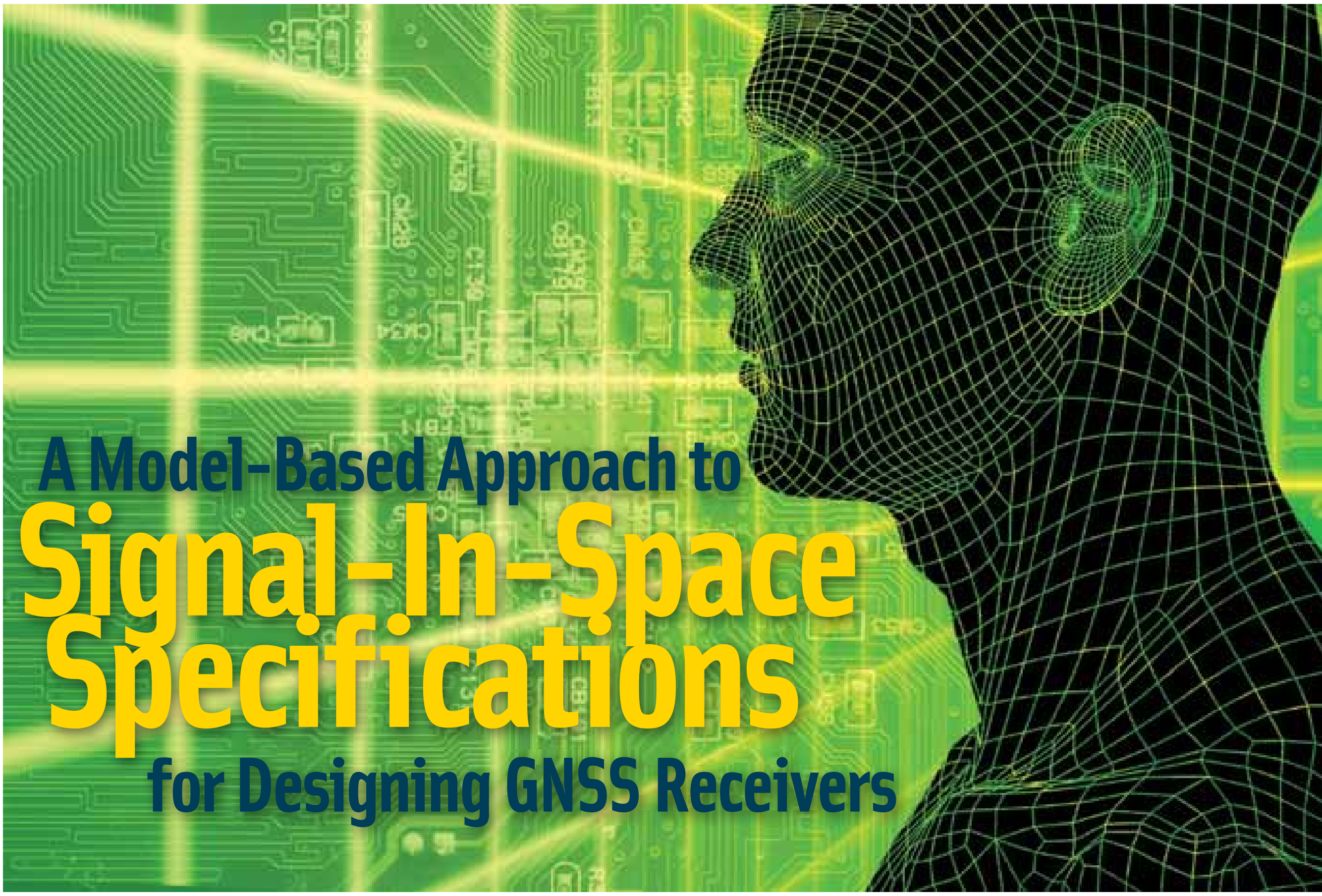

A Model-Based Approach to Signal-In-Space Specifications for Designing GNSS Receivers

The authors present a preliminary case study on the suitability of the Interface Communication Modelling Language (ICML) as a possible approach for the Galileo Open Service signal-in-space interface specification.

DANIELE GIANNI, JOACHIM FUCHS, PIERLUIGI DE SIMONE, NICKLAS LINDMAN AND MARCO LISI
EUROPEAN SPACE AGENCY/EUROPEAN SPACE RESEARCH AND TECHNOLOGY CENTER (ESTEC)

Galileo receiver designers require formal interface specifications for the Galileo signal-in-space (SIS) in order to write unambiguous and accurate specifications for Galileo receivers. To compute their positions, Galileo receivers must be able to retrieve timing and orbital information from the data stream conveyed in Galileo analog signals.

