Surveillance Manager Interface Specification

This OMG document replaces the draft adopted specification (dtc/2001-12-01). It is an OMG Final Adopted Specification, which has been approved by the OMG board and technical plenaries, and is currently in the finalization phase. Comments on the content of this document are welcomed, and should be directed to *issues@omg.org* by October 21, 2002.

You may view the pending issues for this specification from the OMG revision issues web page *http://www.omg.org/issues/*; however, at the time of this writing there were no pending issues.

The FTF Recommendation and Report for this specification will be published on November 25, 2002.

OMG Final Adopted Specification
August 2002

Surveillance Manager Interface Specification

Final Adopted Specification
August 2002

ISSUE REPORTING

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

Contents

Contents

Preface

About This Document

Under the terms of the collaboration between OMG and The Open Group, this document is a candidate for adoption by The Open Group, as an Open Group Technical Standard. The collaboration between OMG and The Open Group ensures joint review and cohesive support for emerging object-based specifications.

Object Management Group

The Object Management Group, Inc. (OMG) is an international organization supported by over 600 members, including information system vendors, software developers and users. Founded in 1989, the OMG promotes the theory and practice of object-oriented technology in software development. The organization's charter includes the establishment of industry guidelines and object management specifications to provide a common framework for application development. Primary goals are the reusability, portability, and interoperability of object-based software in distributed, heterogeneous environments. Conformance to these specifications will make it possible to develop a heterogeneous applications environment across all major hardware platforms and operating systems.

OMG's objectives are to foster the growth of object technology and influence its direction by establishing the Object Management Architecture (OMA). The OMA provides the conceptual infrastructure upon which all OMG specifications are based. More information is available at http://www.omg.org/.

The Open Group

The Open Group, a vendor and technology-neutral consortium, is committed to delivering greater business efficiency by bringing together buyers and suppliers of information technology to lower the time, cost, and risks associated with integrating new technology across the enterprise.

The mission of The Open Group is to drive the creation of boundaryless information flow achieved by:

- Working with customers to capture, understand and address current and emerging requirements, establish policies, and share best practices;
- Working with suppliers, consortia and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies;
- Offering a comprehensive set of services to enhance the operational efficiency of consortia; and
- Developing and operating the industry's premier certification service and encouraging procurement of certified products.

The Open Group has over 15 years experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification. The Open Group portfolio of test suites includes tests for CORBA, the Single UNIX Specification, CDE, Motif, Linux, LDAP, POSIX.1, POSIX.2, POSIX Realtime, Sockets, UNIX, XPG4, XNFS, XTI, and X11. The Open Group test tools are essential for proper development and maintenance of standards-based products, ensuring conformance of products to industry-standard APIs, applications portability, and interoperability. In-depth testing identifies defects at the earliest possible point in the development cycle, saving costs in development and quality assurance.

More information is available at http://www.opengroup.org/ .

Associated OMG Documents

The CORBA documentation is organized as follows:

- *Object Management Architecture Guide* defines the OMG's technical objectives and terminology and describes the conceptual models upon which OMG standards are based. It defines the umbrella architecture for the OMG standards. It also provides information about the policies and procedures of OMG, such as how standards are proposed, evaluated, and accepted.
- *CORBA: Common Object Request Broker Architecture and Specification* contains the architecture and specifications for the Object Request Broker.
- *CORBA Services: Common Object Services Specification* contains specifications for OMG's Object Services.
- *CORBA Common Facilities*: contains services that many applications may share, but which are not as fundamental as the Object Services.
- CORBA domain specifications are comprised of stand-alone documents for each specification; however, they are listed under the domain headings, such as Telecoms, Finance, Med, etc.

OMG collects information for each book in the documentation set by issuing Requests for Information, Requests for Proposals, and Requests for Comment and, with its membership, evaluating the responses. Specifications are adopted as standards only when representatives of the OMG membership accept them as such by vote.

To obtain OMG publications, contact the Object Management Group, Inc. at:

OMG Headquarters
250 First Avenue, Suite 201
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
pubs@omg.org
http://www.omg.org

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Helvetica bold - OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier bold` - Programming language elements.

Helvetica - Exceptions

If applicable, terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Acknowledgments

The following company submitted this specification:

- THALES ATM

Reference Documents

OMG, Surveillance Request for Proposal, *transprt/00-01-09*

Eurocontrol, Radar Surveillance in En-Route Airspace and Major Terminal Areas, Edition 1.0 from March 1997, SUR.ET1.ST01.1000-STD-01-01

Demystifying CNS/ATM, CANSO CNS/ATM Working Group, Final Version (June 1999)

Overall CNS/ATM architecture for EATCHIP, ASE.ET1.ST02-ADD-01-00, version 1.0, 18/08/1997

Interface Specification, Application of ASTERIX to ARTAS (DIS/SUR/ARTAS.ASTX.015), version 6.0, 10th Sept. 99)

Surveillance Development Roadmap, Eurocontrol, working draft, edition 0-18, 4 December 2000

System/Segment Specification for ARTAS, version 6.3, 10 October 2000

Overview 1

1.1 Guide to the Material in the Specification

~~The relentless growth of air traffic is causing more and more congestion and delays, and as the situation worsens, it might threaten the freedom to travel, and economic growth. Another, often less visible, problem related to the traffic level is safety: the number of incidents (and sometimes accidents) tends to increase, mainly at airports, because of the increasing traffic. This situation is expected to still worsen rapidly, leading to a breaking point where the capacity cannot meet the demand, with serious consequences on economical growth.~~

~~Given its complexity and the number of stakeholders involved in Air Traffic Management, solutions to this problem may only be at an international level, by introducing new concepts into air traffic management and by favoring homogeneity (e.g., standardization is one way to homogeneity).~~

~~Major ATM regulation organizations (e.g., FAA, EUROCONTROL) have recognized that need for introduction of new concepts and have developed relevant strategies which generally cover a scope encompassing the next ten years. In the Surveillance Domain, these concepts can be summarized through the introduction of new surveillance means to estimate and report aircraft position to the ATM System (e.g., Mode S, ADS). To allow for an optimal - in terms of expected increase of capacity and safety - use of these new capabilities, a relevant improved Ground Surveillance Architecture shall be designed and consequently put into operation. Such an architecture, for the part concerning the ATM System, is precisely the one that defines the functional scope of the current RFP this specification.~~

~~Moreover, these strategies have also recognized the need to develop interfaces standardization for interoperability to the greater extent as possible greatest extent possible. This interoperability should be effective inside ATM systems, between ATM systems, and between ATM Systems and external Systems.~~

~~Such a need is clearly justified through new concept requirements (extended co-operation between ATM systems and between ATM systems / external systems to improve predictability and allow for collaborative decision making) and also through cost rationales for system developments (capability to change a component of the system without impacting the whole system). It is expected that the interest of such standardization will be common to the Users of Technology (i.e., ATS Providers) and Industrials as the current situation leads to inflating budget and planning because of the low level of standardization of ATM Systems.~~

