of report elements.
2. What types of periods and balance types can be combined? Rules ensure the compatibility and semantic validity of combined data.
3. The logical consistency and integrity of report components: Constraints prevent errors and misalignments in reporting.

The specific ***Concept Arrangement Patterns*** are as follows:

- ***Set***:
    - A simple grouping of related concepts or facts without requiring any specific arithmetic relationships.

- ***Roll Up***:
    - Represents a summation relationship where individual items (e.g., line items) roll up into a total.
    - Example: Summing expenses across categories to produce a total expense.
- ***Roll Forward***:
    - Tracks change over time, starting with an opening balance, adding changes (e.g., income or expense), and ending with a closing balance.
    - Example: Cash flow statements.
- ***Roll Forward Info***:
    - Add explanatory details or metadata to a roll forward, such as descriptions or supporting notes.
- ***Adjustment***:
    - Represents changes or corrections applied to data, such as restatements or journal adjustments.
- ***Variance***:
    - Compares two data points to calculate differences, often used for budget vs. actual comparisons.
    - Example: Comparing forecasted vs. actual revenue.
- ***Arithmetic Expression***:
    - Represents a formula or rule that defines a calculation beyond simple summation or comparison.
    - Example: Profit = Revenue - Expenses.
- ***TextBlock***:
    - Represents narrative or textual content within a report, such as notes, policies, or disclosures.
    - **Attributes**:

- ***TextBlocks*** can include formatted content and often serve as containers for descriptive text.

- Specific levels or types of ***TextBlock*** (e.g., Notes, Policies, Disclosures) are standard in financial reporting.

Key Features of the Patterns

- ***Whole-Part Structure***:

  - The patterns rely on a hierarchical structure in which smaller components aggregate into larger, more comprehensive elements (e.g., line items ***roll up*** into totals).

- ***Integrity Rules***:

  - Each pattern is associated with integrity rules that govern the data types and relationships allowed within the pattern. These rules ensure consistency and validity.

  - Example Rules:

    - Time periods must align in ***Roll Forward*** patterns.

    - Balance types (e.g., ***debit*** vs. ***credit***) must be consistent across calculations.

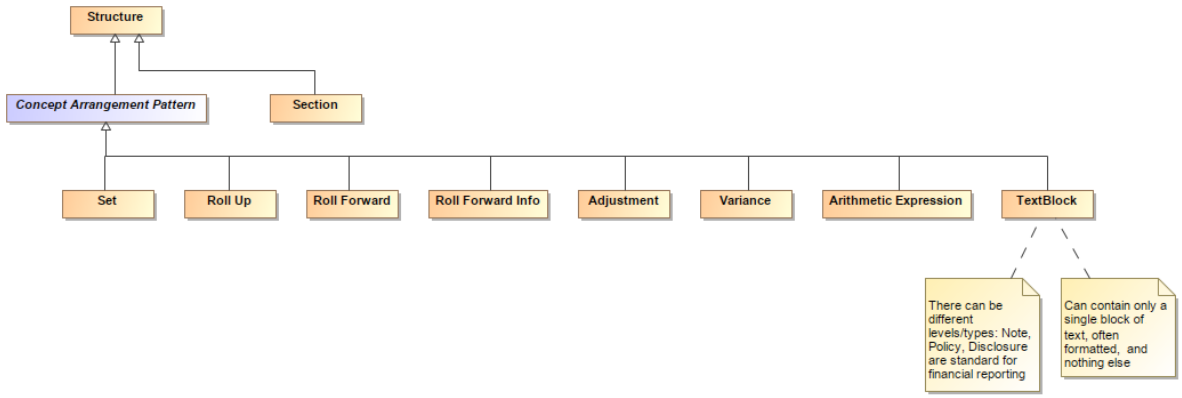


Figure 8: Concept Arrangement Patterns

7.9.1 Set Concept Arrangement Pattern

Figure 9 represents a structural pattern that organizes a sequence of elements without imposing any numeric or computational relationships between them. This pattern is particularly useful for grouping related elements hierarchically or sequentially, emphasizing organization over calculation.

Key Features

1. Non-Numeric Relationships:
    - Unlike other patterns (e.g., ***Roll Up*** or ***Arithmetic Expression***), the ***Set Pattern*** does not involve any mathematical or logical computations.
    - Its purpose is purely organizational, creating a structured sequence of related elements.
2. Sequence Arrangement:
    - The pattern arranges elements in a specific sequence, represented by the ***SetMember*** class.
    - The ***sequence*** attribute (an integer) specifies the order of elements within the set, ensuring that the hierarchy or sequence is clearly defined.
3. Member Elements:
    - Each ***Set*** contains one or more ***SetMembers*** (1..* cardinality), which are individual elements of the set.
    - Each ***SetMember*** references exactly one ***Report Element***, linking the member to its corresponding structural or content component.
4. Exclusions:
    - Not a TextBlock: The Set Pattern differs from ***TextBlocks*** used for narrative or descriptive content.
    - No Mathematical Computations: This pattern is not used for numeric relationships like summation, comparison, or roll-forward logic.

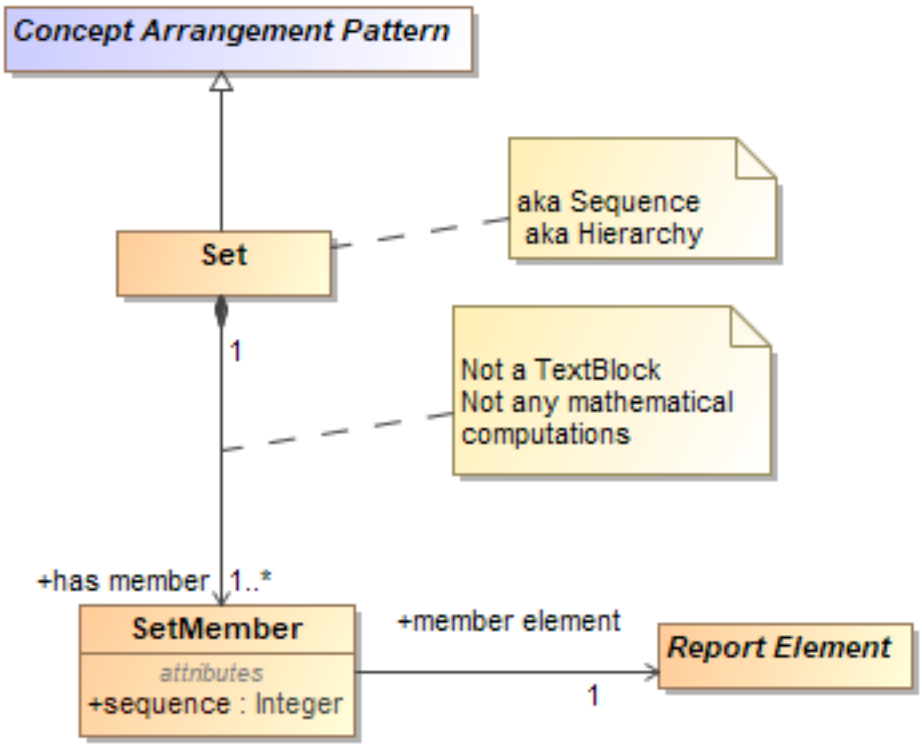


Figure 9: Set Concept Arrangement Pattern

7.9.2 Roll Up Concept Arrangement Pattern

In this pattern sequence of inputs are accumulated (added or subtracted depending on the property isAdded) to form a total for the same period.

Figure 10 represents the foundational structure for organizing and calculating numeric data within financial and other structured reports. This pattern accumulates a sequence of numeric inputs, adding or subtracting them based on their properties to calculate a total for the same time period. It ensures the consistency and accuracy of aggregated data, aligning with best practices for reporting.

Key Features

- Accumulation of Inputs:
    - This pattern aggregates multiple inputs to produce a total value.
    - The ***isAdded*** attribute determines the accumulation logic (addition or subtraction) for the specific input value:
        - ***isAdded*** = true: The value is added to the total.
        - ***isAdded*** = false: The value is subtracted from the total.
- Sequence and Order:
    - The ***sequence*** attribute of ***Numeric Input*** specifies the order of inputs, ensuring that the aggregation follows a defined progression.
- Numeric Consistency:
    - All inputs in a ***Roll Up*** must share the same ***Dimensions*** (e.g., time period, unit of measure) and datatype to ensure consistency and avoid mismatched calculations.
- Numeric Concept:
    - The ***Numeric Concept*** class represents the meaning of all the numeric values.

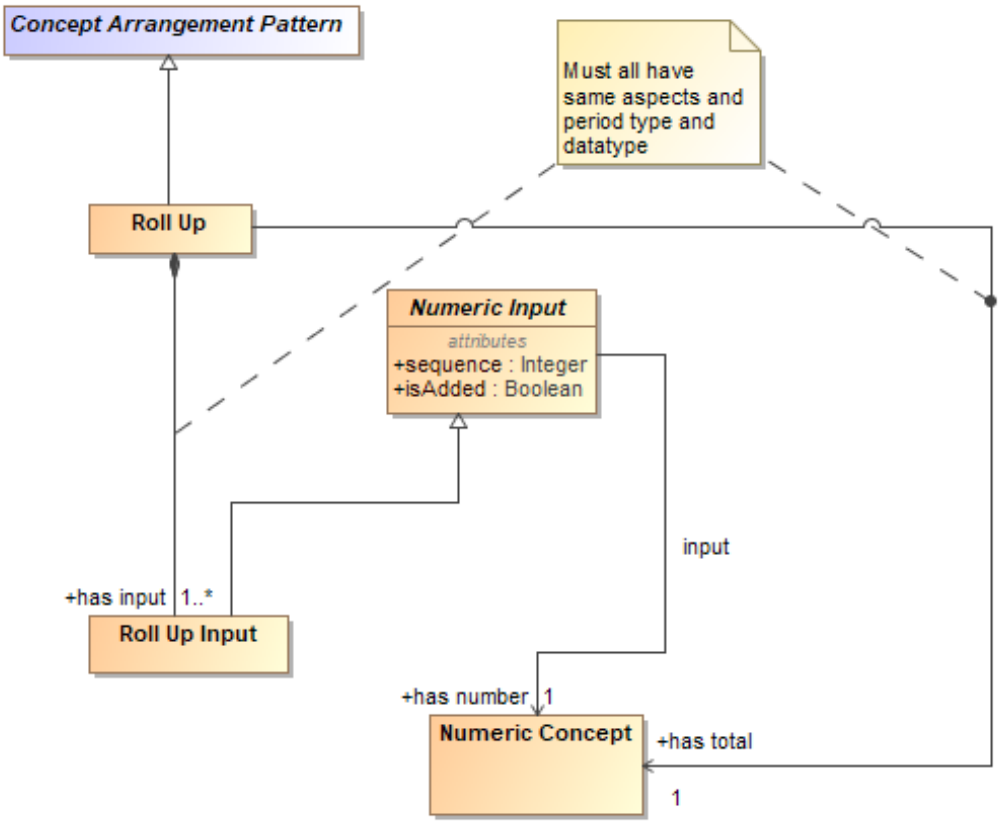


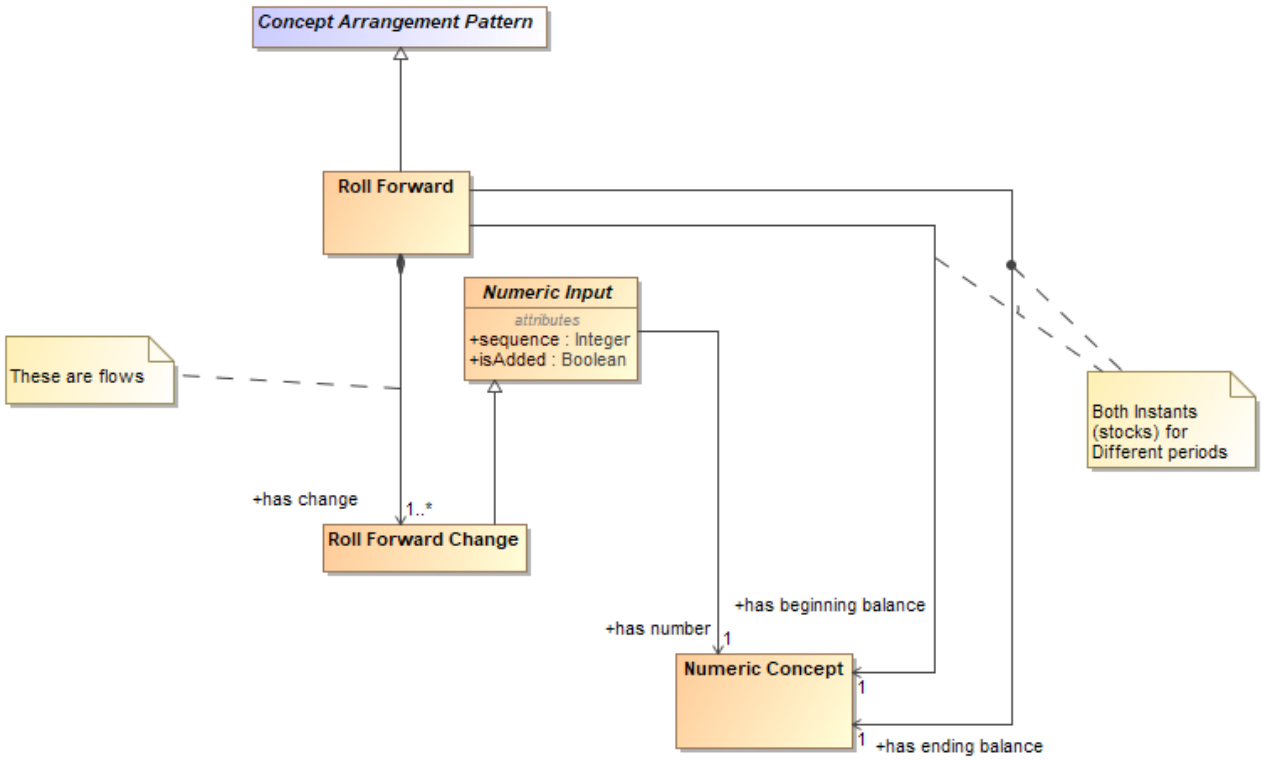
Figure 10: Roll Up Concept Arrangement Pattern

7.9.3 Roll Forward Concept Arrangement Pattern

Figure 11 represents a design to track changes over time by linking beginning balances, incremental changes (inflows and outflows), and ending balances for a given period. This pattern is essential in financial and operational reporting to reconcile values across time periods, ensuring logical consistency and data traceability.

Key Features

- Flows and Stocks:
    - This pattern captures flows (changes over time) and stocks (values at specific instants).
    - **Flows** represent changes within a given period (e.g., revenues, expenses, adjustments).
    - **Stocks** represent balances at specific instants, such as the beginning and ending balances of a reporting period.
- Structure and Components:
    - ***Roll Forward Change***: Represents one or more incremental changes (e.g., inflows or outflows) during the reporting period.
    - ***Numeric Input***: The underlying data element for flows, with attributes:
        - ***sequence***: Determines the order of changes.
        - ***isAdded***: Boolean that specifies whether a change is an inflow (added) or an outflow (subtracted).
    - ***Numeric Concept***: Represents the meaning of each number.
- Period Alignment:
    - The ***beginning balance*** and ***ending balance*** are stocks captured as **instants**, ensuring alignment with specific reporting periods.
    - The flows between these instants (represented by ***Roll Forward Change***) define these balances' differences.