Next to the algorithms and the numerical issues, the correctness of the position solution also depends on the semantically correct interpretation of the data items in terms of their syntax and semantics. Multi-GNSS receivers are particularly challenged as these combine data from multiple and diverse formats in order to calculate the position solutions.

For example, an orbital eccentricity parameter can be represented in several binary and numerical formats. In order to reduce possible misunderstandings between Galileo designers and designers of other types of GNSS receivers, and also to minimize the risk of incorrectly computing a position solution, the Galileo interface specification must be clear, with all ambiguities and inconsistencies eliminated.

Misinterpretation of design specifications is not a new problem in the wider field of engineering. In the system engineering community, these problems are addressed by formalized approaches such as *Model-based System Engineering (MBSE)*, which require design documents to use formal and graphical languages.

The intent of an MBSE approach is to overcome the inherent ambiguities of natural language-based specification documents. MSBE approaches are used not only to support design specifications, but also to support all the phases of system design and life cycle, including intermediate verifications and final system validation.

In this article, we aim to overcome the limitations of the current GNSS SIS interface specification by proposing an MBSE approach for the Galileo SIS Interface Control Document (ICD), which is currently available in a textual format.

To overcome the limitations of the current GNSS SIS interface specifications, we propose use of t*he Interface Communication Modelling Language (ICML)*, a modelling language that enables GNSS designers to formally and graphically specify SIS interfaces, as an alternative to the conventional text used to prepare ICDs.

As a consequence of the increased level of formality, we expect to see an improvement in the design processes of GNSS-based systems, including enhanced communication among stakeholders, reduced design times, and reduced design risks.

In addition, *ICML* can also lead to the automatic generation of software conversion routines; specification consistency and completeness checking to ensure the correctness of the interface specifications and their consistency with lower-level design specifications; generation of designer friendly and interactive documentation in various formats, including web-based ones; and multi-GNSS interoperability on the receiver side.

An important note: Although no plans have been made to release the Galileo ICD using ICML, in this study we evaluate ICML features that could be particularly valuable for Galileo.

ICML is based on the standard and widely known Unified Modelling Language (UML) and Business Process (BP). This allows us to leverage existing UML and BP modelling tools, thereby gaining advantages from their wide availability and related standards. These system engineering tools also include Object Constraint Language (OCL) for specifying constraints on models, and SysML, system modelling language.

In this article, we first will outline the concepts of MBSE and UML and discuss the advantages that Galileo would obtain from a formal SIS specification in ICML. We then examine the structure of ICML specifications and provide an example *ICML* specification for a simplified and facsimile Galileo F/NAV message, which is the designated navigation message format for the Open Service on the E5a frequency.

Model-Based System Engineering (MBSE)

Traditionally, the design of a complex system relies on a system engineering process that uses on text documents and engineering data in multiple formats from different disciplines. This information is generally developed and shared electronically among all the relevant system stakeholders.

Much of the system engineering effort is spent to ensure that information is consistent across disciplines and maintained throughout the various versions of the document produced while

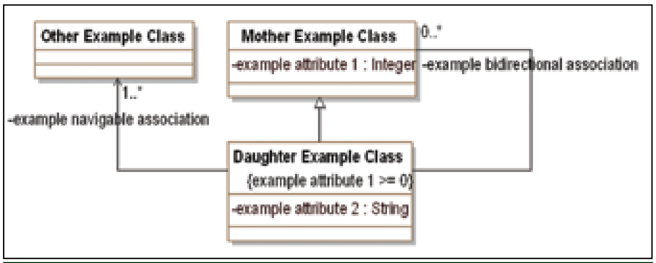

FIGURE 1 Example class diagram

advancing the system design. To assist in this, a *system engineering management plan* (SEMP) document specifies how the entire engineering process develops, including which documents need to be produced and what their inter-relationships are.

Due to its inherent nature, the document-based approach presents fundamental limitations, deriving from manual operation of support activities, dispersed data, and unstructured representation of information. Jointly, these factors affect the traceability of requirements across documents and throughout the design process — affecting design specification consistency and completeness, and thus representing a source of risk for development of the system.

All these problems are further exacerbated when the system under design involves integration of two or more systems, especially in cases where the systems are independently designed. In addition to increased engineering complexity due to the larger number of systems, the communication interfaces become a critical aspect of the entire process.

INCOSE, the International Council on System Engineering, defines MBSE as "the formalized application of modelling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle."

As such, MBSE aims to overcome the limitations of the conventional document-based approach by leveraging computing tools to structure, share and automatically analyse design information. The ultimate purpose is to ensure specification completeness and consistency, traceability of requirements and design choices, reuse of design patterns and specifications, and a shared understanding of the designs among users and designers.