~~Using CORBA technology to offer modular, object-oriented ATM Systems based on client-server principle, is certainly one solution that will allow for the required standardization of the interfaces of those ATM Systems meanwhile re-enforcing the independence of components.~~

~~Therefore, the present document this specification aims at populating this standardization through proposing some relevant interface specifications (using the CORBA Interface Definition Language or IDL) for the Surveillance Manager. as described in OMG Surveillance RFP.~~

~~AIRSYS ATM, issuing the current proposal, is one major ATM Manufacturer offering a complete range of products for the CNS/ATM environment and has constant close dialogue with Air Traffic Service Providers, air traffic and navigation regulation agencies, and airlines throughout the world to contribute to defining new standards and specifications.~~

~~More particularly, AIRSYS ATM is offering a high level of expertise in the Surveillance Domain and its current ATM Systems may integrate two different Multi-Radar Tracker systems both complying with the performance specifications described in the document~~ ~~*Eurocontrol, Radar Surveillance in En-Route Airspace and Major Terminal Areas,*~~ ~~Edition 1.0 from March 1997, SUR.ET1.ST01.1000-STD-01-01 (one is AIRSYS ATM own product; the other one is ARTAS, a Eurocontrol product developed by AIRSYS ATM). AIRSYS ATM also completed an ARTAS2 feasibility study for Eurocontrol, concerning the extension of the current ARTAS functionalities and architecture to cover the same functional scope as the current RFP this specification.~~

~~Thus, it may be ensured that AIRSYS ATM is proposing the current answer with all required expertise.~~

~~*1.2 Overall Design Rationale*~~

~~ATM is an environment where regulation is of an extreme importance to allow for necessary performances and safety of the systems. Therefore, interfaces shall not overcome or substitute any past or future decision that should be taken by national or international regulation authorities.~~

~~Transition is one of the most difficult problems facing ATM systems. Even though object-oriented CORBA based architecture will, in the future, facilitate the evolution of those systems, the current situation will make it difficult to drive sufficient budget to take full, immediate advantage of the capabilities offered by this new technology. Therefore, the solution shall be implemented quite easily by current functions.~~

~~Besides, it shall be taken care that this solution will be upgraded in a well-understood, manageable, cost-effective sequence of improvements that keep pace with user needs and with respect to overall safety, capacity, system efficiency, regularity, and environmental requirements.~~

~~*1.2.1 Scope*~~

~~Interface standardization should only be performed when underlying function performances –in the general sense- have been clearly specified and validated (through prototyping or other means), and there is an agreement at appropriate community level upon these performances.~~

~~Thus, in the surveillance architecture presented in this specification, the following functions are not available in operational environment and even still in the process of functional evaluation (i.e., their functional performances are not yet stabilized):~~

- ~~Track Support and Hand-over Management~~
- ~~Dynamic Airborne Parameter Cache~~
- ~~Fusing -with Radar Data- of position/kinematics information coming from different types of Data Sources (ADS, Mode S).~~
- ~~Surveillance Server with capabilities extended to support the provision of Derived Aircraft Parameters.~~

~~Furthermore, the introduction of new technologies to provide surveillance report information to the ground will not be immediate. For example, a report prepared by the Civil Air Navigation Services Organization (CANSO), see the document *Demystifying CNS/ATM*, CANSO CNS/ATM Working Group, Final Version (June 1999), foresees that Mode S Enhanced Surveillance, ADS-B (based on ICAO SARPs compliant systems), A-SMGCS will not be operationally available before 2005.~~

~~Concerning Enhanced Surveillance, through Mode S or ADS-B, this is also the assumption of Eurocontrol as defined in the document *Surveillance Development Roadmap*, Eurocontrol, working draft, edition 0-18, 4 December 2000.~~

~~In this area (Enhanced Surveillance), such other processes of standardization such as ASTERIX have not yet reached stability.~~

~~Therefore, this specification takes these previous considerations into account and does not define interfaces when covering one of the above items.~~

~~*1.2.2 Evaluation of the Design*~~

~~*1.2.2.1 Interfaces for which Object Framework?*~~

~~It is recalled (from the OMG document, *Surveillance Request for Proposal*, transprt/00-01-09), “A key goal of OMG is create a standardized object-oriented architectural framework for distributed applications based on specifications that enable and support distributed object. The Reference Model identifies and characterizes the~~

~~components, interfaces, and protocols that compose the OMA…A second part of the Reference Model introduces the notion of domain-specific *Object Frameworks. An Object Framework component is a collection of co-operating objects that provide an integrated solution within an application or technology domain and which is intended for customization by the developer or user…. Object Frameworks are collections of co-operating objects categorized into Application, Domain, Facility, and Service Objects. Each object in a framework supports (through interface inheritance) or makes use of (via client requests) some combination of Application, Domain, CORBA facilities, and CORBA services interfaces. A specification of an Object Framework defines such things as the structure, interfaces, types, operation sequencing, and qualities of service of the objects that make up the framework. This includes requirements on implementations in order to guarantee application portability and interoperability across different platforms.*"~~

~~Basically, when looking at the architecture design given in Figure 2 of the OMG document, *Surveillance Request for Proposal*, transprt/00-01-09, the RFP would seem to concern the interfaces of a "***Surveillance Manager Object Framework" (between objects composing the Surveillance Manager), but also interfaces of an "ATM Ground System Object Framework", for those in relation with ATM users of the surveillance information, and also interfaces of a "CNS/ATM System Object Framework" for those concerning exchanges with Surveillance Data Sources.***~~

~~Therefore, the design proposed and the relevant requested interfaces would not be coherent to one Object Framework only, but would be crossing partially three "encapsulated" Object Frameworks (*Surveillance Manager Object Framework ∈ ATM Ground System Object Framework ∈ CNS/ATM System Object Framework).*~~

~~The scope of each of these architectures may be clearly understood when referring to the document *Overall CNS/ATM Architecture for EATCHIP*, ASE.ET1.ST02-ADD-01-00, version 1.0, 18/08/1997 (Chapter 3, Scope and context of the CNS/ATM). To simplify: the *CNS/ATM system would be composed of ATM, CNS and Aeronautical Environment System.*~~