Concept Arrangement Pattern
Roll Forward
Numeric Input
attributes
+sequence : Integer
+isAdded : Boolean
These are flows
Both Instants (stocks) for Different periods
+has change
1..*
Roll Forward Change
+has beginning balance
+has number
1
Numeric Concept
1
1
+has ending balance

7.9.4 Roll Forward Info Concept Arrangement Pattern

Figure 12 represents a specialized variation of the Roll Forward Concept Arrangement Pattern, with a key distinction: it does not incorporate any numeric changes or flows. Instead, this pattern provides additional contextual or explanatory information associated with a beginning balance and an ending balance for a specific reporting period.

Key Features

- No Changes:
    - Unlike the ***Roll Forward Pattern***, this pattern excludes numeric changes (e.g., inflows, outflows, or adjustments).
    - It focuses solely on the beginning and ending balances, serving as a static representation of these values over time.
- Informational Context:
    - This pattern is typically used to present explanatory or descriptive information that adds meaning to the reported balances.
    - For example, it may include supporting details, metadata, or narratives relevant to the balances.
- Structure and Components:
    - ***Roll Forward Info***: As the pattern's primary container, linking the balances and associated information.
    - ***Numeric Concept***: represents the meaning of all the numeric values.

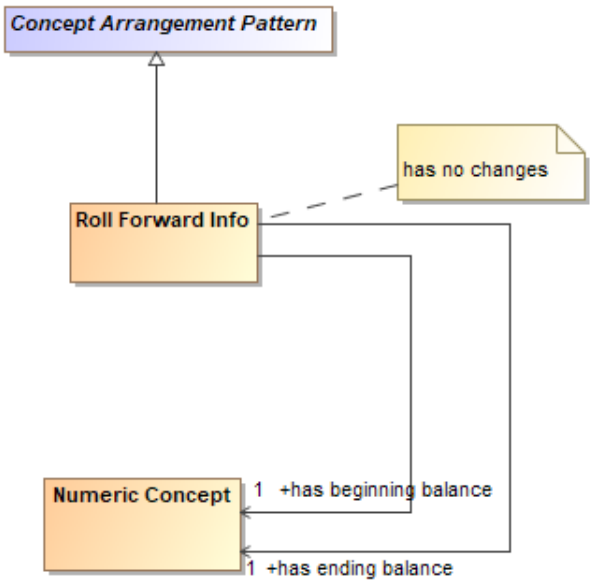


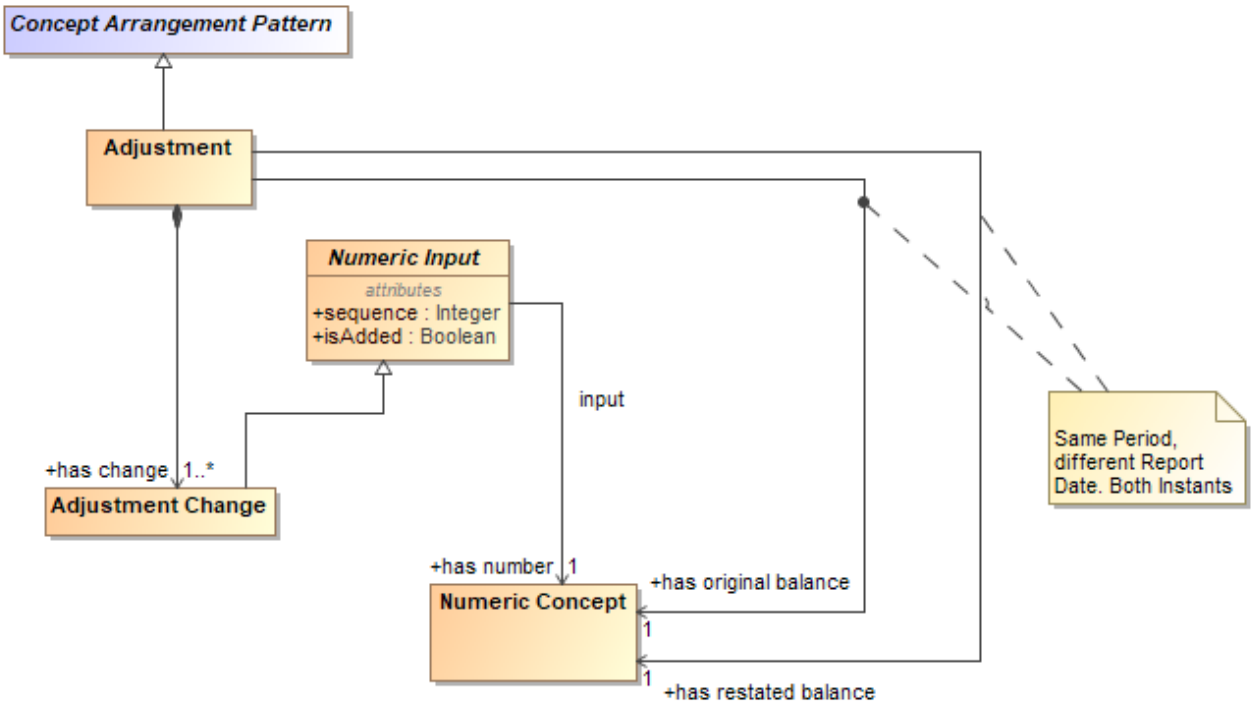
Figure 12: Roll Forward Info Concept Arrangement Pattern

7.9.5 Adjustment Concept Arrangement Pattern

Figure 13 represents the design to handle corrections or adjustments to an original numeric value, resulting in a restated (corrected) value for the same reporting period. This pattern is particularly useful for financial and operational reporting when errors or retroactive changes need to be captured and reconciled in a structured manner.

Key Features

- Corrections to Original Data:
    - This pattern adjusts an ***original balance*** using one or more ***Adjustment Changes*** (e.g., corrections, restatements).
    - The final result is a ***restated balance*** for the same reporting period.
- Structure and Components:
    - ***Adjustment***: The central pattern that organizes and manages the correction process.
    - ***Adjustment Change***: Represents individual corrections applied to the original balance.
    - ***Numeric Input***: The underlying data for the adjustment, with attributes:
        - ***sequence***: Specifies the order in which adjustments are applied.
        - ***isAdded***: A Boolean indicating whether the adjustment is positive (added) or negative (subtracted).
    - ***Numeric Concept***: Represents the meaning of all of the numeric values.
- Same Reporting Period:
    - The ***original balance*** and the ***restated balance*** apply to the same reporting period but may be reported at different times (e.g., after corrections are made).


Concept Arrangement Pattern
Adjustment
Numeric Input
attributes
+sequence : Integer
+isAdded : Boolean
input
+has change
1..*
Adjustment Change
+has number
1
Numeric Concept
+has original balance
1
1
+has restated balance
Same Period, different Report Date. Both Instants

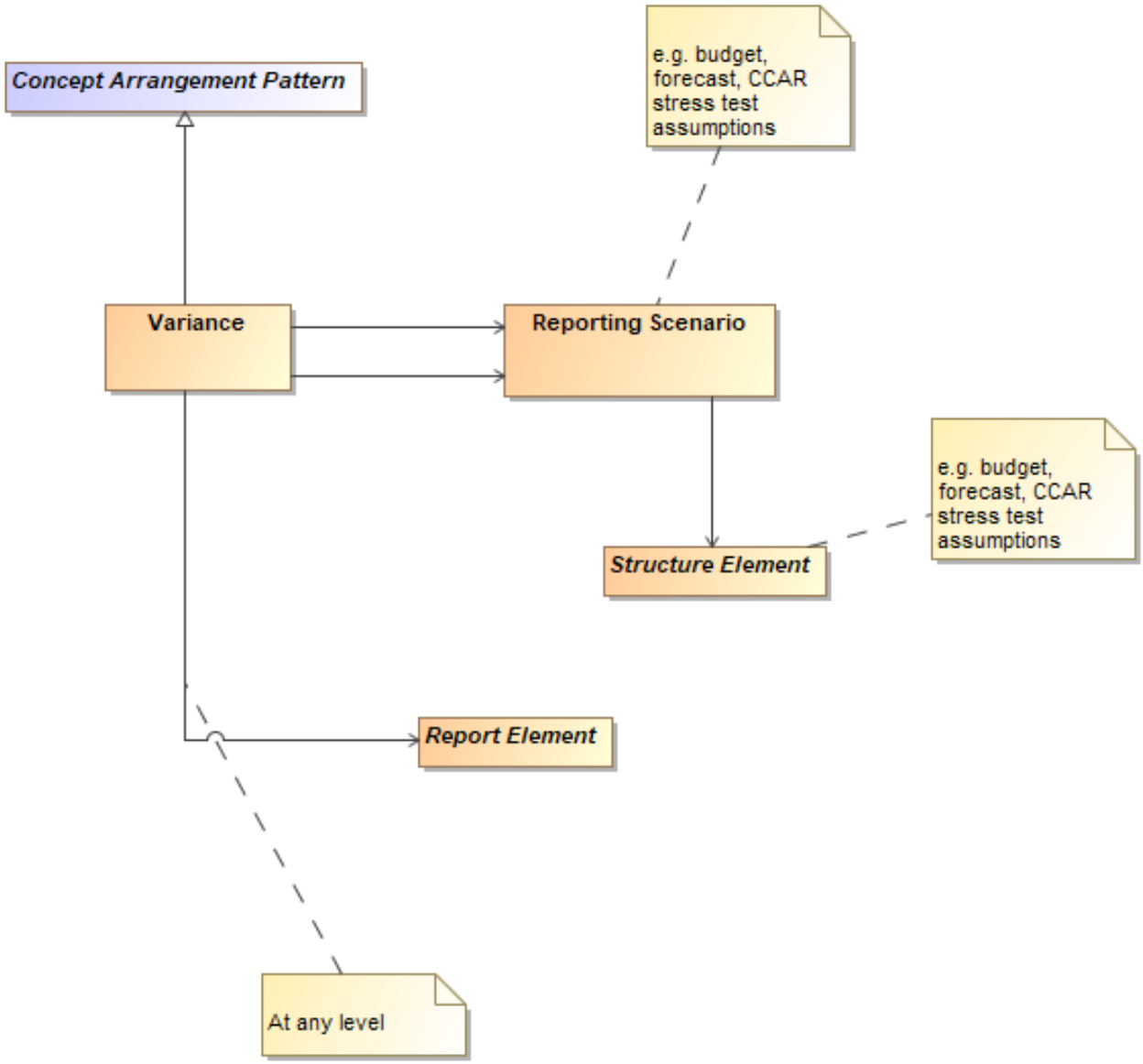
7.9.6 Variance Concept Arrangement Pattern

The pattern in Figure 14 facilitates the comparison of elements or structures across two distinct scenarios or assumptions. This pattern is particularly useful for identifying differences between projected and actual values or contrasting alternative outcomes under various assumptions. It provides a structured way to analyze and report financial, operational, or strategic variances.

Key Features

- Comparison Across Scenarios:
    - This pattern evaluates the same element (e.g., a ***Report Element***) or structure under two different ***Reporting Scenarios***.
    - Common examples include:
        - Budget vs. Actual: Analyzing differences between planned and realized results.
        - Forecast vs. Actual: Comparing predictions to outcomes.
        - Stress Test Assumptions: Evaluating variances under hypothetical conditions.
- Structure and Components:
    - ***Variance***: The central concept organizes the comparative analysis.
    - ***Reporting Scenario***:
        - Represents the context or condition (e.g., budget, forecast, actual) for the comparison.
        - Allows alignment of assumptions across different ***Structure Elements***.
    - ***Structure Element***:
        - Represents the hierarchical arrangement of ***Report Elements*** subject to comparison.
    - ***Report Element***:
        - The individual data points or components which differ between the scenarios, which may exist at any level of the reporting hierarchy.
- Flexibility Across Reporting Levels:

- The pattern can be applied at any level of the reporting hierarchy, enabling detailed or high-level variance analysis.

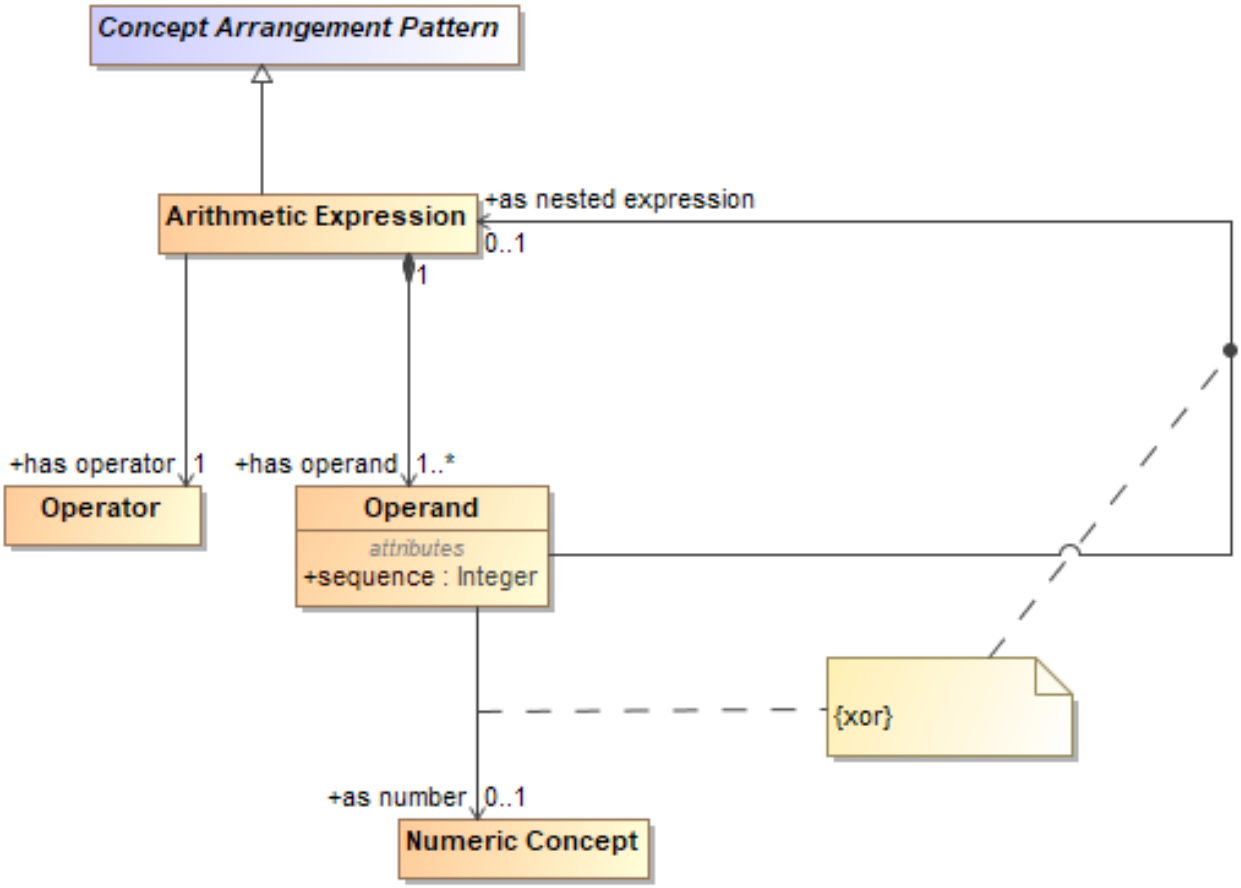


7.9.7 Arithmetic Expression Concept Arrangement Pattern

Figure 15 models mathematical calculations within a structured reporting environment. This pattern is essential for financial and business reporting, where derived metrics rely on formulas. The structure supports numeric concepts, operators, and operands while ensuring logical consistency and validation through constraints.