As a result, the application of MBSE obtains several advantages, presented here with examples of the benefits:

- **Enhanced communications:** Enabling a shared understanding of the system across the development team and with other stakeholders, and the ability to integrate views of the system from multiple perspectives.
- **Reduced development risk:** Providing requirements validation and design verification throughout the process, as well as more accurate cost estimates to develop the system.
- **Improved quality:** Providing more complete, unambiguous and verifiable requirements; more rigorous traceability between requirements, design, analysis and testing; enhanced design integrity.
- **Increased productivity:** Analyzing the effects of requirements and design changes more quickly, reusing existing models to support design evolution, reducing errors and time spent on integration and testing, and automating document generation.
- **Enhanced knowledge transfer:** Standardizing specification and design information so that it can be accessed via query and retrieval software.

MBSE achieves these advantages by building a system model and model repository. A system model is represented digitally and can include information about the system's specification, design, analysis, and verification.

In order to reduce possible misunderstandings between Galileo designers and designers of other types of GNSS receivers, the Galileo interface specification must be clear, with all ambiguities and inconsistencies eliminated.

The model can be produced using a software tool qualitative metrics need to be satisfied by the model for the specific design study, such as model fidelity, model breadth, and model depth. A model repository provides a database of model blocks that can be shared among all the actors involved in the system design, across all the design and development phases.

The software industry has always been at the forefront of the modelling languages, information processing tools and technologies. As a natural consequence, MBSE currently takes advantage of the available modelling languages including UML, BP, and associated tools such as Magic Draw or System Architect along with technologies (e.g., XML), which originated in the software domain and were eventually tailored to the needs of system engineering.

Unified Modelling Language

In this section we outline the basic concept of UML upon which ICML is defined.

UML is a standard graphical modelling language for the representation of structural and behavioural aspects of systems, using the concept of *objects*. The language consists of a set of diagrams, each defining a set of palettes and a set of relationships that can be used to relate the palettes.

For example, a diagram for the representation of structural aspects is the class diagram. This type of diagram is used to describe classes, or types of objects, in terms of properties and relationships, for general conceptual and detailed modelling.

Figure 1 presents a sub-set of palettes for the class diagram. In this figure, each box, or *class palette*, represents an object type identified by name, such as Mother Example Class or Other Example Class, and associated attributes, (e.g., example attribute 1, of type integer, or example attribute 2, of type string).

By drawing a line between class palettes, we can define relationships of several types, for example, bidirectional and monodirectional navigable associations, or inheritance, illustrated by the straight arrow from Daughter Example Class to Mother Example Class.

Association relationships indicate that a class is associated with, aggregated from, or composed by another Class, with a numerical multiplicity (e.g., 0..* - zero or more, or 1..* - one or moce). *Inheritance relationships* indicate that a daughter class is a type of a mother Class, such that the daughter class inherits mother class's properties and further specializes the mother class, for example, defining a car as a type of motor vehicle.

In addition to the diagrams, UML also provides a profiling mechanism that enables modellers to specialize the semantics used by standard UML palettes to concepts of a specific domain. Examples of UML specializations are plentiful, from languages used for enterprise architectures, such as the UK Ministry of Defense Architectural Framework (MoDAF) or the European Space Agency Architectural Framework (ESAAF) to languages used in conventional system engineering, such as SysML.

Recently, SysML has been provided with a QUVD (Quantities, Units, Dimensions and Values), a UML specialization for physical quantities (e.g., Angle) and measurement units, (e.g., Radian, Adimensional Unit).

An example of UML class designation in the field of digital communications could describe bit sequence and bit sequence structure (i.e., sequence acronym and length). The semantics of bit sequence structure can be traced to particulae classes in a specialized UML model. Consequently, specialized UML diagrams convey more accurate information as they express narrower domain-specific semantics.

This profiling mechanism relies on the use of << and >> string tags. However, a UML profile can visually be represented using class diagrams while defining inheritance relationships between UML class and profile class specializations.

Later we will use this approach to present the Galileo interface specification using ICML, highlighting the classes belonging to the ICML definition. Because UML diagrams can be digitally stored using XML Metadata Interchange (XMI), an interchangeable XML-based format, UML diagrams can be easily shared among software tools and computer systems.

ICML Benefits for the Galileo SIS Interface Specification

The Galileo SIS ICD will be used by a large number and variety of receiver designers responsible for defining the specification of Galileo user equipment. New receiver designs may need to reuse and/or modify existing GNSS hardware and software design specifications to conform to Galileo ICD, or solve interoperability issues in existing receivers that use GNSSes.