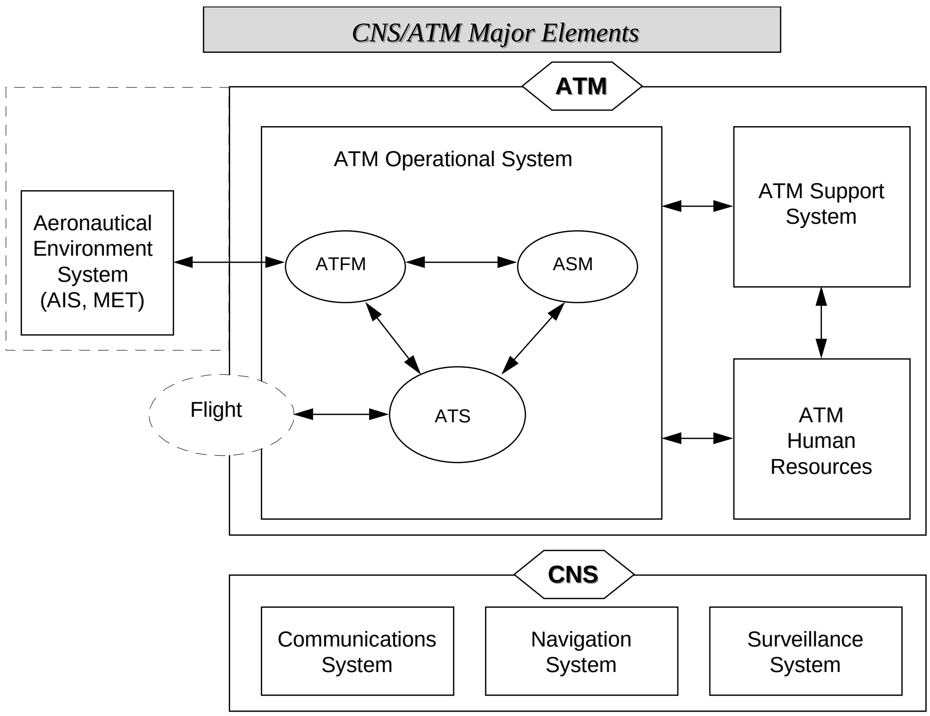

Figure 1-1 ~~CNS/ATM Major Events~~

~~Air Traffic Management (ATM) consists of a ground part (Ground-BasedATM) and an air part (Airborne ATM (e.g., FMS, TCAS...)), where both are needed to ensure a safe and efficient movement of aircraft during all phases of operations.~~

~~The primary component of Ground-Based ATM is Air Traffic Service (ATS), and its adjunct components are Airspace Management (ASM) and Air Traffic Flow Management (ATFM). ATS (referred to as Ground ATM System in this specification) may in fact be composed of several ATM Ground System components. A Surveillance Manager is part of one ATM Ground System.~~

~~Communications, Navigation and Surveillance (CNS) form part of the infrastructure used by the Air traffic Management for the identification and location of air traffic, and for the distribution of information. The CNS system covers more than the technical architecture. It encompasses the organization, people, infrastructure, procedures, rules and information used to provide the services.~~

~~This is to stress that this specification should be inscribed in a logical sequence for the overall CNS/ATM object framework architecture with a clear rationale justifying for that sequence.~~

~~1.2.3 Interfaces~~

~~Interfaces of the Surveillance Manager with the Surveillance Data Sources~~

~~Generally, AIRSYS ATM agrees with those (functional) interface requirements as described in this specification; however, it also supports that the definition of these interfaces should be linked to a more general RFP (if possible to describe one) considering a "~~*~~CNS/ATM System Object Framework." These interfaces do concern server objects (i.e., Data Sources) external to the architecture of a Ground ATM System but relevant to the overall CNS/ATM System architecture as CNS components. In this architecture, one Ground ATM System would be one component, another Ground ATM System would be another one, an Airline Operational Centre another one, the Surveillance Data Sources other ones, etc.~~*

~~Then, if it seems feasible, from now, to start the standardization of a CORBA architecture for the Ground ATM System and to prove its feasibility, the question is much more complex for a global CNS/ATM System.~~

~~When coming to the CNS environment, it involves lots of existing (and to come) communication infrastructures for which investment is very high and decisions for implementation complex (not coming from the same authority). So if one goes for an attempt to Surveillance Data Sources Interfaces IDL standardization, he should really take care that there is a true support for standardization (and relevant performing infrastructure), which may only be obtained at international level and through complex procedures.~~

~~Therefore, with regards to these Surveillance Data Sources, the following assumptions are made:~~

- ~~For the time being,~~ **~~no IDL standardization is assumed for these interfaces: the interfaces are those required by the local conditions for connection to Data Source (e.g., it could be ASTERIX 01/02, Ethernet/Mac LLC1 for a radar).~~**
- ~~In the proposed~~ *~~ATM Ground System Object Framework architecture, the responsibility to interface with the Surveillance Data Sources for the ATM System is given to the Surveillance Manager component (hypothesis on the Ground ATM System architecture).~~*

~~Interfaces of Surveillance Manager with ATM Users of Information~~

~~Apparently, there is a problem in requesting description of some interfaces of an Object Framework when,~~ *~~a priori~~*~~, the feasibility could not be proven at the level of that Object Framework (i.e., Definition of the interfaces between Surveillance Manager and ATM users of elaborated Surveillance Information would require a prove of feasibility at Ground ATM System level).~~

~~In an ideal way, this should proceed from a top-down approach:~~

- ~~Define (and have acceptance of) the architecture of the~~ *~~ATM Ground System Object Framework, i.e., the collection of ATM objects composing that architecture (Surveillance Manager, Flight Data Processing, Safety Nets, Controller Visualization…).~~*

- ~~Define the dynamic behavior of that system through, as part of it, defining the interfaces (here IDL specifications) offered by all objects.~~
- ~~Implement such a system architecture and verify its operational performances.~~

~~However, for various reasons, it does not seem feasible to aim, at once, at a complete specification and standardization of the *ATM Ground System Object Framework (as said in paragraph General assumptions for the rationale, incremental approach shall be favored in the ATM environment).*~~

~~Therefore, AIRSYS ATM supports the RFP underlying strategy to progress "object by object" for the *ATM Ground System Object Framework, focusing on the Domain Interfaces for one Object and building from successive RFPs the interface specifications of the complete Object Framework.*~~

~~Moreover, AIRSYS ATM supports the current identification of Surveillance Manager as being one basic component of the *ATM Ground System Object Framework (Even though, at least, the complete identification –and acceptance- of all basic components/objects composing the Object Framework would be highly desirable at that step. Which is not).*~~

~~Thus, the current document is proposing specifications concerning interface between Surveillance Manager and ATM users of elaborated Surveillance Information (i.e. of the Air Situation Picture).~~

~~Moreover, these specifications will consider:~~

- ~~The functional restrictions as described in paragraph Scope,~~
- ~~That the interface proposed should be robust towards expected performances but as simple to allow for a first implementation with minimal changes to currently existing Surveillance Manager and Surveillance Users functions.~~

~~*Interfaces of the Surveillance Manager Object Framework (between "potential" Objects composing the Surveillance Manager)*~~

~~As said in Section 1.2.2.1, "Interfaces for which Object Framework?," on page1-3, the architecture described in Figure 1-1 on page 1-5 could address the *Surveillance Manager Object Framework as it seems to identify components as Multi-Sensor Tracker, Dynamic Airborne Parameter Cache, Track Support and Hand-over Management, Surveillance Environment Assessment and Surveillance Server communicating through the CORBA II ORB.*~~

~~In fact, after studying the RFP and particularly what interfaces are required, our interpretation is that the figure is presenting a mixed functional-design view and is not presenting the above functions (Multi-Sensor Tracker, Dynamic Airborne Parameter Cache…) as objects in the *Surveillance Manager Object Framework architecture, but as functional parts of the Surveillance Manager component. Therefore no specifications of IDL interfaces between those "components" will be defined in the current specification.*~~