Key Features

- Central Expression Structure:
    - The ***Arithmetic Expression*** serves as the core construct for representing mathematical operations. It can include simple calculations or nested expressions for hierarchical operations.
    - Nested expressions are supported through the ***as nested expression*** relationship, enabling multi-step calculations.
- Operator and Operand Integration:
    - ***Operators*** specify the type of operation (e.g., addition, subtraction, multiplication, or division) defined through the ***has operator*** relationship.
    - ***Operands*** represent inputs for operations, which can be constants, numeric concepts, or nested expressions.
    - Operands include a ***sequence*** attribute, which orders the inputs to ensure correctness in multi-input calculations.
- Numeric Concept Usage:
    - Numeric values, either as operands or results, are represented by Numeric Concept elements. These values are identified via the ***as number*** relationship.
- Constraint Validation:
    - The pattern enforces logical consistency through constraints such as {xor}, ensuring that either a Numeric Concept or another Arithmetic Expression is used as an operand, but not both simultaneously.
- Flexibility for Complex Calculations:
    - Supports both basic arithmetic formulas (e.g., Profit = Revenue - Expenses) and nested, multi-layered computations (e.g., Net Income = (Total Revenue - Total Expenses) / Total Assets).


Concept Arrangement Pattern
Arithmetic Expression
+as nested expression
0..1
1
+has operator
1
+has operand
1..*
Operator
Operand
attributes
+sequence : Integer
{xor}
+as number
0..1
Numeric Concept

7.10 Hypercube Structure

Hypercubes, or Multi-Dimensional Tables, shown in Figure 16, are specialized structures that organize and analyze data across multiple ***Dimensions***. They extend the general whole-part structure by incorporating dimensions, or axes, that define the context for data elements. These dimensions provide the framework for arranging data hierarchically and enable sophisticated reporting and analysis. ***Hypercubes*** are particularly useful in financial and operational reporting, where data relationships and aggregations must be precisely structured and maintained. Each Hypercube consists of ***Line Items*** (the core data elements) and a set of ***Dimensions***, which can either be explicitly defined or dynamically generated through ***Datatypes***.

Key Features

- Multi-Dimensional Structure:
    - ***Hypercubes*** organize data along multiple axes or ***Dimensions***, each representing a unique aspect of the data, such as time, location, or product category.
- ***Line Items***:
    - The ***Dimensions*** arrange and contextualize the core reportable elements within the table, such as revenue, expenses, or sales figures.
- Dimension Categories:
    - ***Explicit Dimensions***: Defined by a finite, named set of ***Members*** (e.g., regions or departments).
    - ***Typed Dimensions***: Use custom ***Datatypes*** (e.g., dates or numerical ranges) for dynamic or infinite ***Members***.
- Hierarchical Member Relationships:
    - ***Members*** within ***Dimensions*** can be arranged hierarchically, supporting roll-ups or aggregations. For example, individual sales data can be aggregated to regional totals and then to a global total.
    - The ***isHierarchyAdditive*** attribute ensures that Members contribute arithmetically to their parent level.
- Support for Aggregation:

        - Fact Values associated with Members can be aggregated to higher levels within a Dimension, often culminating in an All Member representing the complete set for that Dimension.

- Domain Definition:

    - ***Explicit Dimensions*** are tied to a ***Domain***, which defines the total population of possible ***Members*** for a ***Dimension***, ensuring data completeness and consistency.

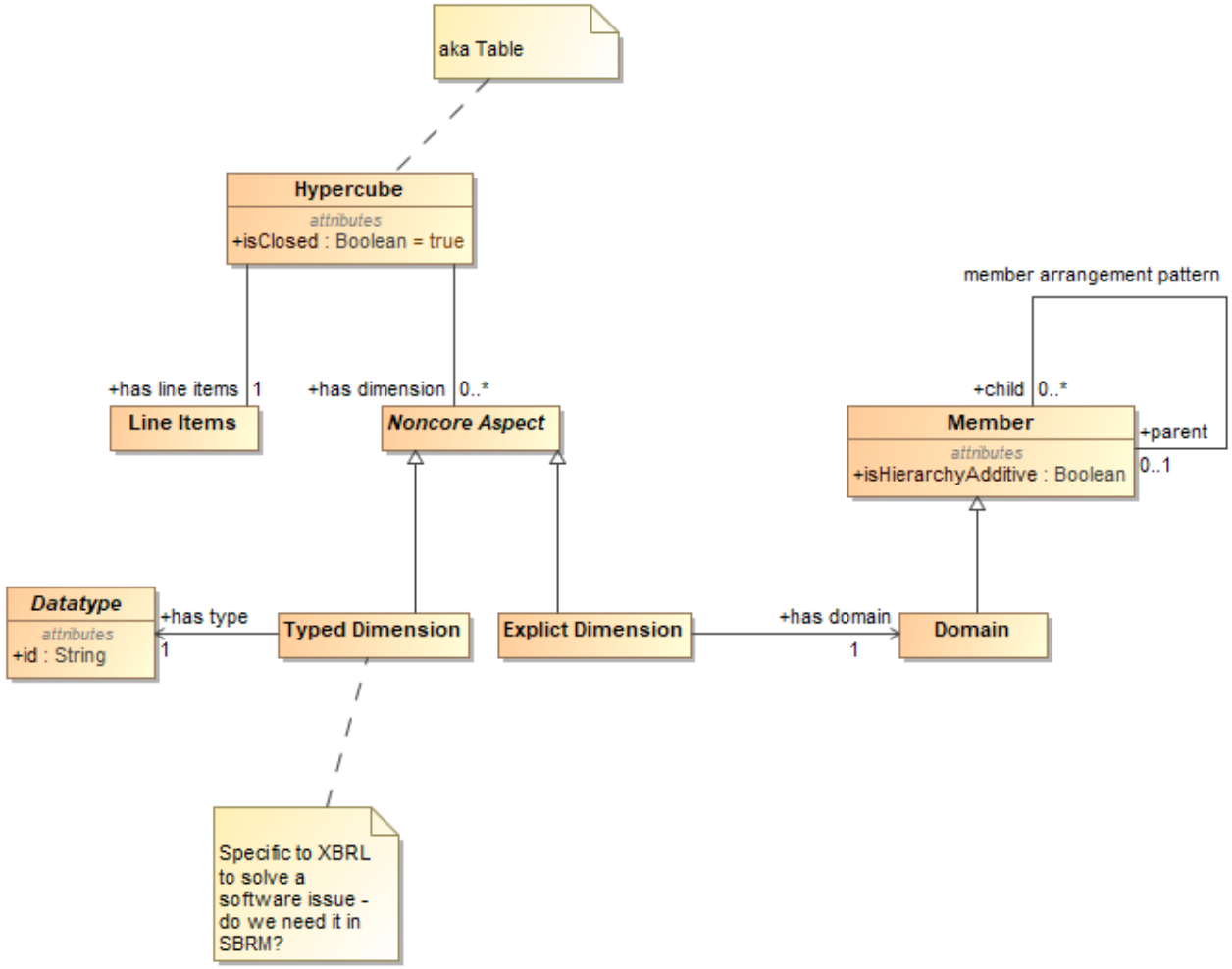
- Flexibility Through Typed Dimensions:

    - Using custom ***Datatypes***, ***Typed Dimensions*** provide flexibility in dynamically defining ***Members***, expanding the applicability of ***Hypercubes*** to diverse reporting scenarios.

- Attribute for Closure:

    - The ***isClosed*** attribute specifies whether the Hypercube contains a complete set of dimensions critical for ensuring data integrity in certain reporting use cases.

By leveraging these features, Hypercubes offer a powerful and flexible mechanism for organizing and presenting multi-dimensional data while ensuring consistency, accuracy, and compliance with reporting standards.



7.11 Datatypes

Figure 17 shows SBRM's specification comprehensive set of general Datatypes that extend beyond traditional XML-oriented types (whose names start ***xsd***). These Datatypes support diverse reporting needs, including numerical and non-numerical data. While foundational Datatypes such as numeric values, strings, and booleans remain central, SBRM introduces specialized types like ***shares***, ***percentages***, and ***text*** blocks for broader applicability. The Datatypes framework is extensible, allowing additional custom types to be defined as needed.

Key Features

- Generalized Datatypes:
    - Moves beyond XML-specific constraints, offering a broader and more versatile foundation for data representation.
    - Enables reuse in non-XML-based applications or systems.

- Classification:
    - Numeric Datatypes:
        - Includes basic XML (***xsd***) types like ***Integer***, ***Decimal***, and ***Float*** and domain-specific types like ***monetary*** values, ***shares***, and ***per-share*** amounts.
        - ***Fractional*** representation (with attributes ***numerator*** and ***denominator***) allows precise mathematical modeling.
    - NonNumeric Datatypes:
        - Covers standard types such as S***tring*** and B***oolean*** alongside specialized representations like ***TextBlock***, ***XML***, and ***JSON***.
- Extensibility:
    - The framework allows custom types to be added to support evolving business or domain-specific requirements.
- Precision and Flexibility:
    - ***Numeric DataTypes*** include attribute ***rounding*** for precision.
    - Specialized types (e.g., ***TextBlock*** and ***JSON***) cater to narrative and structured data requirements.
- Cross-Domain Support:
    - Datatypes like ***Monetary*** values and ***Shares*** are tailored for financial and economic domains, while general-purpose types like ***String*** and ***Boolean*** address universal needs.

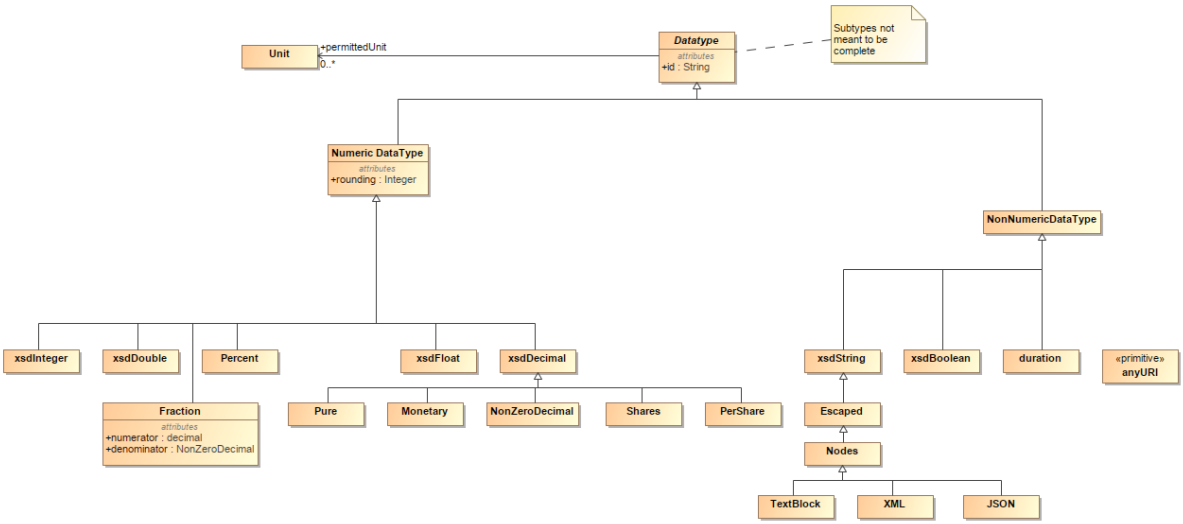


Figure 17: Datatypes

8. Ontology

The metamodel is also (more rigorously) represented as a set of ontologies, expressed in the Web Ontology Language (OWL). To provide additional semantic validation, optional constraints are expressed separately using SHACL (Shapes Constraint Language).

The ontology representation provides a more formalized structure than the UML diagrams in previous chapters, supporting reasoning, automated validation, and semantic interoperability across diverse applications. The files containing these ontologies are machine-readable and modular, allowing users to apply only the necessary components for their specific context.

8.1 Namespace Definitions for External Ontologies

The namespaces and prefixes corresponding to external elements required for use in the SBRM ontologies defined herein are provided in Table 8-1. These namespaces provide the foundational vocabularies and structural elements required for building and using the SBRM ontologies. The namespaces include OMG's Commons ontologies (e.g., cmns-av, cmns-doc), which provide foundational building blocks for modeling annotations, documents, and other elements.

The prefixes provided in Table 8-1 are normative, and their use is required in any conformant application or extension.

Table 8-1 Prefixes and Namespaces for Referenced/External Vocabularies

Namespace Prefix	Namespace
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
sh	http://www.w3.org/ns/shacl#
xsd	http://www.w3.org/2001/XMLSchema#
dct	http://purl.org/dc/terms/
skos	http://www.w3.org/2004/02/skos/core#
cmns-av	https://www.omg.org/spec/Commons/AnnotationVocabulary/
cmns-col	https://www.omg.org/spec/Commons/Collections/
cmns-cxtdsg	https://www.omg.org/spec/Commons/ContextualDesignators/
cmns-doc	https://www.omg.org/spec/Commons/Documents/

cmns-docc	https://www.omg.org/spec/Commons/DocumentComponents/
cmns-dsg	https://www.omg.org/spec/Commons/Designators/
cmns-dt	https://www.omg.org/spec/Commons/DatesAndTimes/
cmns-pts	https://www.omg.org/spec/Commons/PartiesAndSituations/
cmns-txt	https://www.omg.org/spec/Commons/TextDatatype/
deo	http://purl.org/spar/deo
doco	http://purl.org/spar/doco
po	http://www.essepuntato.it/2008/12/pattern#

8.2 Namespace Definitions for the Ontologies Included in this Specification

Table 8-2 provides the namespace declarations required for use of the ontologies included in this specification. The prefixes provided in Table 8-2 are normative, and their use is required in any conformant application or extension.

The namespace approach taken for the SBRM ontologies included herein is based on OMG guidelines and is constructed as follows:

- The standard protocol, authority, and top-level specification part of any OMG specification namespace, which is https://www.omg.org/spec/
- The abbreviation for the specification: in this case ***SBRM***
- The ontology name

Note that the namespaces end "/" as opposed to the common alternative of "#" (referred to as a "slash" rather than "hash" approach), in order to accommodate server-side applications, since for web access text after a "#" would not be sent to the server.

Namespace prefixes are constructed as follows with the components separated by "-":

- The abbreviation used for prefix purposes across SBRM: ***sbrm***
- An abbreviation for the ontology name

The namespaces and prefixes for the individual ontologies are summarized in Table 8-2. Note that these are not versioned, although version IRIs are included in every OWL ontology and are documented in the metadata for each of them. See the metadata tables for the individual ontologies for that level of detail, below.

Table 8-2 Prefix and Namespaces for the SBRM Ontologies

Namespace Prefix	Namespace
sbrm-frst	https://www.omg.org/SBRM/FinancialReportElements/
sbrm-rev	https://www.omg.org/SBRM/ReportingEnvelope/
sbrm-rst	https://www.omg.org/SBRM/ReportStructure/

8.3 Reporting Envelope Ontology

The Reporting Envelope Ontology defines a "reporting envelope," a framework for organizing reports and their associated roles, such as submitters and recipients. This ontology corresponds to the Reporting Envelope metamodel detailed in Section 7.1 and extends its principles to provide a more formal representation using OWL (Web Ontology Language).

Designed to be high-level and flexible, this ontology supports various domain-specific reporting scenarios, including regulatory, business, and operational reports. It leverages foundational ontologies from the OMG Commons Ontology Library to ensure interoperability and reusability across diverse domains.