Although the use of ICML might not solve such problems, it could considerably ease the design and implementation of the receiver specification needed to do so. Furthermore, by helping create an explicit and unambiguous Galileo ICD, ICML would make it easier for designers to identify differences between the Galileo ICD and those of other GNSSes. As a consequence, necessary receiver design modifications would be more easily determined, further easing the drafting of the receiver specification.

Similarly, interoperability issues concerning the representation of different data and data types should be also more clearly visible. As a result of all this, available and reconfigurable receiv-

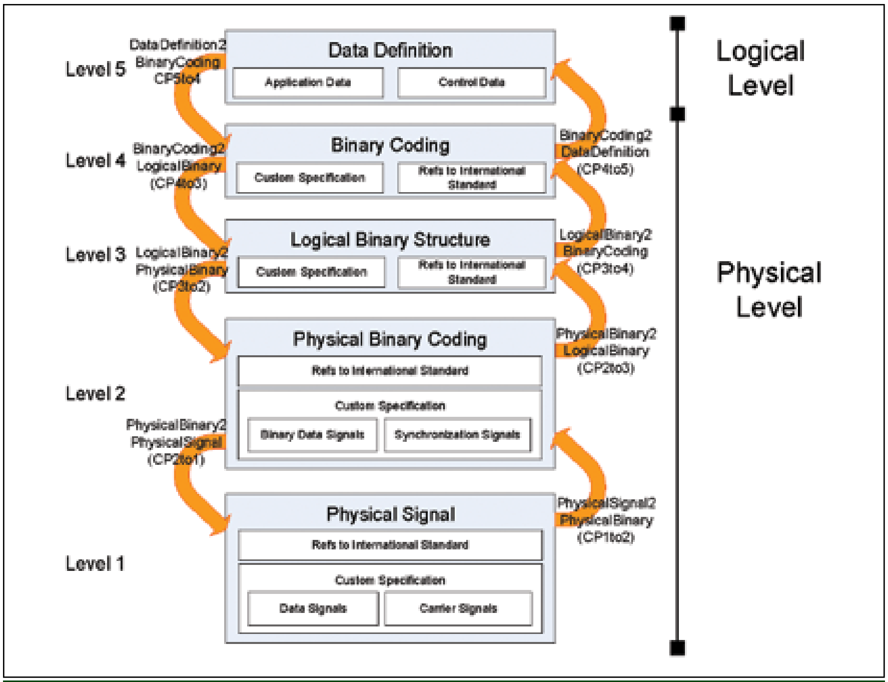

FIGURE 2 Structure of SIS interface specifications in ICML

ers could be reprogrammed basing on these differences.

More generally, ICML would promote the understanding of the Galileo ICD, easing the utilization of the Galileo signal and becoming a strategic advantage for establishing Galileo in the GNSS panorama.

ICML Specification

ICML covers both the structural and the implementation aspects of GNSS SIS interface specifications. The structural aspects concern the definition of the data structures. The implementation aspects concern how data values are dealt with. A graphical implementation of the full ICML specification is illustrated in Figure 2.

In **Figure 2**, the structural aspects are identified by the five light blue rectangles, each representing one of the five Galileo ICD specification levels: data definition, binary coding, logical binary structure, physical binary coding, and physical signal.

Data definition covers the specification of the logical structure of the message data. The structure is described in terms of data items, which are further characterized by associated semantic and pragmatic or definitions, that is, the contextual interpretation of the data. Details such as data item value constraints, measurement units, and physical quantities are also defined within this level. Data items can be application data, such as Galileo satellite orbital parameters, or control data, which support verification of the correctness and integrity of the application data.

The *binary coding* level covers the specification of the binary data item structures and coding. The binary sequence structures must be defined for each data item identified in the data definition level.

Logical binary structure specifies the aggregation of binary sequences in terms of frames, sub-frames, and pages. In particular, this level concerns two aspects of the specifications. The first is the sequence ordering foy digital modulation processes in the navigation message. The second is the association between sequences of application data and sequences of control data.

The *physical binary coding* level specifies the partitioning structure for binary sequence segments resulting from digital modulation. The structure is defined by a set of data blocks, which contain logical binary data from the upper level, and by block interleaves, which contain synchronization sequences. Digital synchronization and control signals transmitted by a satellite can also be defined at this level to support wider sets of specifications.

Finally, the *physical signal level* specifies the structure of analog signals. Physical properties such as phase, frequency, and signal shape, as well as synchronization and control of signals, are defined within this level. Additionally, the specification describes how the analog signal is defined by representing the mapping between signal properties and respective binary strings.