~~Even if the previous may sound obvious for some people, we found it necessary to include such a paragraph for readers who may encounter some difficulties with Figure 1-1 on page 1-5.~~

~~of the RFP (It is advised to have a clearer representation in a later version of the RFP).~~

~~However, such a design in objects would certainly need to be studied in the future and corresponding specifications for the~~ ~~*Surveillance Manager Object Framework issued. This could be done at the time when the IDL interfaces for the Surveillance Manager have matured, and Dynamic Airborne Parameter Cache and Track Support and Hand Over Management functions have been tested and clear operational requirements do exist for those functions. Then, a design identifying these functions as objects would really proceed from a coherent top-down approach*~~~~.~~

~~1.3 OMG approach of the ATM Domain~~

~~As said previously, any RFP should be described within a context and issued as resulting from a logical scheme.~~

~~In order to help in the elaboration of these context and scheme, the current paragraph is proposing some approach of the CNS/ATM Domain that could be debated by the OMG for adoption.~~

~~Therefore, Figure 1-2 on page 1-9 presents the process of (OMG type) standardization of the CNS/ATM domain in an “ideal” top-down approach.~~

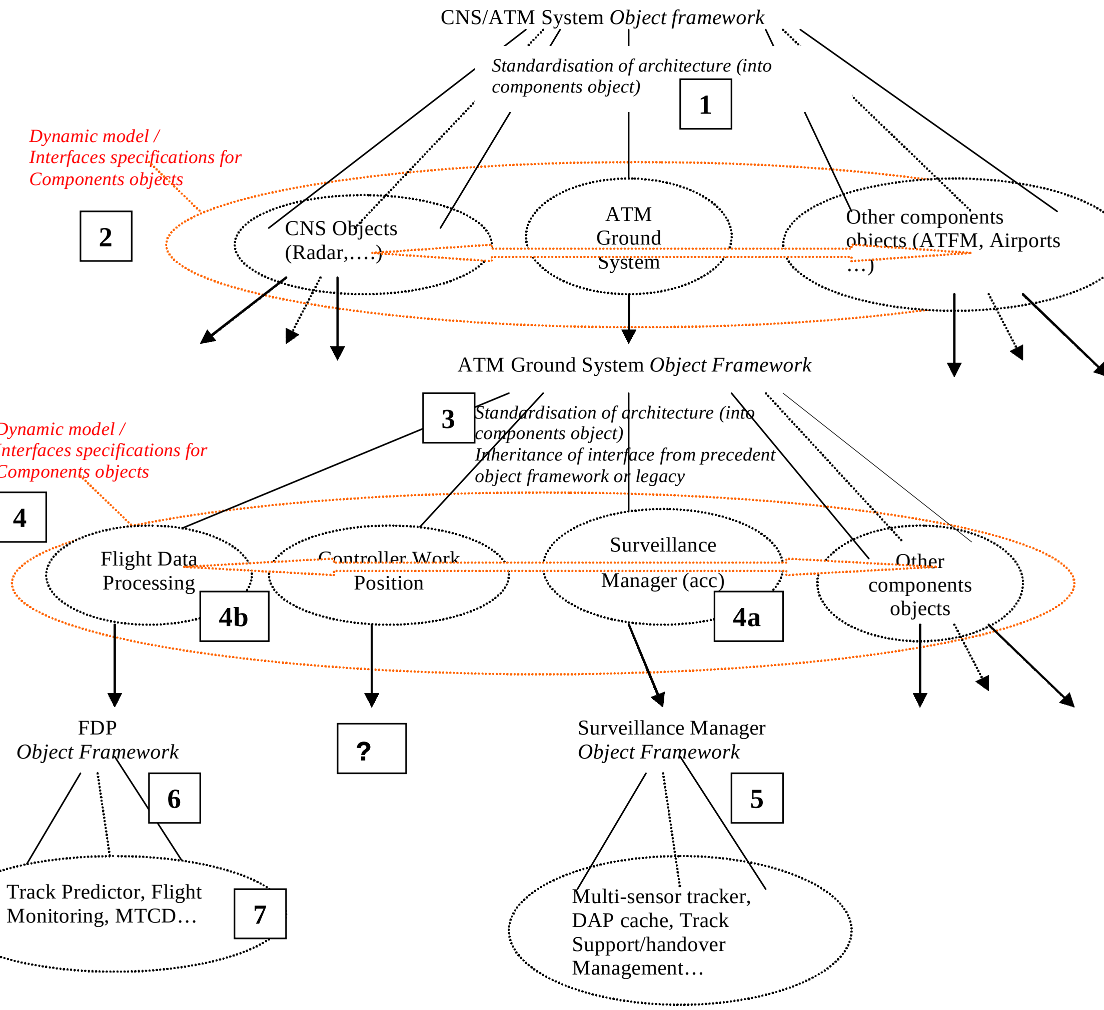

Figure 1-2 ~~CNS/ATM Domain (OMG type) Standardization: top-down approach~~

~~Figure 1-2 presents the process of (OMG type) standardization of the CNS/ATM domain in an "ideal" top-down approach.~~

~~The boxed numbers in the figure correspond to the steps listed in the following table. The table presents the AATM view of what may effectively be done, by now, in that process.~~

STEP		Evaluation	Recommended action for OMG
1	~~**Elaboration of the CNS/ATM System *Object Framework* architecture (into *component objects*)**~~	~~Currently there is no clear identification of such an architecture for standardization. However, we may take assumptions on some potential component objects (basing our analysis on the EATCHIP architecture):~~ ~~Each type of surveillance Data Source may be a Component Object (Classical Radars, Mode S Stations, ADS-B Stations, etc.).~~ ~~At this level, the ATM Ground System may be identified as a component. Other components could be identified (ATFM, Meteo Servers, Airports, Environment Data servers, ATN Router, etc.).~~	~~**(See below, first point)**~~
2	~~**Dynamic Model Execution for the *CNS/ATM System Object Framework* architecture / IDL Interfaces specifications**~~	~~Due to the heterogeneity of the concerned Components Objects, of the very different types of manufacturers for those components, of the different CNS/ATM environments throughout the world and of the safety aspects to consider, any interface standardization with respect to this framework should be issued by international regulation authorities concerned with those subjects.~~ ~~This standardization is emerging through standards as ASTERIX, OLDI, ADEXP, ODIAC, ATN, Meteo standards, IFPS, etc., but it will be a long (and iterative) process.~~	~~**Such a high level CNS/ATM architecture (with all applicable standards interfaces and performances) should be produced, as the initial reference to OMG ATM domain, but International Regulations Authorities already concerned with OMG ATM tasks (i.e., EUROCONTROL and FAA).**~~ ~~Concerning "translation" of those existing standards into IDL interfaces specification, some domains may be quite mature to perform IDL specifications (e.g., Classical radar) and therefore RFPs particular to these domains could be issued.~~ ~~The problem is that Industrials concerned by those domains should be convinced of the future use of those IDLs to answer such RFPs. This in fact is linked to the existence of a communication infrastructure (at various levels: regional, national, and international) supporting these new architectural concepts.~~