This ontology has been applied in multiple projects, such as the Financial Industry Business Ontology (FIBO), the Retail Industry Ontology (RIO), and the Pistoia Alliance's Identification of Medicinal Products Ontology (IDMP-O). These real-world applications demonstrate its versatility and effectiveness in addressing domain-specific reporting requirements.

Metadata for the Reporting Envelope ontology, which uses annotation properties from Dublin Core, SKOS, and the Commons Annotation Vocabulary, is given in Table 8-3, below.

Table 8-3 Reporting Envelope Ontology Metadata

Metadata Term	Value
OntologyIRI	https://www.omg.org/spec/SBRM/ReportingEnvelope/
rdfs:label	Reporting Envelope Ontology
dct:abstract	This ontology defines the general notion of a reporting envelope and related party concepts. It is designed to be very high level such that it can be extended to support any kind of domain-specific report, including but not limited to a business or regulatory report.
cmns-av:copyright	Copyright (c) 2018-2024 EDM Council, Inc.

cmns-av:copyright	Copyright (c) 2018-2024 Object Management Group, Inc.
cmns-av:copyright	Copyright (c) 2018-2024 Thematix Partners LLC
cmns-av:copyright	Copyright (c) 2019-2024 Federated Knowledge, LLC
cmns-av:copyright	Copyright (c) 2022-2024 Adaptive, Inc.
cmns-av:copyright	2022-2024 Auditchain Labs AG
dct:license	https://opensource.org/licenses/MIT
owl:versionIRI	https://www.omg.org/spec/SBRM/20241101/ReportingEnvelope/
owl:imports	https://www.omg.org/spec/Commons/AnnotationVocabulary/ https://www.omg.org/spec/Commons/ContextualDesignators/ https://www.omg.org/spec/Commons/DatesAndTimes/ https://www.omg.org/spec/Commons/Documents/ https://www.omg.org/spec/Commons/PartiesAndSituations/
skos:note	This ontology was originally designed for use in the Financial Industry Business Ontology (FIBO) for representing regulatory and other business reports, such as credit reports. It has since been extended based on usage in other projects, such as the emerging OMG Retail Industry Ontology (RIO) and Pistoia Alliance's Identification of Medicinal Products Ontology (IDMP-O) project.

Key Concepts

The Reporting Envelope Ontology introduces several critical concepts:

- ***Reporting Envelope***: A container that organizes a report and its associated parties, such as submitters and recipients.
- ***Report***: A document presenting information for a specific purpose, often within a defined period or context.
- **Related Parties**: Entities involved in the reporting process, including:
  - ***Submitter***: The party submitting the report.
  - ***Intended Recipient***: The party authorized to receive the report.
  - ***Reporting Party***: The entity responsible for creating the report.

These classes include properties to define relationships, such as submission date, report date, and the reporting period.

Use of External Ontologies

The ontology imports several foundational ontologies:

- **Commons Annotation Vocabulary**: For metadata annotations.
- **Commons Documents**: To describe reports as formal documents.
- **Commons Parties and Situations**: To define roles like submitter and recipient.
- **Commons Dates and Times**: Standardize temporal properties such as report and submission dates.

The Reporting Envelope Ontology provides a structured, extensible framework for modeling reports and their associated parties. Its integration with the OMG Commons Ontology Library ensures compatibility with other ontologies, supporting diverse reporting use cases across domains. By formalizing reporting concepts, this ontology enables consistent, reusable, and interoperable data models for regulatory, business, and operational contexts.

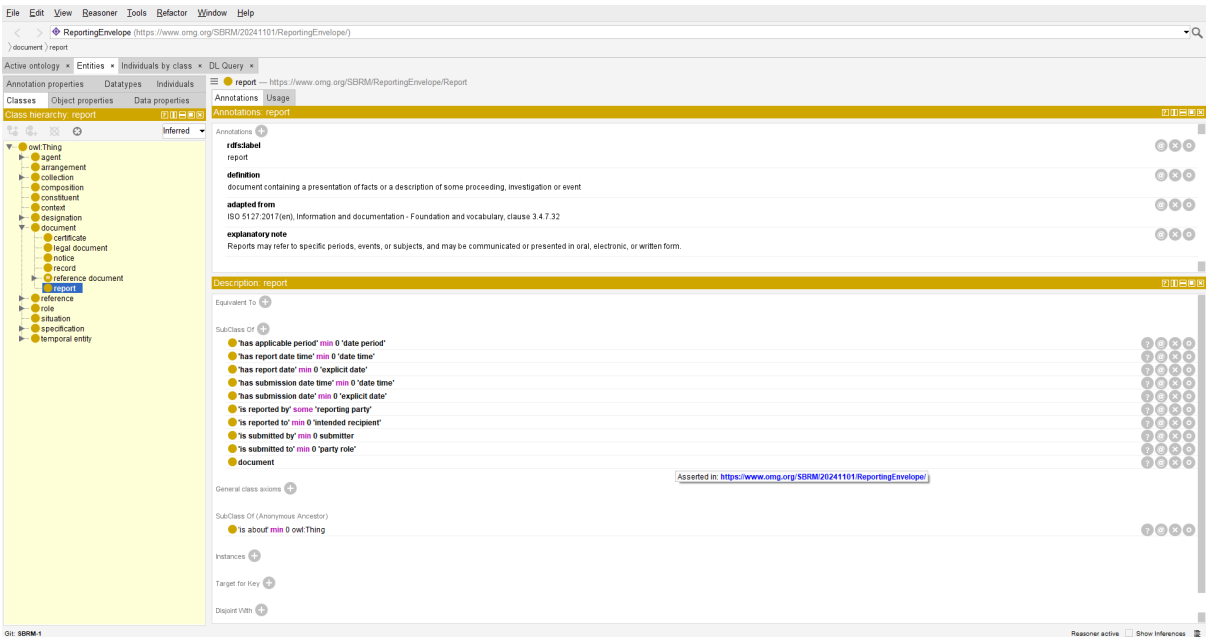


Figure 8-1 Overview of the Report Class in Protégé

An overview ***Reporting Envelope*** ontology, showing detail of the main ***Report*** class, is given in Figure 8-1, above, which is a screenshot from the Protégé tool from Stanford University[1]. The detailed annotations and axioms that comprise the ***Reporting Envelope*** ontology are provided in Table 8-4, below.

Table 8-4 Reporting Envelope Ontology Details

Classes

Name	Annotations	Class Expressions
IntendedRecipient (intended recipient)	Definition: party that is responsible for and authorized to receive something, such as a message, package, or business or regulatory report Adapted from: ISO 11568:2023(en) Financial services — Key management (retail), clause 3.73	Parent Class: cmns-pts:PartyRole
Report (report)	Definition: document containing a presentation of facts or a description of some proceeding, investigation or event Explanatory note: Reports may refer to specific periods, events, or subjects, and may be communicated or presented in oral, electronic, or written form. Adapted from: ISO 5127:2017(en), Information and documentation - Foundation and vocabulary, clause 3.4.7.32	Parent Class: cmns-doc:Document Property Restriction: ≥ 0 isReportedTo.IntendedRecipient Property Restriction: ≥ 0 isSubmittedBy.Submitter Property Restriction: ≥ 0 cmns-cxtdsg:hasApplicablePeriod.cmns-dt:DatePeriod Property Restriction: ≥ 0 hasReportDateTime.cmns-dt:DateTime Property Restriction: ≥ 0 hasSubmissionDateTime.cmns-dt:DateTime Property Restriction: ≥ 0 hasReportDate.cmns-dt:ExplicitDate Property Restriction: ≥ 0 hasSubmissionDate.cmns-dt:ExplicitDate Property Restriction: ≥ 0 isSubmittedTo.cmns-pts:PartyRole Property Restriction: ∃ isReportedBy.ReportingParty
ReportingParty (reporting party)	Definition: party that develops and provides a report, often in response to some contractual, legal, business, or regulatory requirement	Parent Class: cmns-pts:PartyRole Property Restriction: ∃ cmns-doc:specifies.Report
Submitter (submitter)	Definition: party delivering something, such as a regulatory report Explanatory note: Note that in many cases, but not all, the reporting party and submitter are the same party.	Parent Class: cmns-pts:PartyRole Property Restriction: ≥ 0 submits.Report

Properties

Name	Annotations	Property Axioms
hasReportDate (has report date)	Definition: date on which a report was issued	Parent Property: cmns-dt:hasDateOfIssuance, cmns-dt:hasExplicitDate Range: cmns-dt:ExplicitDate
hasReportDateTime (has report date time)	Definition: date and time at which a report was issued	Parent Property: cmns-dt;hasDateTime Range: cmns-dt:DateTime
hasSubmissionDate (has submission date)	Definition: date on which a report was provided to its intended recipient	Parent Property: cmns-dt:hasDateOfIssuance, cmns-dt:hasExplicitDate Range: cmns-dt:ExplicitDate
hasSubmissionDateTime (has submission date time)	Definition: date and time at which a report was provided to its intended recipient	Parent Property: cmns-dt;hasDateTime Range: cmns-dt:DateTime
isReportedBy (is reported by)	Definition: indicates the party that reports something	Parent Property: cmns-pts:hasPartyRole Range: ReportingParty
isReportedTo (is reported to)	Definition: indicates the party for which a report has been prepared and to whom the report should be delivered	Parent Property: cmns-pts:hasPartyRole Range: IntendedRecipient
isSubmittedBy (is submitted by)	Definition: indicates the party that submits something	Parent Property: cmns-pts:hasPartyRole Range: Submitter Inverse: submits
isSubmittedTo (is submitted to)	Definition: indicates the party to which something is submitted	Parent Property: cmns-pts:hasPartyRole Range: PartyRole
reportsOn (reports on)	Definition: indicates a subject matter, observation(s), assessment(s), focus or other topic of a report	Parent Property: cmns-doc:isAbout Domain: Report

submits (submits)	Definition: presents something (a proposal, application, report, or other document) for consideration or review	Domain: Submitter

[1] https://protege.stanford.edu/

8.4 Report Structure Ontology

This ontology defines concepts used for specifying the structure of a report. It is designed to be very high level such that it can be extended to support any kind of domain-specific report, including but not limited to a business or regulatory report. It reuses ontologies defined in the OMG's Commons Ontology Library.

Metadata for the Report Structure ontology, which uses annotation properties from Dublin Core, SKOS, and the Commons Annotation Vocabulary, is given in Table 8-5, below.

Table 8-5 Report Structure Ontology Metadata

Metadata Term	Value
OntologyIRI	https://www.omg.org/spec/SBRM/ReportStructure/
rdfs:label	Report Structure Ontology
dct:abstract	This ontology defines concepts used for specifying the structure of a report. It is designed to be very high level such that it can be extended to support any kind of domain-specific report, including but not limited to a business or regulatory report.
cmns-av:copyright	Copyright (c) 2018-2024 EDM Council, Inc.
cmns-av:copyright	Copyright (c) 2018-2024 Object Management Group, Inc.
cmns-av:copyright	Copyright (c) 2018-2024 Thematix Partners LLC
cmns-av:copyright	Copyright (c) 2019-2024 Federated Knowledge, LLC
cmns-av:copyright	Copyright (c) 2022-2024 Adaptive, Inc.
cmns-av:copyright	2022-2024 Auditchain Labs AG
dct:license	https://opensource.org/licenses/MIT
owl:versionIRI	https://www.omg.org/spec/SBRM/20241101/ReportStructure/
owl:imports	https://www.omg.org/spec/Commons/AnnotationVocabulary/ https://www.omg.org/spec/Commons/ContextualDesignators/ https://www.omg.org/spec/Commons/DatesAndTimes/ https://www.omg.org/spec/Commons/Documents/ https://www.omg.org/spec/Commons/PartiesAndSituations/

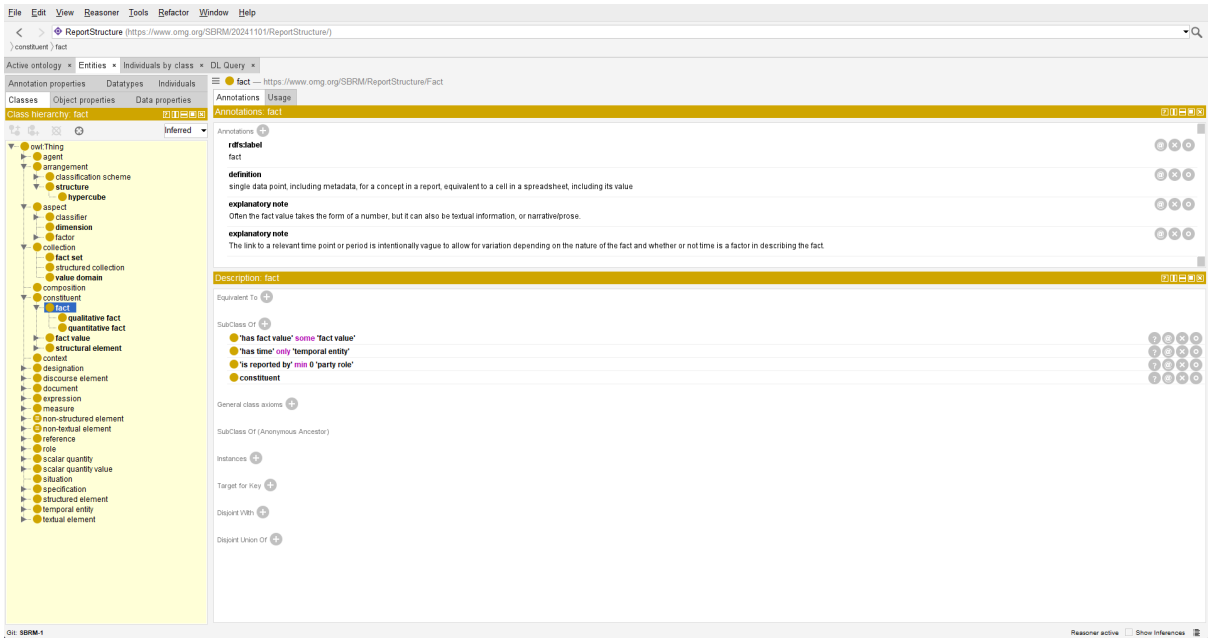


Figure 8-2 Overview of the Report Structure and Fact Class in Protégé

An overview the ***Report Structure*** ontology, showing detail of the main ***Fact*** class is given in Figure 8-2, above.

8.5 Financial Report Elements Ontology

This ontology defines concepts used for specifying financial components of a report. It is designed to be very high level such that it can be extended to support any kind of domain-specific report, including but not limited to a business or regulatory report. It reuses ontologies defined in the OMG's Commons Ontology Library.

Metadata for the Financial Report Elements ontology, which uses annotation properties from Dublin Core, SKOS, and the Commons Annotation Vocabulary, is given in Table 8-5, below.