In Figure 2, the orange arrows represent implementation specifications, i.e., the *conversion processes (CPs)* for the message values between adjacent levels. For the five levels of ICML specifications, eight CPs can be specified using BPMN:

- DataDefinition2BinaryCoding (CP5to4), which defines the process for the derivation of the logical binary sequences representing data values.
- BinaryCoding2LogicalBinary (CP4to3), which defines the process for the aggregation of logical binary sequences in a binary message.
- LogicalBinary2PhysicalBinary (CP3to2), which defines the process for the derivation of physical binary sequence from logical binary sequences, i.e., the sequence of statements that a binary coding schema (e.g. convolution or encryption) implements.
- PhysicalBinary2PhysicalSignal (CP2to1), which defines the process for the generation of the analogue signal from physical binary sequences.
- PhysicalSignal2PhysicalBinary (CP1to2), which defines the process for the interpretation of the analogue signal in terms of a physical binary string.
- PhysicalBinary2LogicalBinary (CP2to3), which defines reciprocal process of LogicalBinary2PhysicalBinary, by specifying reverse convolution and decryption operations.
- LogicalBinary2BinaryCoding (CP3to4), which defines the process

- for the reconstruction of full binary sequences from the truncated logical binary sequence in the binary message.
- BinaryCoding2DataDefinition (CP4to5), which defines how the data values can be obtained from the logical binary sequences.

An important note: the ICML specification might not always need to include the complete definition of the foregoing processes if their content is trivial or can be unambiguously assumed. For example, when conversion processes conform to international standards or practices, such as Viterbi coding, the definition of these processes can be substituted by a textual note referencing the standard.

Example Specification for Galileo F/NAV

We will now present a simplified and facsimile Galileo SIS interface specification for the F/NAV message as an example of MBSE applied to GNSS ICDs. For the sake of conciseness, we will illustrate only the ICML specification for the logical binary data and physical binary data levels.

We assume that at the data definition level, the following data items are defined: OMEGADOT and Eccentricity e, as application data, and Page Type Field and CRC (cyclic redundancy check) as control data. The binary coding of these data items can be specified within ICML Level 4, which will also provide the basis for the Level 3 definition of logical binary representation. This representation concerns how Level 4 sequences are combined to form the binary message, including the binary control sequences for start, end, and synchronization.

The class diagram in **Figure 3** defines the message structure. In this diagram, the Reduced F/NAV Message Structure represents the *Logical Binary Message Structure* definition and consists of the F/NAV Start Sequence Structure, F/NAV Data Frame Structure, and F/NAV End Sequence Structure.

The F/NAV Start Sequence is formally named F/NAV Message Start Sequence and consists of three bits, 101. The F/NAV Data Frame Structure consists of F/NAV Sub-Frame 1 Structure and F/NAV Sub-Frame 2 Structure.

Sub-Frame 1 conveys the semantics "Satellite 1 and 2 orbital data (reduced set)" and must be interpreted as "original data copy." Sub-Frame 1 structure is defined in terms of F/NAV Page 1 structure and F/NAV Page 2 structure. For brevity, the description of Page 2 structure is omitted.

The F/NAV Page 1 structure is described in **Figure 4**. The structure semantics is inherently related to the semantics of the sequences composing the structure and, therefore, is not explicitly mentioned. Conversely, the structure pragmatics is not derivable from the inner elements and consequently must be defined by the pragmatics, *"The data must be interpreted as belonging to Satellite 1"*.

The Page 1 structure consists of structures for Type Field, Omegadot, e1 and CRC sequences. Each structure is associated to the respective sequence defined at Level 4, and to semantic and pragmatic descriptions. For example, Page 1 Type Field Sequence of Logical Bits Structure, represents the Level 4 Type Sequence of Bits Structure. Also, the Page 1 Type Field Sequence of Logical Bits Structure, must be interpreted as "concerning Page 1 Data" and as "Control Data on Page Application Data."