STEP		Evaluation	Recommended action for OMG
3	~~**Elaboration of the Ground ATM System *Object Framework* architecture (into *component objects*)**~~	~~There are already some architecture models existing (e.g., AVENUE, which is a project within the European Commission 4th Framework Programme).~~	~~**OMG should identify and get approval of a first version of this architecture. This will make clear in which context are "operating" such RFP as the current one for Surveillance Manager.**~~
4	~~**Dynamic Model Execution for the Ground ATM System *Object Framework* architecture / IDL Interfaces specifications**~~	~~It is clear that there is current interest (and fertile ground) to standardize and promote such specifications at the level of Ground ATM systems. However, it is not possible to standardize the total dynamic model of operation at once.~~ ~~At this time, there is no evidence that any organization could produce such complete specifications and prove their operational feasibility.~~ ~~This would certainly create a huge debate between usual Providers of such Systems and delay for acceptance could be very long.~~	~~**The way to proceed is to populate the specifications through successive RFPs concerning particular components of the architecture (this is what is actually done by OMG). This sequence of RFPs should be determined and the relevant logic described.**~~ ~~However, approval of an RFP (for a particular component) should always be done with the perspective of the whole Ground ATM System operation.~~ ~~Acceptance should only be provisional.~~ ~~The formal acceptance is only possible when the sequence of RFPs is completed and there is demonstration of a corresponding ATM System fulfilling required operational performances.~~ ~~When proposing an answer for a particular RFP, proposal for revision of previous "adjacent" RFPs should be possible when clearly justified upon feasibility or conceptual improvement, and demonstration.~~ ~~This is quite evident that there will be iteration on the various RFPs and on the global specifications.~~
4a	**RFP for description of the interfaces of the Surveillance Manager in the Ground ATM System architecture**		**As this component is mainly a Server function towards the other objects of the architecture, Airsys ATM approves this RFP to be elected as one of the first RFPs in the sequence.** **The present document is the AATM relevant proposal.**

STEP		Evaluation	Recommended action for OMG
4b	~~**RFP for description of the interfaces of Flight Data Processor**~~	~~This is somehow the "heart" of the ATM Ground System and clearly the most complex function for which specifications of interfaces should be considered very cautiously.~~ ~~There already exists some proposed decomposition (e.g., AVENUE) of the FDP components into some object components (Trajectory Predictor, Flight Monitor, etc.).~~ ~~This decomposition (and relevant interface specifications) will be taken further away as part of different European projects concerned with the development of solutions for a "European Flight Data Processor."~~	~~**It would be interesting to group the steps 4a, 6 and 7. That is to have an agreement on the decomposition of FDP into components (at a first level) and require interfaces specifications for that architecture integrated into the Ground ATM System framework.**~~ ~~**This RFP should only be issued when there is evidence that some industrial projects are launched to develop such solutions. (The costs for development with regards to new generation of FDP are very high and the technical aspects very complex. Such standardization should rely on more solid ground than only prototyping).**~~
?		~~**The question mark concerns a previously issued OMG RFP for ATM Display Manager interfaces.**~~ ~~**Looking at the proposal, it is very difficult to figure out from which logical sequence is derived this RFP as it does not seem to contribute to the description of the Ground ATM System Dynamic Model (interfaces between components), but aims at standardizing a design for HMI functions.**~~ ~~**Is this really the goal of the OMG in the ATM Domain when it is also required that proposals should be implementation independent ?**~~	
5	~~**Elaboration of the Surveillance Manager *Object Framework* architecture (into *component objects*).**~~		~~This is not feasible for the time being.~~
6,7	~~See 4b~~		

Specification 2

2.1 Introduction

This part of the document will be populated by the FTF.

~~The process used to provide specifications is shown in Figure 2-1. Each step is described as well.~~

~~The process followed to answer the RFP and provide according specifications is the following:~~

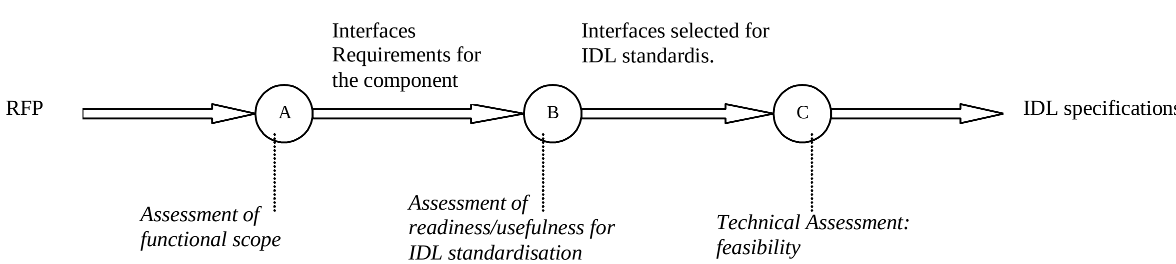

Figure 2-1

~~Chapter 1 already describes Steps A and B (with all necessary justifications). These steps are summarized below.~~

~~In the part I so far, we have already gone through steps A and B (with all necessary justifications). These steps are roughly summarised hereafter:~~

~~STEP A~~

~~Functional Scope~~	**~~Assessment~~**	**~~Interfaces~~**
~~Multi-radar fusion with the level of performances as specified in and provision of relevant Air Situation picture~~	~~Yes~~	~~Interfaces with Classical radar and Mode S Stations.~~ ~~Interface to provide Surveillance Information to ATM User.~~ ~~Interface with other Surveillance Managers to exchange tracking information (optional).~~ ~~(No interface foreseen to serve MSEA information : there is no current analysis of what relevant information should be standardized)~~
~~Integration of Mode S elementary surveillance~~	~~Yes~~	
~~Integration of ADS-B~~	~~No (not any operational plan before 2005)~~	
~~Integration of ADS-C~~	~~No (not any operational plan before 2005)~~	
~~DAP Cache (as consequence of three previous items)~~	~~No (see previous items)~~	
~~Centralized Management of Clustered Mode S Stations~~	~~No (not a function mature enough to aim at interface specifications)~~	
~~ARTAS extended service capabilities (Tracking continuity, Service Continuity)~~	~~Optional (one Surveillance Manager may operate perfectly without implementation~~	
~~MSEA~~	~~Yes~~	
~~Airport Surveillance Function~~	~~No (Should be a different RFP)~~	
~~In fact, functional scope of Surveillance Manager could be summarised as the one of ARTAS V6 (see document)~~		

~~STEP B~~

~~**Interfaces**~~	~~**Assessment for IDLization**~~	~~**Interfaces selected for IDLization**~~
~~Interfaces with Classical radar and Mode S Stations.~~	~~No (whenever necessary, should be part of another RFP)~~	~~Interface to provide Surveillance information to ATM User.~~
~~Interface to provide Surveillance information to ATM User.~~	~~Yes~~	
~~Interface with other Surveillance Manager to exchange tracking information (optional).~~	~~No (function is optional and when available (i.e., ARTAS) no proof of use in current ATM environment)~~	