Table 8-6 Financial Report Elements Ontology Metadata

Metadata Term	Value
OntologyIRI	https://www.omg.org/SBRM/FinancialReportElements/
rdfs:label	Financial Report Elements Ontology
dct:abstract	This ontology defines concepts used for specifying financial components of a report. It is designed to be very high level such that it can be extended to support any kind of domain-specific report, including but not limited to a business or regulatory report.
cmns-av:copyright	Copyright (c) 2018-2024 EDM Council, Inc.

cmns-av:copyright	Copyright (c) 2018-2024 Object Management Group, Inc.
cmns-av:copyright	Copyright (c) 2018-2024 Thematix Partners LLC
cmns-av:copyright	Copyright (c) 2019-2024 Federated Knowledge, LLC
cmns-av:copyright	Copyright (c) 2022-2024 Adaptive, Inc.
cmns-av:copyright	2022-2024 Auditchain Labs AG
dct:license	https://opensource.org/licenses/MIT
owl:versionIRI	https://www.omg.org/SBRM/20241101/FinancialReportElements/
owl:imports	https://www.omg.org/spec/Commons/AnnotationVocabulary/ https://www.omg.org/spec/Commons/ContextualDesignators/ https://www.omg.org/spec/Commons/DatesAndTimes/ https://www.omg.org/spec/Commons/Documents/ https://www.omg.org/spec/Commons/PartiesAndSituations/

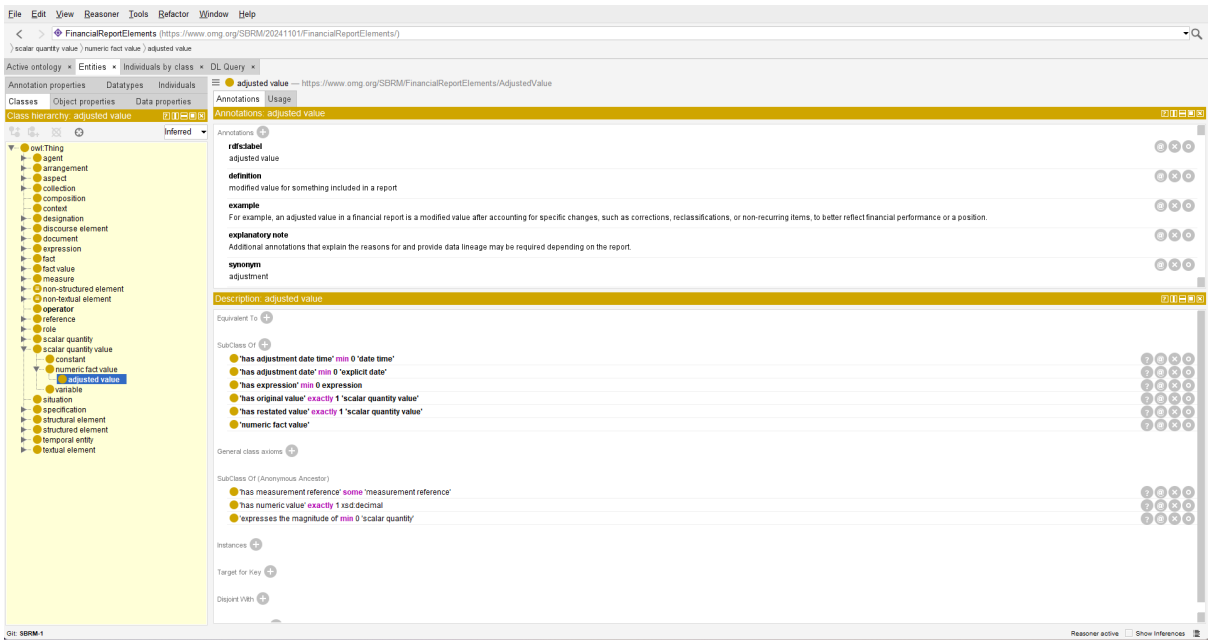


Figure 8-3 Overview of the Financial Report Elements Hierarchy and Adjusted Value Class in Protégé

An overview of the ***Financial Report Elements*** ontology, showing detail of the ***adjusted value*** class, is given in Figure 8-3, above.

Annex A. Not Used

The name is retained to avoid the need for renumbering other sections.

Annex B. Document Components Ontology (Normative)

This ontology is to be added to OMG's Commons Ontology Library specification.

The following row should be added, in alphabetic sequence, to Table 7-2:

`cmns-docc https://www.omg.org/spec/Commons/DocumentComponents/`

The following should be added as a new final section 8.x

8.x Ontology: Document Components

The document components ontology extends the SPAR document component ontologies for defining table elements and diagrams. It is intended to be used for describing deep links into documents by describing their structure without the need of explicit markup.

Metadata for the Reporting ontology, which uses annotation properties from Dublin Core, SKOS, and the Commons Annotation Vocabulary, is given in Table Annex B. Document Components Ontology (Normative)-1 Document Components Ontology Metadata, below.

Table Annex B. Document Components Ontology (Normative)-1 Document Components Ontology Metadata

Metadata Term	Value
OntologyIRI	https://www.omg.org/spec/Commons/DocumentComponents/
rdfs:label	Document Components Ontology
dct:abstract	This ontology provides extensions to the SPAR document component ontologies for defining table elements and diagrams. It is intended to be used for describing deep links into documents by describing their structure without the need of explicit markup.
dct:contributor	Elisa Kendall, Thematix Partners LLC
dct:contributor	Thomas Weber, OSTHUS, a PHARMALEX Company
cmns-av:copyright	Copyright (c) 2024 EDM Council, Inc.
cmns-av:copyright	Copyright (c) 2024 Pistoia Alliance, Inc.
dct:license	https://opensource.org/licenses/MIT
owl:versionIRI	https://www.omg.org/spec/Commons/20240501/DocumentComponents/
owl:imports	https://www.omg.org/spec/Commons/AnnotationVocabulary/ https://www.omg.org/spec/Commons/Collections/ https://www.omg.org/spec/Commons/Documents/

https://www.omg.org/spec/Commons/TextDatatype/

http://purl.org/spar/deo

http://purl.org/spar/doco

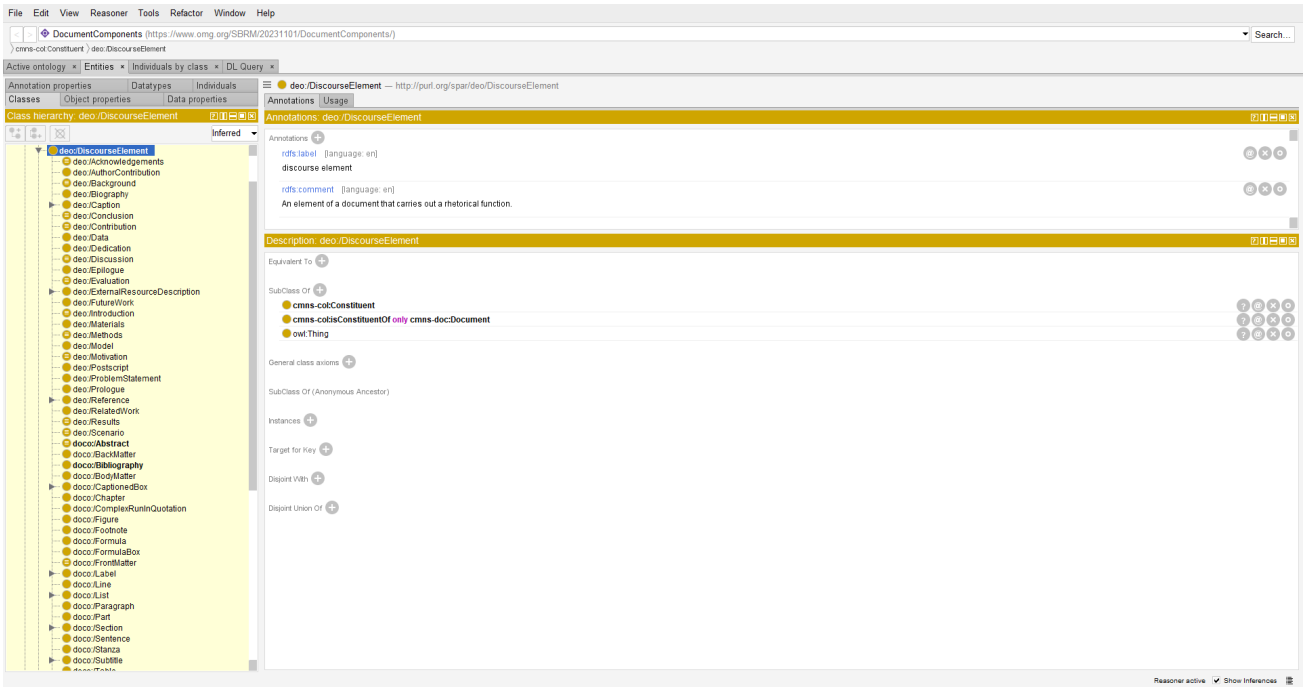


Figure Annex B. Document Components Ontology (Normative)-18 High-Level Overview of the Discourse Elements Supported in SPAR Ontologies

An overview showing the rich class hierarchy for discourse elements supported in the SPAR ontologies is provided in **Figure Annex B. Document Components Ontology (Normative)-18**, above. **Figure Annex B. Document Components Ontology (Normative)-19** depicts the extension included in the Document Components ontology for better support of table columns, and **Figure Annex B. Document Components Ontology (Normative)-20** provides a similar view of table row extensions.

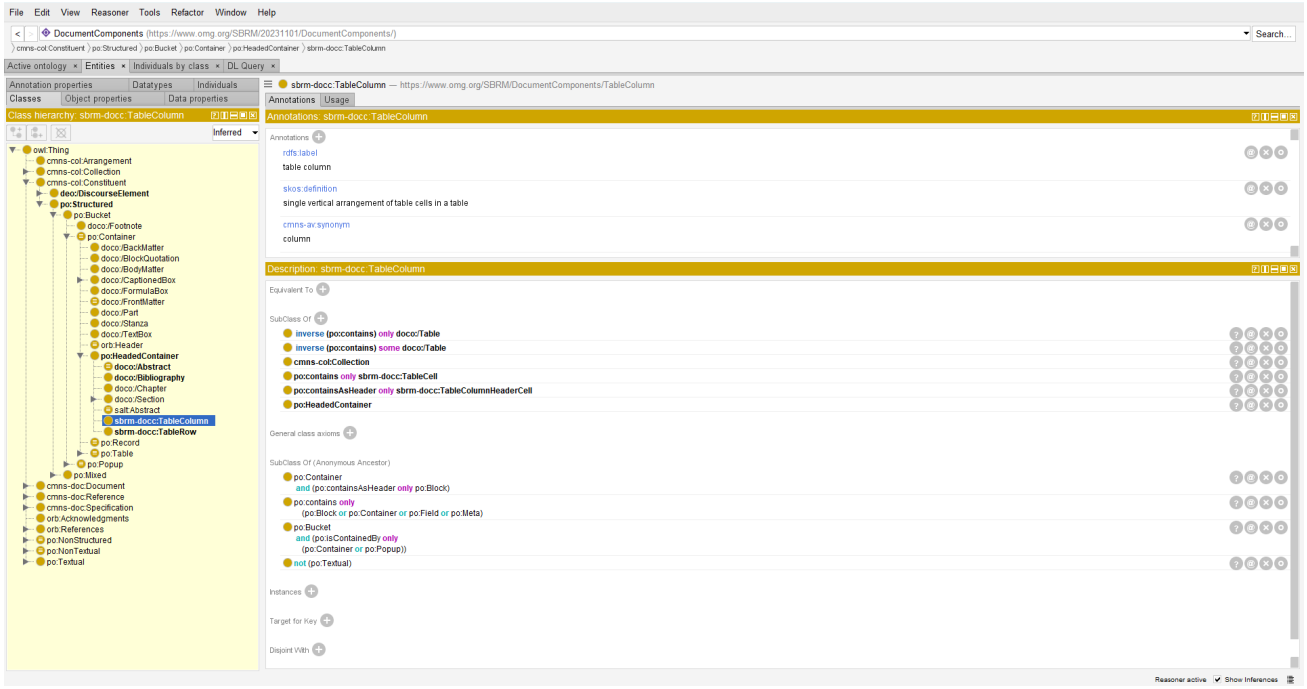


Figure Annex B. Document Components Ontology (Normative)-19 Document Component Extensions for Table Column Details

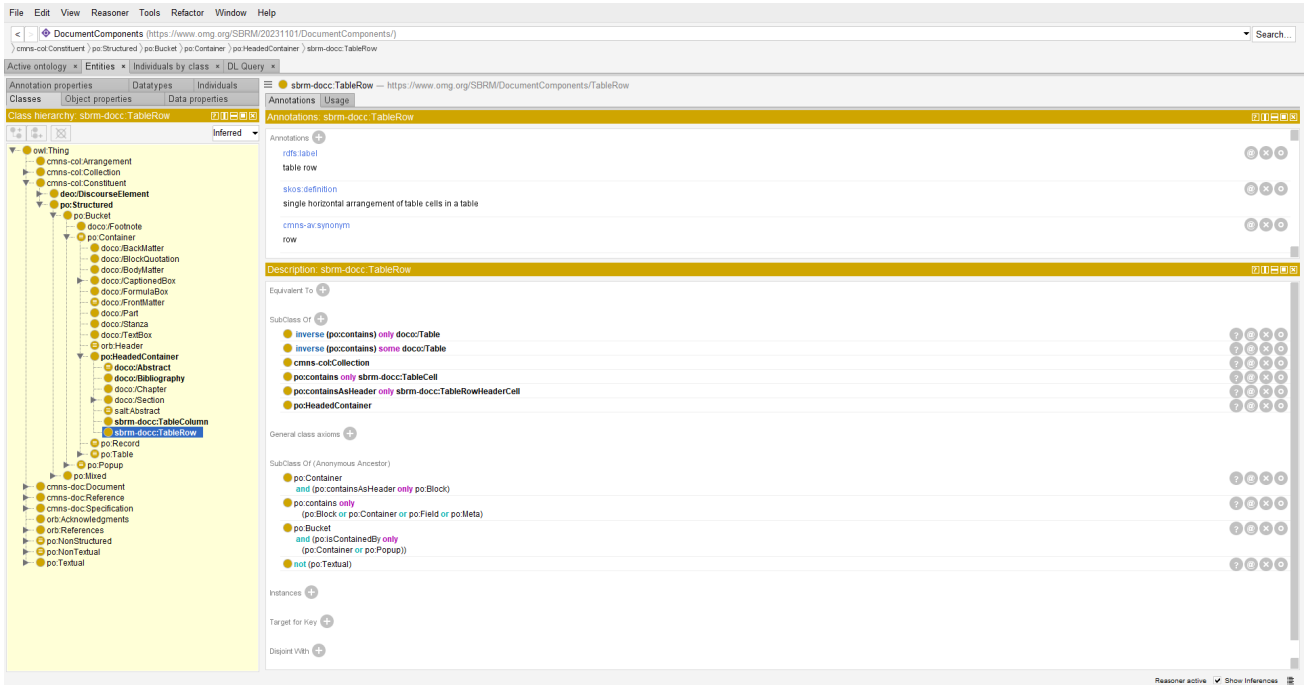


Figure Annex B. Document Components Ontology (Normative)-20 Document Component Extensions for Table Row Details

The detailed annotations and axioms that comprise the Document Components ontology, extending the SPAR ontologies, are provided below.