At the lower level, the ICML specification defines the physical binary structure, which concerns how sequences resulting from digital processing, such as convolution or encryption, are structured and combined with control and

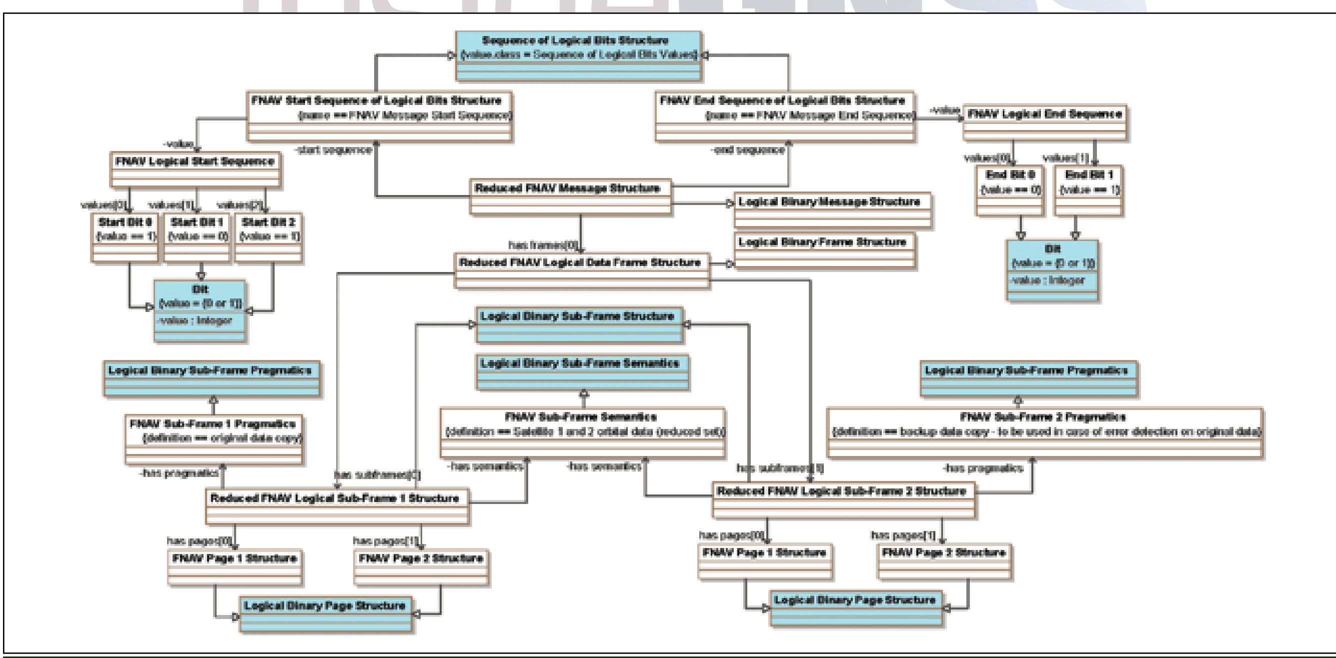

FIGURE 3 Reduced F/NAV Message Logical Binary Data Structure

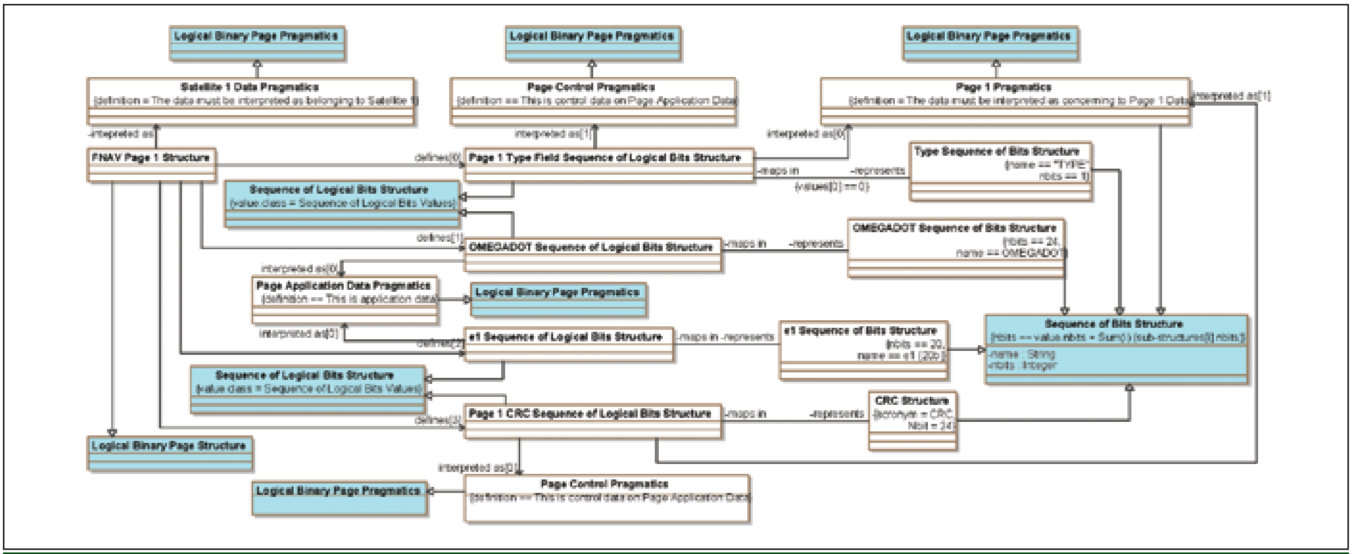

FIGURE 4 Reduced F/NAV Message Logical Binary Data Structure – Page 1 Type Definition.