~~STEP C~~

~~Step C concerns the assessment of specifications (for selected interfaces) that are feasible within the full context of operation of the Ground ATM System; that is,~~

- ~~a Surveillance Manager implementing that solution could be provided,~~
- ~~those specifications could also be implemented in a short timeframe (technical and economical assumption) by the other Ground ATM components, clients of Surveillance Information,~~
- ~~the specifications allow for the operation of a resulting Ground ATM System with the required performances.~~

~~Concerning the first item, three different implementation solutions were foreseen. These solutions may be ranged from 1 to 3 depending on the level of effort necessary to implement them:~~

1. ~~The existing Surveillance Manager components, fulfilling the required functional performance (e.g., ARTAS), are not modified but a Gateway is added between Surveillance Manager and the "CORBA infrastructure" (the communication between the Surveillance Manager and the Gateway make use of the Surveillance Manager existing interface).~~
2. ~~The existing Surveillance Manager components are modified to interface directly with the "CORBA infrastructure." However, the IDL interfaces are specified so that the "application intelligence" existing both in current Surveillance Manager and Clients is preserved (*code is not changed*) as much as possible (this "intelligence" allowing performance of the component according to ASTERIX 252, 030. In ARTAS, this "intelligence" is the server function).~~
3. ~~The existing Surveillance Manager components are re-designed/re-written to take full advantage of the available CORBA facilities (e.g., the server function could be "rewritten" to rely on CORBA services such as filter, etc.).~~

~~It is supposed the first solution could only be proposed for prototype Systems assessing first feasibility of ATM Ground System on a CORBA architecture (not a robust –*from feasibility and cost point of view*– solution for future required extension of services).~~

~~In this specification, we consider an implementation as solution 1. System prototyping is the current status and this solution still offers the best compromise between performance and cost at that level.~~

~~If we consider the current service capabilities offered by a Surveillance Manager as ARTAS, we may distinguish between:~~

- ~~The broadcast Mode, where the service is defined at the level of Surveillance Manager and any user may "receive" that service but cannot modify it.~~
- ~~The Point to Point Mode, where the User subscribes to a "private service;" that is, has the capability to define and modify its services (transmission characteristics or condition of service, type of surveillance information, etc.).~~

~~In this specification, we only specify a service allowing for "broadcast mode" type of operation based on CORBA Cos Typed event service (CosTypedEvent). The service will support the full ASTERIX category 30 surveillance information.~~

~~This service will be sufficient to perform first evaluation of ATM Ground System architecture (nowadays, almost all Users still use broadcast services) and will allow for low cost implementation on the Surveillance Manager (i.e., Gateway) side and on the client side.~~

~~2.2 Roadmap for Specifications Updates~~

~~It is clear that these IDL specifications will have to be updated to consider all those aspects that were logically skipped in the current specification.~~

~~Although a precise Roadmap may not be given, it is possible to replace them in the general process described in this chapter and present a general scheme.~~

~~(Any event triggers a process ending with updates of specifications and therefore a new RFP. It is possible to consider several incoming events for the same version RFP). Figure 2-2 illustrates this process.~~

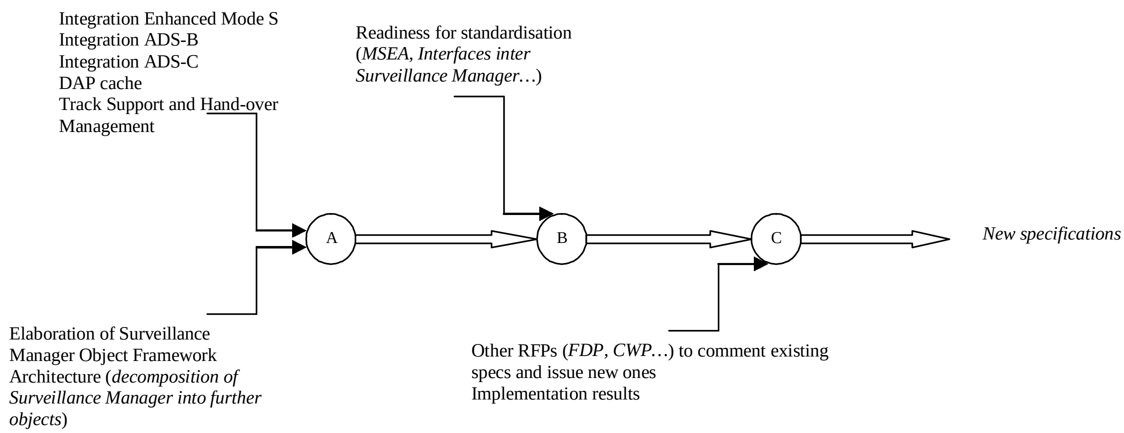
Integration Enhanced Mode S
Integration ADS-B
Integration ADS-C
DAP cache
Track Support and Hand-over
Management
Readiness for standardisation
(*MSEA, Interfaces inter*
Surveillance Manager…)
A
B
C
New specifications
Elaboration of Surveillance
Manager Object Framework
Architecture (*decomposition of*
Surveillance Manager into further
objects)
Other RFPs (*FDP, CWP…*) to comment existing
specs and issue new ones
Implementation results

Figure 2-2

Interface Specification 3

3.1 Introduction

The interface is based on AXTERIX category 030. This is related to the Exchange of Air Situation Pictures.

The Interface definition is a translation of AXTERIX syntax in IDL syntax. The data CATEGORY 030 : Exchange of Air Situation Pictures defines all items that can be transmitted by the Surveillance Server to its User(s) in the frame of any Track Information Service.

This interface shall be used with typed event COSservice. The Surveillance service is defined according to the service name defined in the COSnaming service.

According to PART I and PART II considerations a subset of Category 30 Data Item is used in the IDL definition.

Table 3-1 Data Items of Category 030 Used in the IDL Definition

Data Item Reference Number	Description
I030/050	ARTAS TRACK NUMBER
I030/060	TRACK MODE 3/A
I030/070	TIME OF LAST UPDATE
I030/080	ARTAS TRACK STATUS
I030/100	CALCULATED TRACK POSITION (CARTESIAN)
I030/130	CALCULATED TRACK ALTITUDE
I030/180	CALCULATED TRACK VELOCITY (POLAR)

Table 3-1 Data Items of Category 030 Used in the IDL Definition

I030/200	MODE OF FLIGHT
I030/220	CALCULATED RATE OF CLIMB/DESCENT
I030/240	CALCULATED RATE OF TURN

This subset is has been defined in the Avenue research program.