Table Annex B. Document Components Ontology (Normative)-2 Document Components Ontology Details

Classes

Name	Annotations	Class Expressions
deo:DiscourseElement		Parent Class: cmns-col:Constituent Property Restriction: ∀ cmns-col;isConstituentOf: cmns-doc:Document
doco:Abstract		Parent Class: deo:DiscourseElement
doco:Bibliography		Parent Class: deo:DiscourseElement
doco:ListOfFigures		Parent Class: cmns-col:Collection
doco:ListOfTables		Parent Class: cmns-col:Collection
doco:TableOfContents		Parent Class: cmns-col:Collection
po:Block		Parent Class: cmns-col:Constituent
po:HeadedContainer		Parent Class: cmns-col:Constituent
po:Structured		Parent Class: cmns-col:Constituent
FigureCaption (figure caption)	Definition: title or brief explanation appended to a figure	Parent Class: deo:Caption, po:Block Property Restriction: {owl:inverseOf hasCaption} ∀ doco:Figure Property Restriction: {owl:inverseOf hasCaption} ∃ doco:Figure
TableBodyCell (table body cell)	Definition: table cell that is not a row or column header Synonym: body cell Synonym: inner cell	Parent Class: TableCell Class Axiom: ⌐ TableColumnHeaderCell Class Axiom: ⌐ TableRowHeaderCell
TableCaption (table caption)	Definition: title or brief explanation on a table	Parent Class: deo:Caption, po:Block Property Restriction: {owl:inverseOf hasCaption} ∀ doco:Table Property Restriction: {owl:inverseOf hasCaption} ∃ doco:Table

TableCell (table cell)	Definition: element in a table that is arranged in a tabular layout, and that is part of a table row, a table column or both Note: This is similar to the HTML td element.	Parent Class: po:Block Property Restriction: {owl:inverseOf po:contains} ∃ doco:Table
TableColumn (table column)	Definition: single vertical arrangement of table cells in a table Synonym: column	Parent Class: po:HeadedContainer, cmns-col:Collection Property Restriction: {owl:inverseOf po:contains} ∀ doco:Table Property Restriction: {owl:inverseOf po:contains} ∃ doco:Table Property Restriction: ∀ po:contains.TableCell Property Restriction: ∀ po:containsAsHeader.TableColumnHeaderCell
TableColumnHeaderCell (table column header cell)	Definition: table cell that is a header for the table column Synonym: column header	Parent Class: TableCell Class Axiom: ⌐ TableRowHeaderCell
TableRow (table row)	Definition: single horizontal arrangement of table cells in a table Synonym: row	Parent Class: po:HeadedContainer, cmns-col:Collection Property Restriction: {owl:inverseOf po:contains} ∀ doco:Table Property Restriction: {owl:inverseOf po:contains} ∃ doco:Table Property Restriction: ∀ po:contains.TableCell Property Restriction: ∀ po:containsAsHeader.TableRowHeaderCell
TableRowHeaderCell (table row header cell)	Definition: table cell that is a header for the table row Synonym: row header	Parent Class: TableCell

Properties

Name	Annotations	Property Axioms
po:contains	Definition: relates a document element to its sontents	Parent Property: cmns-col:comprises
hasCaption (has caption)	Definition: relates a document element to a caption on it	Parent Property: cmns-doc;refersTo Range: deo:Caption

hasExactText (has exact text)	Definition: relates a textual element to its precise literal text	Parent Property: cmns-txt:hasTextValue Domain: po:Textual
hasTableColumnIndex (has table column index)	Definition: horizontal position of a table body cell or table column header cell Note: Column indexes start with 1 and are ordered left to right, with header cells first. Both header and body cells are counted with respect to the index, so if there is a single row header column then the leftmost body cell has a column index of 2.	Domain: {TableBodyCell ∩ TableColumnHeaderCell ∩ TableColumn} Range: xsd:positiveInteger
hasTableColumnSpan (has table column span)	Definition: number of columns the table cell should span Note: The column span is counted beginning with and is including the left most column. Note: This is similar to the HTML colspan attribute.	Domain: {TableBodyCell ∩ TableRowHeaderCell} Range: xsd:positiveInteger
hasTableRowIndex (has table row index)	Definition: horizontal position of a table body cell or table column header cell Note: Row indexes start with 1 and are ordered top to bottom with header cells first. Both header and body cells are counted with respect to the index, so if there is a single header row, then the topmost body cells has a row index of 2.	Domain: {TableBodyCell ∩ TableRowHeaderCell ∩ TableRow} Range: xsd:positiveInteger
hasTableRowSpan (has table row span)	Definition: number of rows the table cell should span Note: The row span is counted beginning with and is including the top most row. Note: This is similar to the HTML rowspan attribute.	Domain: {TableBodyCell ∩ TableRowHeaderCell} Range: xsd:positiveInteger
hasTextualContent (has textual content)	Definition: relates a document element to its textual content Usage note: If the element contains mixed content, then the text may be different from the literal text value.	Parent Property: cmns-txt:hasTextValue

Annex C Standard Business Report Model Narrative Introduction (Informative)

This clause contains a narrative building up the structure of a (financial) report from the most primitive facts to more sophisticated structures. It is illustrated with examples in both traditional XBRL structures (with footnotes linking to live web-based examples), and the SBRM equivalent, showing both the relevant parts of the metamodel (as diagrams) and the equivalent data (as RDF corresponding to the equivalent SBRM ontology)

C.1 Report Basics

A scalar is a fact which has no characteristics; it stands on its own. For example, the value of pi is a scalar, the value of pi never changes; it always has the same value for everyone. (Pi or π is the ratio of a circle's circumference to its diameter and always has the value of equal to 3.14)

Fact Value
3.14

A business **report**[1] communicates information in the form of numbers and words. A **fact**[2] defines a single, observable, reportable piece of information contained within a business report, or **fact value**[3], contextualized for unambiguous interpretation or analysis by one or more distinguishing aspects (a.k.a. characteristics). For example, below are two facts with the values of "2,000" and "1,000". However, the two facts above are not contextualized.

Fact Value
2,000
1,000

An **aspect**[4] describes a fact. An aspect provides information necessary to describe a fact or distinguish one fact from another fact within a report. For example, below you see the concept aspect of the numbers "2,000" and "1,000" which relates to the concepts "Revenues" and "Net income" respectively:

Concept Aspect	Fact Value
Revenues	2,000
Net income	1,000

To fully describe a fact, you need more than just one aspect. A fact must always provide three **core aspects**[5]: reporting entity that reported the fact, calendar period of the reported fact, and the concept that describes the reported fact. Below you see two facts which are

---

1 Member Nonaggregating, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/MemberNonaggregation.html
2 Variance, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Variance.html
3 Member Arrangement Pattern, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/MemberArrangementPattern.html
4 Line Items, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/LineItems.html
5 Abstract, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Abstract.html

characterized by three core aspects which are used to differentiate the two facts from one another.

Reporting Entity Aspect	Calendar Period Aspect	Concept Aspect	Fact Value
ABC Company	Jan 1, 2019 to Dec 31, 2019	Revenues	2,000
ABC Company	Jan 1, 2019 to Dec 31, 2019	Net income	1,000

In addition to the core aspects that you always must use, creators of reports can also provide additional **noncore aspects**[6]. A noncore aspect is simply some additional aspect that is created to further distinguish facts beyond the capabilities of the three core aspects. Below you see the noncore aspect "Legal Entity Aspect" has been added to the two facts we have been working with:

Reporting Entity Aspect	Legal Entity Aspect	Calendar Period Aspect	Concept Aspect	Fact Value
ABC Company	Consolidated entity	Jan 1, 2019 to Dec 31, 2019	Revenues	2,000
ABC Company	Consolidated entity	Jan 1, 2019 to Dec 31, 2019	Net income	1,000

Fact values can be **numeric**[7] or **nonnumeric**[8]. Numeric fact values require additional information to describe the units of the numeric fact and the rounding that is used to report the numeric fact. **Units**[9] and **rounding**[10] are properties of the fact value that provide information necessary to describe numeric fact values. Below you see that the units of "US Dollars" and that the rounding of the fact value is "Thousands of dollars":

Reporting Entity Aspect	Legal Entity Aspect	Calendar Period Aspect	Concept Aspect	Fact Value	Units	Rounding
ABC Company	Consolidated entity	Jan 1, 2019 to Dec 31, 2019	Revenues	2,000	US Dollars	Thousands of dollars
ABC Company	Consolidated entity	Jan 1, 2019 to Dec 31, 2019	Net income	1,000	US Dollars	Thousands of dollars

To summarize where we are thus far and to be crystal clear; below you see one fact. That single fact is characterized by a set of four aspects. The numeric fact value is described as having units of "US Dollars" and that the fact value is rounded to the nearest "Thousands of dollars".

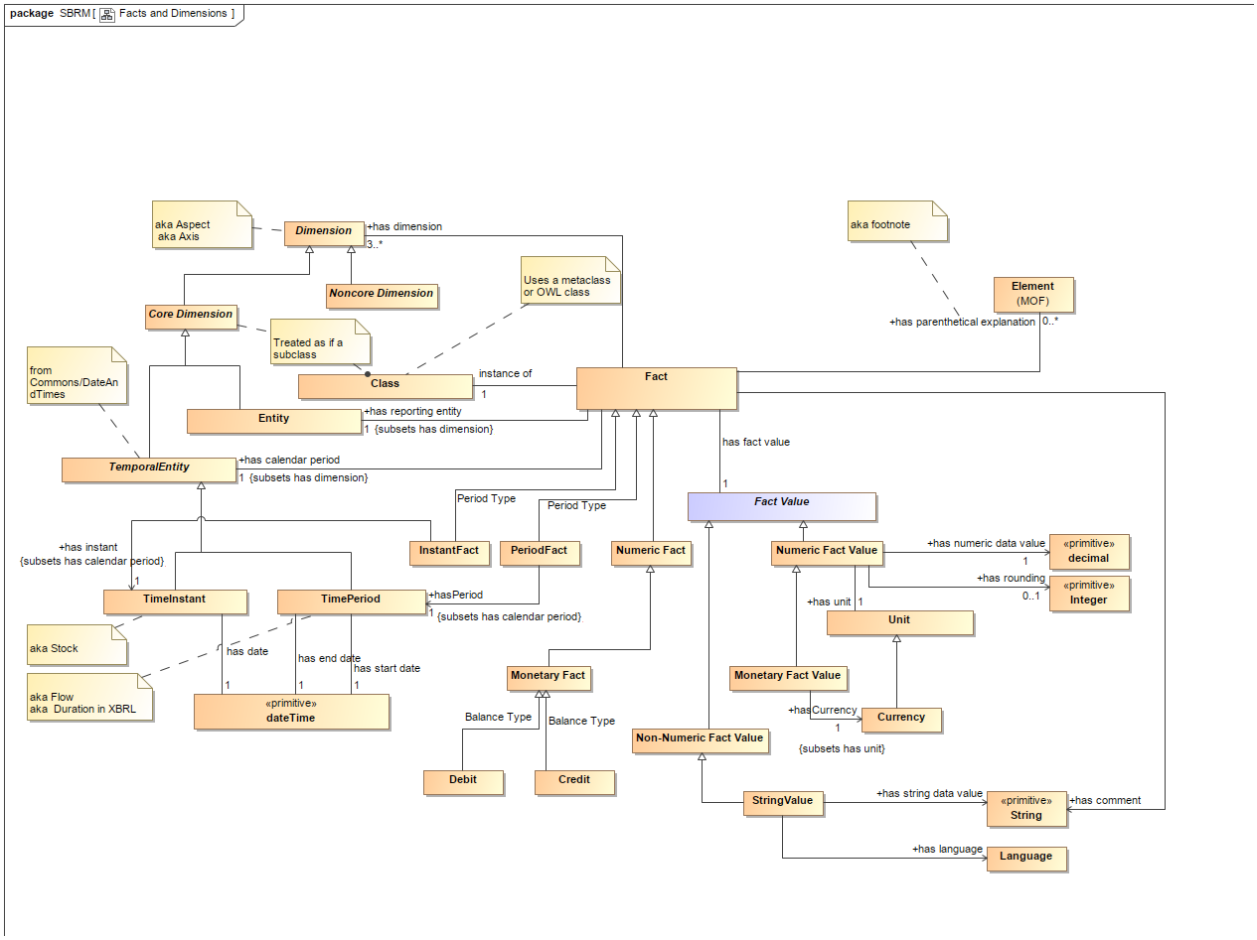
Reporting Entity Aspect	Legal Entity Aspect	Calendar Period Aspect	Concept Aspect	Fact Value	Units	Rounding
ABC Company	Consolidated entity	Jan 1, 2019 to Dec 31, 2019	Revenues	2,000	US Dollars	Thousands of dollars

6 Mechanical Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/MechanicalRule.html
7 Table, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Table.html
8 Complex Computation, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ComplexComputation.html
9 *Comparison of Renderings for Concept Arrangement Patterns*, http://accounting.auditchain.finance/sbrm/2023/Prototype/conformance-suite/Production/ComparisonOfConceptArrangementPatternRenderings.pdf
10 Roll Up, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/RollUp.html

A **fact set**[11] is a set of facts which go together (tend to be cohesive and share a certain common nature) for some specific purpose within a business report. Synonyms for fact set are block and fact table. Below you see three facts that make up a fact set that are used to describe the breakdown of revenues by geographic area.

Reporting Entity Aspect	Legal Entity Aspect	Geographic Area Aspect	Calendar Period Aspect	Concept Aspect	Fact Value	Units	Rounding
ABC Company	Consolidated entity	All Geographic Areas Combined	Jan 1, 2019 to Dec 31, 2019	Revenues	2,000	US Dollars	Thousands of dollars
ABC Company	Consolidated entity	North America	Jan 1, 2019 to Dec 31, 2019	Revenues	1,000	US Dollars	Thousands of dollars
ABC Company	Consolidated entity	South America	Jan 1, 2019 to Dec 31, 2019	Revenues	1,000	US Dollars	Thousands of dollars

Here is the metamodel representing the material covered so far:



And here is the RDF representing the data above, using [Turtle] syntax.

[11] Disclosure, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Disclosure.html

First the definitions for this report type, providing extensions to standard SBRM for specific fact types and noncore aspects.

```
pattern: a owl:Ontology ;
  owl:imports sbrm:, lcc: .
pattern:Revenues a owl:Class ;
  rdfs:subClassOf sbrm:MonetaryFact, sbrm:Credit, sbrm:PeriodFact .
pattern:hasGeographicArea a owl:ObjectProperty ;
  rdfs:subPropertyOf sbrm:hasAspect ;
  rdfs:range lcc:GeographicRegion .
pattern:hasLegalEntity a owl:ObjectProperty ;
  rdfs:subPropertyOf sbrm:hasAspect .
```

Now the data:

```
sample:ABC a sbrm:Entity; rdfs:label "ABC Company".
sample:FY-2019 a sbrm:Period;
  sbrm:hasStartDate "2019-01-01T00:00:00Z"^^xs:dateTime ;
  sbrm:hasEndDate "2019-12-31T00:00:00Z"^^xs:dateTime ;
.
sample:RevenuesAll a pattern:Revenues ;
  sbrm:hasReportingEntity sample:ABC ;
  sbrm:hasPeriod sample:FY-2019 ;
  pattern:hasGeographicArea sample:All ;
  pattern:hasLegalEntity pattern:ConsolidatedEntity ;
  sbrm:hasFactValue sample:RevenuesAllV.
sample:RevenuesAllV a sbrm:MonetaryFactValue ;
  sbrm:hasNumericDataValue "2000"^^xs:decimal ;
  sbrm:hasUnit sbrm:USD ;
  sbrm:hasRounding -3 ;
.
sample:RevenuesNA a pattern:Revenues ;
  sbrm:hasReportingEntity sample:ABC ;
  sbrm:hasPeriod sample:FY-2019 ;
  pattern:hasGeographicArea LCC:NorthAmerica ;
  pattern:hasLegalEntity pattern:ConsolidatedEntity ;
  sbrm:hasFactValue sample:RevenuesNAV.
sample:RevenuesNAV a sbrm:MonetaryFactValue ;
  sbrm:hasNumericDataValue "1000"^^xs:decimal ;
  sbrm:hasUnit sbrm:USD ;
  sbrm:hasRounding -3 ;
.
sample:RevenuesSA a pattern:Revenues ;
  sbrm:hasReportingEntity sample:ABC ;
  sbrm:hasPeriod sample:FY-2019 ;
  pattern:hasGeographicArea LCC:SouthAmerica ;
  pattern:hasLegalEntity pattern:ConsolidatedEntity ;
  sbrm:hasFactValue sample:RevenuesSAV.
sample:RevenuesSAV a sbrm:MonetaryFactValue ;
  sbrm:hasNumericDataValue "1000"^^xs:decimal ;
  sbrm:hasUnit sbrm:USD ;
  sbrm:hasRounding -3 ;
.
```

Rules[12] guide, control, suggest, or influence behavior. Rules cause things to happen, prevent things from happening, or suggest that it might be a good idea if something did or did not happen. Rules help shape judgment, help make decisions, help evaluate, help shape behavior. Don't make the mistake of thinking that rules are completely inflexible and that you cannot break rules. Sure, maybe there are some rules that can never be broken. Maybe there are some rules that you can break. It helps to think of breaking rules as penalties in a football game. The point is that the guidance, control, suggestions, and influence offered by rules are a choice of business professionals. The meaning of a rule is separate from the level of enforcement someone might apply to the rule.
So, considering our fact set below we know that the value "2,000" is for the concept "Revenues", for the period "Jan 1, 2019 to Dec 31, 2019", relates to the legal entity "Consolidated entity", of the reporting entity "ABC Company" and is the total of all "Geographic Areas". "North America" and "South America" are part of the *whole* "All Geographic Areas Combined". A rule that expresses that relationship might be expressed as:

"All Geographic Areas Combined = North America + South America".