synchronization sequences. The structure is defined by the class diagram in Figure 3. In this diagram, F/NAV Message Data Physical Structure represents the physical binary structure definition and consists of the F/NAV Start Sequence Structure, F/NAV Data Block 0, F/NAV Data Block 1 and F/NAV End Sequence Structure.

F/NAV Physical Start Sequence has the formameannotation, F/NAV Phys Start Seq, and is equivalent to the binary sequence 01. F/NAV Data Block 0 has a bit length of 70 and is formally identified by the name F/NAV Block 0. F/NAV Data Block 0 is limited by the binary sequence 0101, both at the beginning and end of the data sequence.

F/NAV Data Block 1 is similarly defined; however, its definition is not included in the diagram for the sake of brevity.

Finally, the end sequence is identified by F/NAV Physical End Sequence, which has formal name, F/NAV Phys End Seq, and is equivalent to the binary sequence 010.

Conclusions

Galileo will bring considerable advantages to navigation systems using GNSS, and to the Galileo civilian community, providing independence, increased accuracy, integrity and reliability. Promoting the use of Galileo signals means Galileo receiver designers must be able to accurately specify the Galileo receivers design, derived from the interface with the Galileo SIS.

Specifically, Galileo designers must eliminate Galileo ICD ambiguities and inconsistencies in order to reduce risk, thus making the design and development of Galileo receivers more cost- and time-efficient.

ICML brings a model-based system engineering approach into the specification of GNSS SIS interfaces, thus obtaining numerous advantages, including enhanced communication and reduced design risks. The approach supports a wide number of automatic exploitations (such as checking specification completeness and consistency, document generation, and so forth).

ICML may also be used to identify multi-GNSS interoperability issues on the receiver side involving the retrieval, interpretation, and combination of orbital and timing data from multiple GNSSes for the positioning computation.

Because ICML is based on the UML and BP standard modelling languages originated from the software community, ICML specifications can be created using the plethora of available software tools.

A final caveat: this article represents a preliminary case of study on the suitability of such an approach for the Galileo SIS interface specification and no endorsement is made on the adoption of ICML for Galileo interface.

Acknowledgements

The authors would like to express their sincere appreciation to Serge Valera of the EGSE and Ground Systems Section, European Space Agency, for his insightful comments.

References

[1] Eriksson, H. E., and M. Penker, B. Lyons, and D. Fado *UML 2 Toolkit,* Wiley, 2008

[2] *Galileo OS SIS ICD Issue 1 Revision 1,* September 2010, retrieved from <http://ec.europa.eu/enterprise/policies/satnav/galileo/open-service/index_en.htm>

[3] Gianni D., and J. Lewis-Bowen, N. Lindman, and J. Fuchs, "Modelling Methodologies in Support of Complex Systems of Systems Design and Integration: Example Applications", *Proceedings of the 4th International Workshop on System & Concurrent Engineering for Space Applications (SECESA2010),* Lausanne, Switzerland, October, 2010

[4] Kossiakoff, A., and W. N. Sweet, *Systems Engineering Principles and Practice,* John Wiley, 2003.

[5] MoDAF Detailed Guidance, <http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/ModafDetailedGuidance.htm> (last accessed October 2010).

[6] "Model-Based Systems Engineering," *The Journal of the National Council on Systems Engineering (INCOSE),* Volume 1, Number 1, 1994, pages 83-92