3.2 Description of Data Items

3.2.1 Description of Data Items of CATEGORY 030 and IDL Definition

3.2.1.1 I030/050 :ARTAS TRACK NUMBER

Definition: Identification of an ARTAS track.

```
typedef unsigned long Natural;
    const   Natural MIN_NATURAL = 0;
    const   Natural MAX_NATURAL = 2147483647;

typedef Natural TrackId;
    const   TrackId NULL_TRACK_ID = 0;
```

3.2.1.2 I030/060 :TRACK MODE 3/A

Definition: Mode 3/A identity associated to the track.

```
typedef octet SsrCode[4];

  struct RealModeA
  {
     boolean        is_validated;
     boolean        is_garbled;
     boolean        is_track_mode_changed;
     SsrCode        ssr_code;
  };
```

3.2.1.3 I030/070 :TIME OF LAST UPDATE

Definition: Absolute time stamping of the information provided in the track message, in the form of elapsed time since last midnight.

```
typedef SFloat ADuration;
  const     ADuration DAY_IN_SECONDS = 86400.0;
```

3.2.1.4 I030/080 :ARTAS TRACK STATUS

Definition: Status of an ARTAS track.

```
enum TargetType
  {
    TEST_TARGET,
    LIVE_TARGET,
    UNKNOWN_TARGET
  };

  enum TrackType
  {
    TENTATIVE_TRACK,
    CONFIRMED_TRACK,
    UNKNOWN_TRACK
  };


  enum RadarUpdate
  {
    PR_SSR_TRACK,
    PR_MULTITRACK,
    SSR_MULTITRACK,
    PR_SSR_MONOTRACK,
    SSR_MONOTRACK,
    PR_MONOTRACK,
    UNKNOWN_RADAR_UPDATE
  };

  enum SlantRangeCode
  {
    SLR_USING_MODEC,

    SLR_USING_CALCULATED_HEIGHT,


    SLR_USING_ASSUMED_HEIGHT,

    SLR_NOT_CORRECTED,
    UNKNOWN_SLANT_RANGE_CODE

  };

  enum SpecialCode
  {
    DEFAULT_SPECIAL_CODE,
    UNLAWFUL_INTERFERENCE,
    RADIOCOMMS_FAILURE,
    EMERGENCY,
    UNKNOWN_SPECIAL_CODE
```

```
};

struct TrackStatus
{
    TargetType          target_type;
    TrackType           track_type;
    boolean             uses_aircraft_derived_data;
    boolean             is_coasted;
    RadarUpdate         radar_update;
    boolean             is_terminated;
    boolean             is_created;
    SlantRangeCode      slant_range_code;
    SpecialCode         special_code;
    boolean             is_amalgamated;
    boolean             is_spi_set;
    boolean             is_military_emergency;
};
```

3.2.1.5 I030/100 : CALCULATED TRACK POSITION (CARTESIAN)

Definition: Calculated position of an aircraft expressed in Cartesian co-ordinates.

```
struct XY2DPosition {
    Miles     x_pos;
    Miles     y_pos;
};
```

3.2.1.6 I030/130 : CALCULATED TRACK ALTITUDE

Definition: Calculated altitude of an aircraft.

```
enum CalculationMode
  {
     THREED_HEIGHT,
     TRIANGULATED_HEIGTH,
     FROM_COVERAGE_HEIGTH,
     ASSUMED_HEIGHT,
     UNKNOWN_CALCILATION_MODE
  };

typedef long Integer;
const Integer MIN_INTEGER = -2147483648;
const Integer MAX_INTEGER = 2147483647;

typedef Integer      Feet;

  struct CalculatedTrackAltitude
  {
    CalculationMode    calculation_mode;
    Feet               track_altitude;
```

```
   };

~~typedef octet Empty;~~

  ~~union OptCalculatedTrackAltitude switch (boolean) {~~
  ~~case TRUE :~~
     CalculatedTrackAltitude calculated_track_alitude;
  ~~case FALSE :~~
~~     Empty field;~~
~~   };~~
```

3.2.1.7 I030/180 : CALCULATED TRACK VELOCITY (POLAR)

Definition: Calculated track velocity expressed in polar co-ordinates.

```
typedef float SFloat;
typedef SFloat  Knots;
   typedef SFloat  Azimuths;
   const Azimuths        MIN_AZIMUTHS = 0.0;
   const Azimuths        MAX_AZIMUTHS = 360.0;


   struct TrackVelocity
   {
     Knots       groundspeed;
     Azimuths    heading;
   };

   union OptTrackVelocity switch (boolean) {
   case TRUE :
      TrackVelocity track_velocity;
   case FALSE :
      Empty field;
   };
```

3.2.1.8 I030/200 : MODE OF FLIGHT

Definition: Calculated Mode-of-Flight of an aircraft.

```
struct Tendencies {
      VerticalTendency        vertical;
      HorizontalTendency      horizontal;
      SpeedTendency           speed;


~~   union OptTendencies switch (boolean) {~~
~~   case TRUE :~~
          Tendencies tendencies;
~~   case FALSE :~~
~~        Empty field;~~
```

- };

3.2.1.9 I030/220 :CALCULATED RATE OF CLIMB/DESCENT

Definition: Calculated rate of Climb/Descent of an aircraft.

~~union OptFeet switch (boolean) {~~
~~case TRUE :~~
Feet feet_val;
~~case FALSE :~~
~~Empty field;~~

~~union OptFeetPerMinute switch (boolean) {~~
~~case TRUE :~~
FeetPerMinute feet_mn_val;
~~case FALSE :~~
~~Empty field;~~

3.2.1.10 I030/240 :CALCULATED RATE OF TURN

Definition: Calculated Rate of Turn expressed in degrees per second.

typedef SFloat DegreesPerSecond;

~~union OptDegreesPerSecond switch (boolean) {~~
~~case TRUE :~~
DegreesPerSecond degrees_per_second;
~~case FALSE :~~
~~Empty field;~~
~~};~~

3.2.1.11 I030 Track

Definition: State Vector and list of radar track

struct StateVector
{ // Asterix Equivalent
ADuration last_update_time; // I030/070
TrackStatus track_status; // I030/080
unsigned short track_quality; // I030/090
XY2DPosition cartesian_position; // I030/100
~~Opt~~CalculatedTrackAltitude ~~opt_~~calculated_track_altitude;// I030/130
~~Opt~~Feet ~~opt_~~measured_mode_c; // I030/150
~~Opt~~TrackVelocity ~~opt_~~track_velocity; // I030/180
~~Opt~~Tendencies ~~opt_~~tendencies; // I030/200
~~Opt~~FeetPerMinute ~~opt_~~rate_of_climb_descent; // I030/220
~~Opt~~DegreesPerSecond ~~opt_~~rate_of_turn; // I030/240
};

```
struct RadarTrack
  {
    TrackId                     track_id;               // I030/050
    StateVector                 state_vector;
  };

  typedef sequence<RadarTrack>  RadarTracksList;
```

3.2.1.12 Surveillance Manager interface operation

Definition: Interface to be implemented by a typed push consumer.

```
interface SurveillanceManagerAsterix30
 {
  void TracksUpdate(in RadarTracksList  trackEvent);
 };
```