Rules both describe and can be used to verify that reported facts are consistent with the provided description. There are many different types of rules including mathematical, structural, mechanical, logical, and accounting related rules.

For SBRM, the [SHACL] language is used to express the rules.
In this example it would be:

```
pattern:RevenueShape
 a sh:NodeShape ;
 sh:targetClass pattern:Revenues ;
 sh:sparql [
    a sh:SPARQLConstraint ;
    sh:message "Revenue for All must be the total of all areas" ;
    sh:select """
      SELECT DISTINCT ?this ?total
      WHERE {
       {SELECT(SUM(?in) as ?total)
          WHERE {
            ?region pattern:hasGeographicArea ?a; ?a a
LCC:GeographicRegion;
            ?this ?region/sbrm:hasFactValue/sbrm:hasNumericDataValue ?in}
 GROUP BY ?this }
        ?this sbrm:hasOutput ?out.
       FILTER (?total != ?out)
      }
      """ ;
 ] .
```

[12] Disclosure Mechanics Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/DisclosureMechanicsRule.html

C.2 Report Rendering

An **information model definition**[13] is a structure which is created to represent each fragment of a report. The following pieces, or **report elements**[14], are commonly used to construct the information model description: **Network**[15], **Table**[16], **Axis**[17], **Member**[18], **Line Items**[19], **Abstract**[20], and **Concept**[21]. Table is a synonym for hypercube. Axis is a synonym for dimension. Line items is a synonym for primary items.
Below you see the information model description of the structure of a fragment of a report, in this case one fact set which is used to describe the components of inventory:

#	Label	Report Element Class	Period Type	Balance	Name
1	**Inventory, by Compoment [Table]**	[Table]			gaap:InventoryByCompomentTable
2	**Legal Entity [Axis]**	[Axis]			frm:LegalEntityAxis
3	Consolidated Entity [Member]	[Member]			frm:ConsolidatedEntityMember
4	**Inventory, by Component [Line Items]**	[Line Items]			gaap:InventoryByComponentLineItems
5	Inventory, by Component [Roll Up]	[Abstract]			gaap:InventoryByComponentRollUp
6	Finished Goods	[Concept] Monetary	As Of	Debit	gaap:FinishedGoods
7	Work in Progress	[Concept] Monetary	As Of	Debit	gaap:WorkInProgress
8	Raw Material	[Concept] Monetary	As Of	Debit	gaap:RawMaterial
9	Inventory	[Concept] Monetary	As Of	Debit	gaap:Inventory

Something is important to point out. We mentioned that in XBRL you have core aspects and noncore aspects. In the typical software applications created today, the core aspects reporting entity and calendar period are commonly not represented in the information model description that is typically created by software applications. The graphic above shows that sort of representation.
Below you see a truer information model description which includes the reporting entity and the calendar period. Also, per the US GAAP XBRL Taxonomy, the IFRS XBRL Taxonomy the term "[Axis]" is used as a synonym of "Aspect". Axis and aspect are synonyms and mean exactly the same thing. Also "Period" and "Calendar Period" are exactly the same thing.

---

13 Noncore Aspect, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/NoncoreAspect.html
14 Reporting Entity Aspect, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ReportingEntityAspect.html
15 Report, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Report.html
16 Template, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Template.html
17 Wikipedia, *Accounting Equation*, https://en.wikipedia.org/wiki/Accounting_equation
18 Network, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Network.html
19 Information Model Definition, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/InformationModelDefinition.html
20 Logical Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/LogicalRule.html
21 Consistency Crosscheck Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ConsistencyCrosscheckRule.html

#	Label	Report Element Class	Period Type	Balance	Name
1	**Inventory, by Compoment [Table]**	[Table]			gaap:InventoryByCompomentTable
2	**Reporting Entity [Axis]**	[Axis]			xbrl:ReportingEntityAxis
3	http://regulator.gov/id#1234567890	[Member]			http://regulator.gov/id#1234567890
4	**Period [Axis]**	[Axis]			xbrl:PeriodAxis
5	12/31/2010	[Member]			12/31/2010
6	12/31/2009	[Member]			12/31/2009
7	**Legal Entity [Axis]**	[Axis]			frm:LegalEntityAxis
8	Consolidated Entity [Member]	[Member]			frm:ConsolidatedEntityMember
9	**Inventory, by Component [Line Items]**	[Line Items]			gaap:InventoryByComponentLineItems
10	Inventory, by Component [Roll Up]	[Abstract]			gaap:InventoryByComponentRollUp
11	Finished Goods	[Concept] Monetary	As Of	Debit	gaap:FinishedGoods
12	Work in Progress	[Concept] Monetary	As Of	Debit	gaap:WorkInProgress
13	Raw Material	[Concept] Monetary	As Of	Debit	gaap:RawMaterial
14	Inventory	[Concept] Monetary	As Of	Debit	gaap:Inventory

Another part of the information model description is the mathematical rules that are used to describe and verify the roll up relations of the concepts that are a part of the information model description. Here are the roll up relations that are part of this information model description.

#	Label	Report Element Class	Weight	Balance	Name
1	Inventory	[Concept] Monetary			gaap:Inventory
2	Finished Goods	[Concept] Monetary	+1	Debit	gaap:FinishedGoods
3	Work in Progress	[Concept] Monetary	+1	Debit	gaap:WorkInProgress
4	Raw Material	[Concept] Monetary	+1	Debit	gaap:RawMaterial

Another part of the information model description is the facts within the fact set themselves. Here is the fact set or the **fact table**[22] for the facts that go with the information model description provided above.

#	Reporting Entity [Aspect]	Period [Aspect]	Concept [Aspect]	Legal Entity [Aspect]	Fact Value	Unit	Rounding
1	http://regulator.gov/id#1234567890	12/31/2010	Finished Goods	Consolidated Entity [Member]	600,000	USD	Thousands
2	http://regulator.gov/id#1234567890	12/31/2009	Finished Goods	Consolidated Entity [Member]	600,000	USD	Thousands
3	http://regulator.gov/id#1234567890	12/31/2010	Work in Progress	Consolidated Entity [Member]	300,000	USD	Thousands
4	http://regulator.gov/id#1234567890	12/31/2009	Work in Progress	Consolidated Entity [Member]	300,000	USD	Thousands
5	http://regulator.gov/id#1234567890	12/31/2010	Raw Material	Consolidated Entity [Member]	100,000	USD	Thousands
6	http://regulator.gov/id#1234567890	12/31/2009	Raw Material	Consolidated Entity [Member]	100,000	USD	Thousands
7	http://regulator.gov/id#1234567890	12/31/2010	Inventory	Consolidated Entity [Member]	1,000,000	USD	Thousands
8	http://regulator.gov/id#1234567890	12/31/2009	Inventory	Consolidated Entity [Member]	1,000,000	USD	Thousands

A software application takes the information model description structure, the information model description rules provided, the facts that are included within the fact set, and known best practices for rendering a business report that are coded into the software application in some manner and then generates a human-readable rendering of the reported information for a fragment or fact set of a report.

The following is the **rendering**[23] of the inventory components disclosure that we are working with above:

[22] Numeric Fact Value, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/NumericFactValue.html

[23] Fact, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Fact.html

Reporting Entity [Aspect]	http://regulator.gov/id#1234567890
Legal Entity [Aspect]	Consolidated Entity [Member]

Concept [Aspect]	Period [Aspect]	
	12/31/2020	12/31/2019
Inventory, by Component [Roll Up]		
Finished Goods	600,000	600,000
Work in Progress	300,000	300,000
Raw Material	100,000	100,000
Inventory	1,000,000	1,000,000

Different software applications may provide slightly different renderings using the same input information[24].

Component: (Network and Table)	
Network	JG - Schedule - **Inventory, by Component**
Table	Inventory, by Component [Table]

Reporting Entity [Axis]	1234567890 http://regulator.gov/id	
Legal Entity [Axis]	Consolidated Entity [Member]	
Unit [Axis]	USD	
	Period [Axis]	
Inventory, by Component [Line Items]	2010-12-31	2009-12-31
Inventory, by Component [Roll Up]		
Finished Goods	600	600
Work in Progress	300	300
Raw Material	100	100
Inventory	1,000	1,000

Here is what the information model description might look like in that software application:

Label	Report Element Class	Period	Balance	Preferred Label Role	Name
Inventory, by Component [Table]	[Table]			Standard Label	gaap:InventoryByComponentTable
Legal Entity [Axis]	[Axis]			Standard Label	frm:LegalEntityAxis
Consolidated Entity [Member]	[Member]			Standard Label	frm:ConsolidatedEntityMember
Inventory, by Component [Line Items]	[LineItems]			Standard Label	gaap:InventoryByComponentLineItems
Inventory, by Component [Roll Up]	[Abstract]			Standard Label	gaap:InventoryByComponentRollUp
Finished Goods	[Concept] Monetary	As Of	Debit	Standard Label	gaap:FinishedGoods
Work in Progress	[Concept] Monetary	As Of	Debit	Standard Label	gaap:WorkInProgress
Raw Material	[Concept] Monetary	As Of	Debit	Standard Label	gaap:RawMaterial
Inventory	[Concept] Monetary	As Of	Debit	Standard Label	gaap:Inventory

Here is what the roll up rule relations representation might look like in that software application:

Label	Report Element Class	Balance	Weight	Name
Inventory, by Component [Table]	[Table]		0	gaap:InventoryByComponentTable
Inventory	[Concept] Monetary	Debit	0	gaap:Inventory
Finished Goods	[Concept] Monetary	Debit	1	gaap:FinishedGoods
Work in Progress	[Concept] Monetary	Debit	1	gaap:WorkInProgress
Raw Material	[Concept] Monetary	Debit	1	gaap:RawMaterial

[24] Fact Table, TO DO…

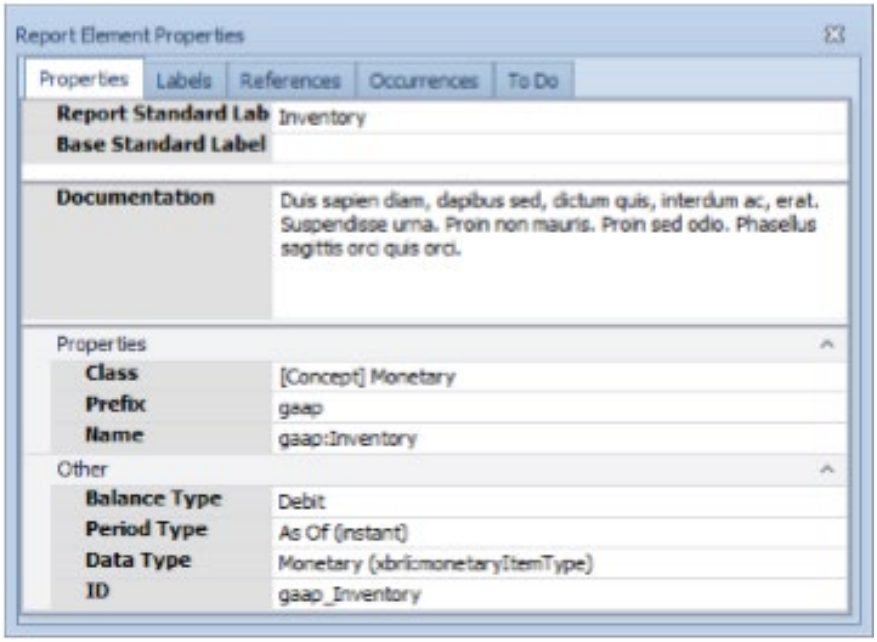
Software applications use the rule relations that describe or explain the relations to verify that reported facts are consistent with that explanation. Here is a software application interface for verifying that the reported facts are consistent with the rules that explain the relations between the facts:

Label	Rendered Value	Op	Reported Value	Calculated Value	Balance	Result	Name
Inventory, by Component [Line Items]							gaap:InventoryByComponentLineItems
Inventory, by Component [Roll Up]							gaap:InventoryByComponentRollUp
Finished Goods	600	+	600		Debit		gaap:FinishedGoods
Work in Progress	300	+	300		Debit		gaap:WorkInProgress
Raw Material	100	+	100		Debit		gaap:RawMaterial
Inventory	1,000		1,000	1,000	Debit	Verified	gaap:Inventory

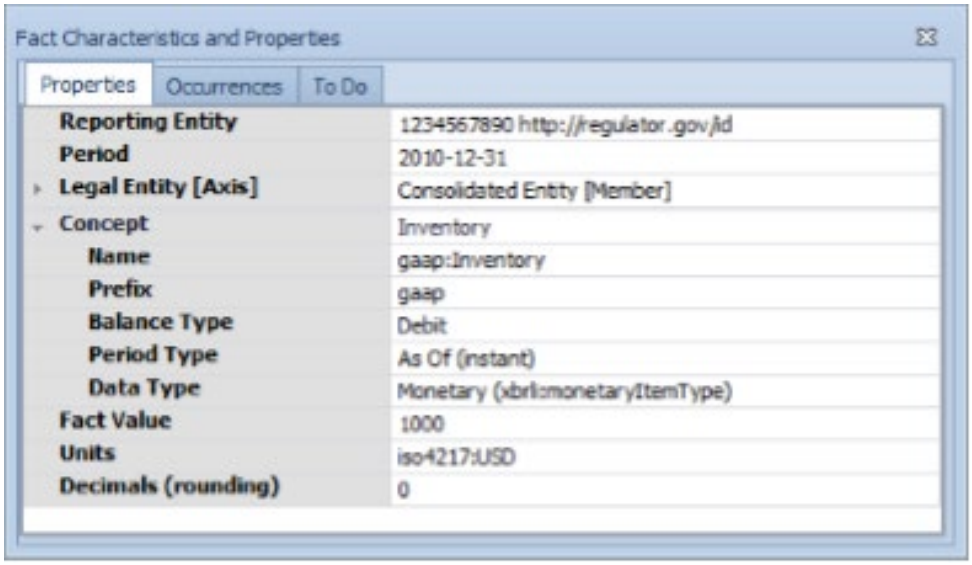
Label	Rendered Value	Op	Reported Value	Calculated Value	Balance	Result	Name
Inventory, by Component [Line Items]							gaap:InventoryByComponentLineItems
Inventory, by Component [Roll Up]							gaap:InventoryByComponentRollUp
Finished Goods	600	+	600		Debit		gaap:FinishedGoods
Work in Progress	300	+	300		Debit		gaap:WorkInProgress
Raw Material	100	+	100		Debit		gaap:RawMaterial
Inventory	1,000		1,000	1,000	Debit	Verified	gaap:Inventory

Alternatively, note that the renderings provided as examples of this fact set contain two green cells which confirm that mathematical relation for the roll up total is consistent with the explanation provided by the rules.