[7] Object Management Group (OMG) Business

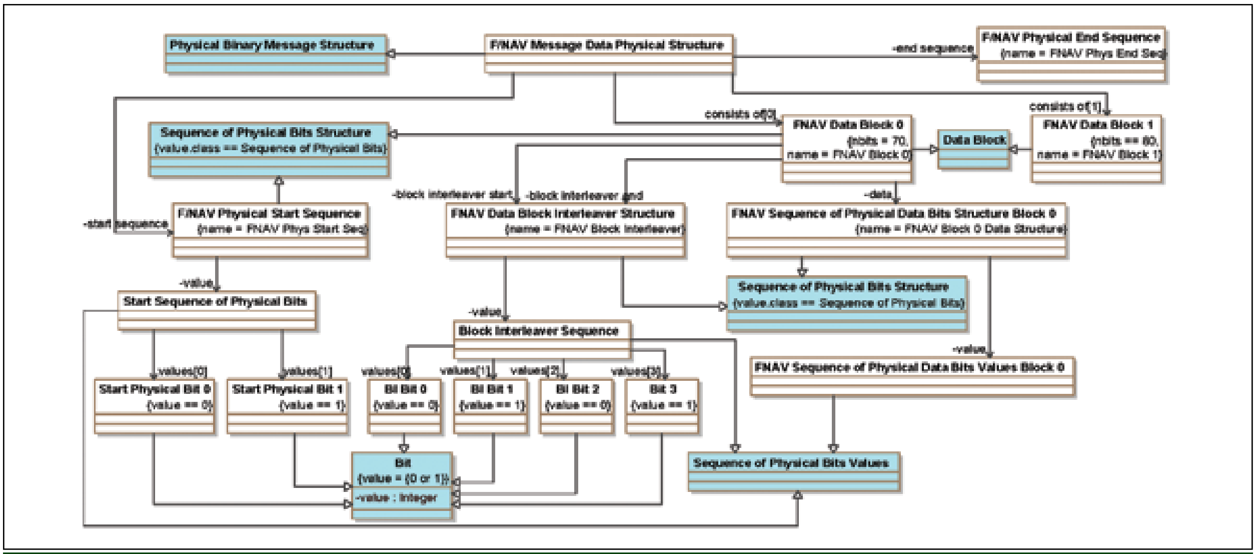

FIGURE 5 Reduced F/NAV message physical binary coding

Process Model Notation Standard Specification, <http://www.omg.org/spec/BPMN>

[8] OMG Object Constraint Language Specification, <http://www.omg.org/technology/documents /formal/ocl.htm>

[9] OMG SysML Annex C, Model Library for Quantities, Units, Dimensions and Values (QUDV) < www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf>

[10] OMG SysML Specification, <http://www.sysml.org>

[11] OMG Unified Modelling Language Standard Specification, <http://www.uml.org>

[12] OMG XML Metadata Interchange (XMI) Standard Specification, <http://www.omg.org/spec/XMI>

[13] W3C Extensible Markup Language Specification, <http://www.w3.org/XML>

Authors

Daniele Gianni is currently an internal research fellow in the System Modelling and Functional Verification Section of the European Space Agency. Gianni received a Ph.D. in computer and control engineering from University of Rome TorVergata (Italy), in the field of modelling and simulation. He held research appointments at Imperial College and the University of Oxford in the United Kingdom and received research scholarships for the Georgia Institute of Technology (United States) and Auckland Bioengineering Institute (New Zealand). You may contact him by e-mail at <Daniele.Gianni@esa.int>.

Joachim Fuchs is head of the System Modelling and Functional Verification Section at the European Space Agency. His background is in software and space system engineering. His group supports ESA projects in the area of assembly, integration and verification (AIV)/assembly, integration, and testing (AIT) and the corresponding verification means, but also in the adaptation of new engineering methods such as model-based approaches.

Pierluigi De Simone INCOSE, IPMA, is currently working on system assembly, integration and verification (AIV) for the Galileo mission in ESA. He has worked in many software developments in the field of graphics, safe mode software, and visual programming, and on many space missions, including Helios, METEOSAT, METOP, and COSMO-SkyMed, . His main interests are in modelling paradigms and cryptography. De Simone has worked at Finmeccanica Group, ECMWF, EUMETSAT and holds a master's degree in physics from the Rome University, Italy. (pierluigi.desimone@incose.org)

Nicklas Lindman is a system engineer working at the European Space Agency in the System Modelling and Functional Verification Section at ESTEC in Noordwijk, The Netherlands. At ESA he supports projects and leads R&D activities in the area of system engineering, system modelling, and system simulation. He also supports standardization activities in these areas. He holds a master's degree in space engineering from Umea University in Sweden.

Marco Lisi is currently Galileo system procurement manager at the European Space Agency, Directorate of GALILEO Programme and Navigation related activities, holding responsibilities for system engineering, operations and security activities in the Galileo project. He has more than 30 years of work experience in the aerospace and telecommunication sectors, holding management positions in R&D, engineering, and programs, and being directly involved in a number of major satellite programs, including Artemis, Meteosat Operational, Meteosat Second Generation, Globalstar, Cosmo-Skymed, and, more recently, Galileo. Lisi is on the board of directors of the Italian Institute of Navigation (IIN) and "Honorary LifePresident" of the Italian chapter of INCOSE (International Council on Systems Engineering), which he founded in 2008. He holds three international patents, in EC and USA, and has authored more than 150 papers. IG