Interface Definition Source 4

```
module SurveillanceAxterix30
{

  //
  // I030/080
  //
  enum TargetType
  {
    TEST_TARGET,
    LIVE_TARGET,
    UNKNOWN_TARGET
  };

  enum TrackType
  {
    TENTATIVE_TRACK,
    CONFIRMED_TRACK,
    UNKNOWN_TRACK
  };

  enum RadarUpdate
  {
    PR_SSR_TRACK,
    PR_MULTITRACK,
    SSR_MULTITRACK,
    PR_SSR_MONOTRACK,
    SSR_MONOTRACK,
    PR_MONOTRACK,
    UNKNOWN_RADAR_UPDATE
  };

  enum SlantRangeCode
  {
```

```
  SLR_USING_MODEC,
  SLR_USING_CALCULATED_HEIGHT,
  SLR_USING_ASSUMED_HEIGHT,
  SLR_NOT_CORRECTED,
  UNKNOWN_SLANT_RANGE_CODE
};

enum SpecialCode
{
  DEFAULT_SPECIAL_CODE,
  UNLAWFUL_INTERFERENCE,
  RADIOCOMMS_FAILURE,
  EMERGENCY,
  UNKNOWN_SPECIAL_CODE
};

struct TrackStatus
{
  TargetType         target_type;
  TrackType          track_type;
  boolean            uses_aircraft_derived_data;
  boolean            is_coasted;
  RadarUpdate        radar_update;
  boolean            is_terminated;
  boolean            is_created;
  SlantRangeCode     slant_range_code;
  SpecialCode        special_code;
  boolean            is_amalgamated;
  boolean            is_spi_set;
  boolean            is_military_emergency;
};

//
//I030/100
//

typedef float      SFloat;
typedef SFloat     Miles;

struct XY2DPosition
{
  Miles  x_pos;
  Miles  y_pos;
};

//
// I030/130
//

enum CalculationMode
{
```

```
  THREED_HEIGHT,
  TRIANGULATED_HEIGTH,
  FROM_COVERAGE_HEIGTH,
  ASSUMED_HEIGHT,
  UNKNOWN_CALCILATION_MODE
};

typedef long  Integer;

const Integer  MIN_INTEGER = -2147483648;
const Integer  MAX_INTEGER = 2147483647;

typedef Integer  Feet;

struct CalculatedTrackAltitude
{
  CalculationMode      calculation_mode;
  Feet                 track_altitude;
};

                typedef octet  Empty;

union OptCalculatedTrackAltitude switch(boolean)
{
  case TRUE :
    CalculatedTrackAltitude  calculated_track_alitude;
  case FALSE :
    Empty  field;
};

//
// I030/150
//

union OptFeet switch(boolean)
{
  case TRUE :
    Feet  feet_val;
  case FALSE :
    Empty  field;
};

//
// I030/180
//

typedef SFloat  Knots;
typedef SFloat  Azimuths;

const Azimuths  MIN_AZIMUTHS = 0.0;
const Azimuths  MAX_AZIMUTHS = 360.0;
```

```
struct TrackVelocity
{
  Knots     groundspeed;
  Azimuths  heading;
};

~~union OptTrackVelocity switch(boolean)~~
~~{~~
~~  case TRUE :~~
    TrackVelocity  track_velocity;
~~  case FALSE :~~
~~    Empty field;~~
~~};~~

//
// I030/200
//

enum VerticalTendency
{
  CLIMB,
  STEADY,
  DESCENT
};

enum HorizontalTendency
{
  LEFT,
  STRAIGHT,
  RIGHT
};

enum SpeedTendency
{
  ACCELERATE,
  DECELERATE,
  MAINTAIN
};

struct Tendencies
{
  VerticalTendency        vertical;
  HorizontalTendency      horizontal;
  SpeedTendency           speed;
};

~~union OptTendencies switch(boolean)~~
~~{~~
~~  case TRUE :~~
    Tendencies  tendencies;
```

```
~~  case FALSE :~~
~~    Empty  field;~~
~~};~~

//
// I030/240
//

typedef SFloat  DegreesPerSecond;

~~union OptDegreesPerSecond switch(boolean)~~
~~{~~
~~  case TRUE :~~
    DegreesPerSecond  degrees_per_second;
~~  case FALSE :~~
~~    Empty  field;~~
~~};~~

typedef SFloat  ADuration;

const       ADuration  DAY_IN_SECONDS = 86400.0;

//
// I030/220
//

typedef SFloat  FeetPerMinute;

~~union OptFeetPerMinute switch(boolean)~~
~~{~~
~~  case TRUE :~~
    FeetPerMinute  feet_mn_val;
~~  case FALSE :~~
~~    Empty  field;~~
~~};~~

struct StateVector
{                                                          // Asterix Equivalent
  ADuration                   last_update_time;            // I030/070
  TrackStatus                 track_status;                // I030/080
  unsigned short              track_quality;               // I030/090
  XY2DPosition                cartesian_position;          // I030/100
  ~~Opt~~CalculatedTrackAltitude ~~opt_~~calculated_track_altitude;// I030/130
  ~~Opt~~Feet                    ~~opt_~~measured_mode_c;       // I030/150
  ~~Opt~~TrackVelocity           ~~opt_~~track_velocity;        // I030/180
  ~~Opt~~Tendencies              ~~opt_~~tendencies;            // I030/200
  ~~Opt~~FeetPerMinute           ~~opt_~~rate_of_climb_descent; // I030/220
  ~~Opt~~DegreesPerSecond        ~~opt_~~rate_of_turn;          // I030/240
};

typedef unsigned long  Natural;
```

```
const     Natural  MIN_NATURAL = 0;
const     Natural  MAX_NATURAL = 2147483647;

typedef Natural  TrackId;

const TrackId  NULL_TRACK_ID = 0;

typedef sequence<TrackId> TrackIdList;

struct RadarTrack
{
 TrackId      track_id;     // I030/050
 StateVector  state_vector;
};

typedef sequence<RadarTrack>  RadarTracksList;

interface SurveillanceManagerAsterix30
{
 void TracksUpdate(in RadarTracksList  trackEvent);
};
};
```

Acronyms and Abbreviations

Acronym/Abbreviation	**Description**
°	Degree (angle)
ARTAS	ATC Radar Tracker And Server
ASTERIX	All Purpose STructured Eurocontrol Radar Information EXchange
ATC	Air Traffic Control
AVENUE	*ATM Validation ENvironment for Use towards EATMS, TRANSPORT RESEARCH PROGRAMME, DG7 - TRANSPORT/AIR TASK N° 4.1.3/24A*
CAT	Data Category
EATCHIP	European Air Traffic Control Harmonisation and Integration Programme
EOP	End of Picture
EWPD	EATCHIP Work Programme Document
f	Scaling factor
FRN	Field Reference Number
FSPEC	Field Specification
FX	Field Extension Indicator
ICAO	International Civil Aviation Organization
LEN	Length Indicator
LSB	Least Significant Bit

NM	Nautical Mile, unit of distance (6 080 feet)
RDP	Radar Data Processing (system)
REP	Field Repetition Indicator
RFS	Random Field Sequencing
RSSP	Radar Systems Specialist Panel
s	second, unit of time
SAC	System Area Code
SIC	System Identification Code
SOP	Start Of Picture
SP	Special Purpose Indicator
SPF	Standard Precision Format
STFRDE	Surveillance Task Force on Radar Data Exchange
UAP	User Application Profile (see Definitions)
UTC	Coordinated Universal Time