Information about the properties of each report element which makes up the information model description should be accessible to the user of the business report:



Information about the properties of each fact which is represented within the report is accessible to the user of the business report:



This same information is provided for each and every fact set that makes up a business report. Facts could be used in multiple fact sets. The facts used in fact sets must be consistent within a fact set and between the individual fact sets that make up a report.
Remember that a financial report is a special type of business report. Every financial report is a business report; but it is not the case that every business report is a financial report. Every financial report has the characteristic of complying with the accounting equation and double-entry accounting.

C.3 Report Structure

A business report can be broken down into fragments. A **fragment**[25] is a set of one to many *fact sets* which go together some specific purpose within a report. For example, a balance sheet is a fragment of a business report that is made up of two fact sets: a roll up of assets and a roll up of liabilities and equity.
Each fact set has a concept arrangement pattern property. A **concept arrangement pattern**[26] specifies the nature of the relationship between the concept aspect of an information model definition.
A **set**[27] is a type of concept arrangement pattern where concepts have no mathematical relations between each other within the fact set. Essentially, a set is a flat list of concepts. A synonym for set is hierarchy.
A **roll up**[28] is a type of concept arrangement pattern which represents a basic roll up type mathematical relationship: Fact A + Fact B + Fact C = Fact D (a set of items and a total of those items).

---

25 Fragment, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Fragment.html
26 Level 2 Policy Text Block, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Level2PolicyTextBlock.html
27 Reporting Style, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ReportingStyle.html
28 Concept Arrangement Pattern, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ConceptArrangementPattern.html

A **roll forward**[29] is a type of concept arrangement pattern which represents a basic roll forward mathematical relation: Beginning balance (stock) + change1 (flow) + change2 (flow) + change3 (flow) = Ending balance (stock). The beginning and ending balances are two different instances in time (stock) and the changes (flow) are between those two instances.
An **adjustment**[30] is a type of concept arrangement pattern which represents a basic mathematical reconciliation between an originally stated value and a restated value usually due to a correction or error: Originally stated balance + adjustment1 + adjustment2 + adjustment3 = restated balance. The originally stated balance and restated balance are the same concept as of the same instant in time that are differentiated by the date those facts are reported. The adjustments are the changes that reconcile the originally stated to the restated balance.
A **variance**[31] is a type of concept arrangement pattern which represents a mathematical difference between two reporting scenarios: Amount (projected scenario) + Amount(variance) = Amount (actual scenario).
A **complex computation**[32] is a type of concept arrangement pattern which represents any arbitrary mathematical relationship between a set of numeric facts. A complex computation is comprised of some flat set of numeric concepts and a rule that represents the mathematical relation between that set of concepts.
A **roll forward info**[33] is a type of concept arrangement pattern which represents a non-mathematical relation of information about a roll forward.
A **text block**[34] is a type of concept arrangement pattern which represents a non-mathematical relationship in the form of prose. A text block concept arrangement pattern is comprised of exactly one concept. There are three sub classes or type of text blocks: **Level 1 Note Text Block**[35], **Level 2 Policy Text Block**[36], and **Level 3 Disclosure Text Block**[37].
Each fact set has a member arrangement pattern property. A **member arrangement pattern**[38] expresses the relations between members within an aspect other than the concept aspect (which is explained by the concept arrangement pattern).
The members of an axis might be related mathematically. **Member aggregation**[39] is a type of member arrangement pattern where the members of an axis roll up the same as the roll up concept arrangement pattern. **Member flat**[40] list is a type of member aggregation pattern where the members for a flat list. **Member nonaggregating**[41] is a type of member arrangement pattern where the members of an axis are not related mathematically but simply are used to differentiate reported facts.

---

29 Member, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Member.html
30 Flow, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Flow.html
31 Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Rule.html
32 Concept, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Concept.html
33 Text, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Text.html
34 Level 3 Disclosure Text Block, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Level3DisclosureTextBlock.html
35 Concept Aspect, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ConceptAspect.html
36 Text Block, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/TextBlock.html
37 Aspect, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Aspect.html
38 Calendar Period Aspect, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/CalendarPeriodAspect.html
39 Fact Value, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/FactValue.html
40 Exemplar, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Exemplar.html
41 Rendering, TO DO…

Reported facts could need additional arbitrary descriptive information. A **parenthetical explanation**[42] provides additional descriptive information about a fact. A synonym for parenthetical information is comment.

A financial **reporting scheme**[43] is a formal specification for how financial reports are to be created and the underlying accounting rules and is usually created by a standards setter or regulator. For example, US GAAP, IFRS, and IPSAS are all financial reporting schemes. Financial reports are not forms. Financial reporting schemes allow for a certain amount of flexibility and variability when reporting certain specific disclosures or subtotals contained within a disclosure. A **disclosure**[44] is a set of one to many fact sets or a set of one to many fragments which form an accounting disclosure that is either required by statutory or regulatory rules or provided at the discretion of a reporting entity. A **template**[45] is a representation of a possible disclosure that can be used as a prototype in the process of creating a report. An **exemplar**[46] is a representation of a disclosure from an existing report of some economic entity that can be leveraged in the process of creating a report.

Because variability exists in the allowed possible approaches that economic entities represent their financial disclosures, different economic entities have different reporting styles. A **reporting style**[47] is a set of relations, consistency crosscheck rules, mapping rules, and impute rules that are used to check fundamental accounting concept relations for a specific type of report or style of reporting. For example, a classified balance sheet and an order of liquidity balance sheet are two different reporting styles for creating a balance sheet.

A **consistency crosscheck rule**[48] is a type of rule that tests the relations of fundamental accounting concept relations within a report against a specified reporting style to make sure there are no inconsistencies or contradictions between reported facts within a report.

An **impute rule**[49] is a type of rule that explains how to logically derive a fact that have not been explicitly reported based on other facts that have been explicitly reported or which have been logically derived from other reported information. For example, an economic entity might not explicitly report the line item “Noncurrent assets”; but does report “Assets” and “Current assets”. Given the impute rule “Assets = Current assets + Noncurrent assets”; the fact value for Noncurrent assets can be reliably derived logically using the other two reported facts and the impute rule.

A **mapping rule**[50] is a type of rule that explains how a base reporting scheme taxonomy concept reported by an economic entity relates to a fundamental accounting concept. For example, the notion of “Cost of Revenue” could be reported using the concept “Cost of Revenue”, or “Cost of Goods and Services Sold”, or “Cost of Goods Sold”, or “Cost of Services Sold”, etc. Basically, mapping rules enable information to be extracted from a report reliably.

---

42 Adjustment, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Adjustment.html
43 Member Flat List, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/MemberFlatList.html
44 Impute Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ImputeTypeRule.html
45 Axis, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Axis.html
46 Report Element, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ReportElement.html
47 Reporting Checklist Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ReportingChecklistRule.html
48 Nonnumeric Fact Value, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/NonnumericFactValue.html
49 Rounding, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Rounding.html
50 Set, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Set.html

A **disclosure mechanics rule**[51] is a type of rule that describes the structural and mechanical representation of a disclosure against a specification or prototype of that disclosure. For example, every disclosure that has the property of concept arrangement pattern of "roll up" must always have a total. A disclosure mechanics rule would specify the concept that would be used to represent that total. A specific disclosure, such as "inventory components roll up" would be required to use a specific concept such as "Inventory, Net" to represent that total. A disclosure mechanics rule would specify that concept. Other concepts might be used as alternatives to some specific total concept to represent a disclosure. A disclosure mechanics rule would specify those alternatives. Every Level 4 Disclosure Detail representation has some complementary Level 3 Disclosure Text Block representation. A disclosure mechanics rule would specify that relation.

A **type or class rule**[52] is a type of rule that expresses an allowed or a disallowed relation between two reporting scheme concepts for some reporting style. For example, the concept "Operating Expense (indirect operating expense)" would never be part of "Cost of Revenue (direct operating expense)", a type or class rule would be used to explicitly disallow this relation. Alternatively, explicitly allowed relations are also expressed using type or class rules.

A **reporting checklist rule**[53] is a type of rule that describes the reportability of a statutory or regulatory disclosure required by a reporting scheme. For example, some disclosures are always required. Other disclosures are required only if a specific line item is reported. Other disclosures could be used as alternatives for some other disclosure.

A **report set**[54] is a set of one to many reports. For example, if you are comparing the reports of an economic entity for the past five years, the five reports that you use to perform that analysis are your report set.

A **reporting entity aspect**[55] is a core aspect that distinguishes the economic entity which creates a report.

A **calendar period aspect**[56] is a core aspect that distinguishes the calendar period of a reported fact. A **stock**[57] is a type of calendar period aspect that is used to represent a fact as of a specific point in time. A synonym for stock is instant. A **flow**[58] is a type of calendar period aspect that is used to represent a fact over a period of time. A synonym for stock is duration.

A **concept aspect**[59] is a core aspect that is used to express the concept that relates to a fact. Synonyms for concept aspect include primary item and line item.

A **fragment arrangement pattern**[60] is the relationship between fragments or the order or sequence of fragments within a report.

---

51 Type or Class Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/TypeClassRule.html
52 Mapping Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/MappingTypeRule.html
53 Member Aggregation, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/MemberAggregation.html
54 Roll Forward Info, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/RollForwardinfo.html
55 Parenthetical Explanation, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ParentheticalExplanation.html
56 Stock, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Stock.html
57 Units, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Units.html
58 Reporting Scheme, TO DO, http://xbrlsite.azurewebsites.net/2018/Library/ReportingSchemes-2018-12-30.pdf
59 Fragment Arrangement Pattern, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/FragmentArrangementPattern.html
60 Level 1 Note Text Block, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Level1NoteTextBlock.html

Prose[61] is a type of fact value that is structure in nature (i.e. a table, an ordered list, an unordered list, paragraphs of text, or any combination of those structures).
Text[62] is a type of fact value that is nonnumeric unstructured text (i.e. not prose).
A **logical rule**[63] is a type of rule expresses logical relations between entities that make up a report.
An **accounting rule**[64] is a type of logical rule that is used to express a logical assertion specifically related to accounting rules.
A **mechanical rule**[65] is a type of logical rule that is used to express the relations between the report elements that make up a disclosure.

61 Roll Forward, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/RollForward.html
62 Accounting Rule, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/AccountingRule.html
63 Core Aspect, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/CoreAspect.html
64 Prose, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/Prose.html
65 Report Set, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/ReportSet.html

Annex D. Simple Narrative Example (Informative)

The following provides a very basic example of using SBRM to represent a business financial report. An example from financial reporting is used to provide this very basic example. Most business professionals are familiar with the accounting equation[66] which is "Assets = Liabilities and Equity" and represents the high-level financial elements that would appear within a balance sheet.

We will use this very short list of terms and the equation to demonstrate how SBRM would be used to create a provably properly functioning logical system.

The first step would be to represent the terms, associations between the terms, assertions, structures, and then express a set of facts. Here is this information in the form of statements which define the pieces of this business report logical system:

- **Terms**:
  - Assets is-a simple term.
  - Liabilities is-a simple term.
  - Equity is-a simple term.
  - Balance sheet is-a functional term.
- **Properties**:
  - Assets has balance type of debit.
  - Liabilities has balance type of credit.
  - Equity has balance type of credit.
  - Assets has period type of instant.
  - Liabilities has period type of instant.
  - Equity has period type of instant.
- **Structure**
  - Balance sheet has-part Assets.
  - Balance sheet has-part Liabilities.
  - Balance sheet has-part Equity.
- **Rules**:
  - Assets = Liabilities + Equity
- **Facts**:
  - ABC Company is an entity.
  - Assets for December 31, 2019 for ABC Company is $5,000 US Dollars.
  - Liabilities for December 31, 2019 for ABC Company is $1,000 US Dollars.
  - Equity for December 31, 2019 for ABC Company is $4,000 US Dollars.

The statements are not machine readable, but they could be[67]. So above you see all of the statements that exist within this very basic example of a financial report that is comprised of a balance sheet that reports high-level facts that relate to the accounting equation.

---

66 Fact Set, http://accounting.auditchain.finance/sbrm/2023/Framework/Details/FactSet.html

67 Wikipedia, *Attempto Controlled English*, https://en.wikipedia.org/wiki/Attempto_Controlled_English

Per all the statements, the logical system appears to be properly functioning. All the terms are defined as are properties, rules, and facts. To be sure the point is being understood consider making two changes to the existing logical system.

Suppose that the value for the fact "Assets" was changed to say $5,500. If that were the case, because there is a rule that specifies that "Assets = Liabilities + Equity" and because of the values of Liabilities and Equity, the logical system would NOT be functioning properly. This is because $5,500 = $1,000 + $4,000 is inconsistent with other supporting statements within the logical system. If that provided rule, "Assets = Liabilities + Equity" did not exist, then the logical system would not be considered "complete" but the inconsistency would not be detected by automated processes.

However, since all the statements are provided and the statements are consistent with one another we can say that the logical system that is used to represent this very basic business report is precise and complete because (a) it does reflect reality given the statements provided and (b) no statement seems to be missing from the logical system.

Above we have verified the logical system using human-readable statements and the rules of logic. We will now explain the key highlights of the logical model using this very basic report to provide a high-level overview of the logical conceptualization of a business report.

An **information model description** is created for the **report**:

#	Label	Report Element Class	Period Type	Balance	Name
1	**Core [Table]**				(Implied)
2	*Balance Sheet [Abstract]*	[Abstract]			core:BalanceSheetAbstract
3	Assets	[Concept] Monetary	As Of	Debit	core:Assets
4	Liabilities	[Concept] Monetary	As Of	Credit	core:Liabilities
5	Equity	[Concept] Monetary	As Of	Credit	core:Equity

This very basic report has one **fact set**:

#	Reporting Entity [Axis]	Period [Axis]	Concept	Fact Value	Unit	Rounding	Parenthetical Explanations
1	GH259400TOMPUOLS65II (http://standards.iso.org/iso/17442)	2020-12-31	Assets	5000	USD	INF	
2	GH259400TOMPUOLS65II (http://standards.iso.org/iso/17442)	2020-12-31	Liabilities	1000	USD	INF	
3	GH259400TOMPUOLS65II (http://standards.iso.org/iso/17442)	2020-12-31	Equity	4000	USD	INF	

The fact set contains three **facts**. The facts are distinguished from one another using three **aspects**: reporting entity, period, and concept. The **fact value** is numeric and described by the **unit** and **rounding property**. The rendering is created

Balance Sheet [Abstract]	Period [Axis]
	2020-12-31
Balance Sheet [Abstract]	
Assets	5,000
Liabilities	1,000
Equity	4,000