This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.20xx.DOI

Reusability, Reconfigurability and Efficiency Optimization of Satellite Network Modeling and Simulation

SU GAO[1], YUE CAO[2], YINQIAO LI[1], YUJUN CHEN[1],SONG JIN[1],SHIKUN XIE[3] and JIHONG LIU[4]
[1]Institute of Telecommunication and Navigation Satellites, China Academy of Space Technology (CAST), Beijing, China
[2]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
[3]Beijing iFlight Technologies co., LTD, Beijing, China
[4]School of Mechanical Engineering and Automation, Beihang University, Beijing 100191,China

Corresponding author: Jihong Liu (e-mail: ryukeiko@buaa.edu.cn).

ABSTRACT

Model-based approach with reusable mechanisms can serve as an effective way for complex system architecture design. Stakeholder needs should be satisfied while product and architecture design need to be consistent with user requirements in all stages during the whole product lifecycle. In this paper, satellite network as an example of complex system is modeled in a reusable, reconfigurable and efficient manner using the system modeling language (SysML) together with pattern viewpoints and simulation constructs. Based upon abstract syntax described using metamodels and a set of profiles, concept reusability is established for the specific domain. Additionally a reusable modeling framework is developed with tailored design patterns and multiple viewpoints. Analysis metamodel, profile and interface are further presented to preserve reusability during iterations among multiple optimization rounds. A novel satellite network simulation model is formulated and multi-objective optimization is solved by transformation under practical application scenarios. A set of metrics are designed to assess and validate the models. Results show that the proposed reusable model has viewpoint coverage of more than 80 percent compared to a half for the baseline OOSEM model. The proposed model thus covers the pattern viewpoints and ontologies in a wider and more frequent way and is more efficient. Design choices made based on the model can be incorporated into this mechanism which is extensible along the system lifespan.

INDEX TERMS Reusability, satellite network, domain specific model, pattern viewpoint, simulation, multi-objective optimization

I. INTRODUCTION

SATELLITE and system architecture design should satisfy requirements derived from stakeholder needs during the whole spacecraft lifecycle. During each stage, specific modeling and simulation tools are applied to solve design problems in every discipline. Although understandable, it is inefficient to employ either a traditional text-based system engineering approach or simply piling-up of models, which will be very difficult to maintain when more system details are needed or requirements are modified. In this context, a holistic approach to support reusability of model transitions over phases is needed to maintain consistency of the satellite network modeling and simulation framework.

Reusable models should allow extensions and tailoring of the model when new needs arise. The extension should be based upon an inherent set of concepts defined with domain ontology. Metamodeling of the system modeling language (SysML) [1] provides concept reusability mechanism by allowing domain specific extensions and parameter reconfiguration to the language. Abstract syntax can be implemented based on Ecore, the Eclipse-specific implementation of the Meta Object Facility (MOF) [2].

Model developed based on the reusable concepts can follow certain methodology, such as the INCOSE Object-Oriented Systems Engineering Method (OOSEM) [1]. It incorporates object-oriented methods of software into system engineering making system architecting more extensible. A modeling and architecture framework using the OOSEM for

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

satellite communication network can be found in [3]. To construct reusable and reconfigurable models that can be applied at various stages of the problem and to other similar type of problems, the idea of pattern has been introduced [4]. It states that a solution that has been useful in one context is probably useful in others. Developed based on reusable concepts in metamodels, these reusable models should be designed within a unified framework to maintain consistency during various stages of the satellite lifecycle [5].

Simulation is carried out for most of the time to assess architecture design indeed satisfy the requirements. Reusable mechanism should thus be extended from within the modeling software to the set of multi-disciplinary simulation software and hardware. Efforts have been made to generate simulation code for different simulation language or environments, such as for Arena [6], Modelica [7] and discrete event system specification (DEVS) [8]. Additionally, simulation should evaluate variants of design variables to provide alternatives for architecture decisions. This can be realized by multi-objective optimization with Pareto front analysis [9]. The interaction between system architects and domain experts [10] should thus be extended to return the multiple architecting points to the modeling software.

Considering multi-disciplinary simulation in the satellite network domain, [11] has set up a multi-objective optimization incorporating communication capacity and satellite energy consumption. But it considered only geostationary earth orbit (GEO) satellite which has simpler application scenario than low earth orbit (LEO) satellite networks. By taking the frequently-changed topology of LEO satellites into account, [12] investigated energy consumption as constraint to the maximization of system capacity in a mission-aware topology design. But the model was not reusable for further detailed implementation and complicated multi-step optimization limited its use in practical scenarios. Energy-efficient routing for LEO network was considered in [13] with power allocation to various functions onboard the satellite investigated. But it did not take the satellite-to-ground transmission link into account, which consumes the most energy in satellite payload.

The main contributions of this paper include:

- Domain specific language is investigated and abstract syntax is described using metamodel for system requirement, function and architecture. A set of profiles, SysML4SatNetSim, is developed based on the metamodels with stereotypes utilized to add new language concepts. Reusable concepts in satellite network domain is thus formulated which can be used during the whole system and product lifecycle.
- SysML model following the OOSEM procedure [3] is reinvestigated by building up a reusable modeling framework to take various user concerns into account. Design patterns for system engineering and the process modeling framework [4] are modified to establish a simulation framework. Tailored reusable patterns are developed with multiple viewpoints. In this way, SysML models will have a rigorous structure which speeds up the modeling cycle by distributing relevant patterns, rather than a random collection of diagrams.
- Analysis metamodel and profile are developed as the basis of simulation reusability, including constructs before, during and after simulation. In addition, multiple architecture points and objective values (rather than a single value for each property) are made available in SysML, to make architecture point selection and trade-off analysis to be carried out in the modeling software. Reusable mechanism is preserved for iteratively simulating more complex system characteristics at subsequent development stages.
- A novel satellite network simulation model is presented for the integration with multi-level SysML models. The extended time-evolving graph (ETEG) [12] is extended to model satellite communication network. Different layers based on communication protocol levels are separated for identifying resource utilization. With practical application scenarios for satellite network, a multi-objective optimization problem is formulated and solved to provide Pareto set back to the SysML software. Consumed energy, considering both user downlink [14] and inter-satellite link [13], reflects comprehensive energy consumption and assists architecture value updates in the SysML model.
- The goodness of models developed are evaluated with reasonable criteria. Model assessment is presented with quantitative methods. A set of metrics are developed extending the assessment in chapter 20 of [4] to reflect viewpoint and ontology coverage compared to standard patterns. Results show that our proposed reusable models cover the model elements in a wider and more frequent way than the non-reusable models. Additionally, the degree of requirement traceability is quantified from SysML matrices which can be used as an effective evaluation for large and complicated systems.

This paper is organized as follows. After a review of related works in Section II, Section III gives an overview of the whole reusable framework. Section IV, V, VI sequentially present SysML-based modeling and architecture design approaches for concept reusability, model reusability and simulation reusability, respectively. Multi-objective optimization problem based on ETEG satellite network is established in Section VII together with its connections with the SysML models. A case study with practical application scenarios is modeled and simulated in Section VIII where results are compared with other works. The model assessment and validation is presented with optimized efficiency calculation in Section IX. The paper is concluded in Section X.

II. RELATED WORKS

The development of a complex system such as satellite network requires the participation of a variety of engineering disciplines and technical domains. To achieve common understanding among a diverse set of domain specific terminology and multiple modeling viewpoints, reference architecture which can provide common model for all disciplines should be applied as the system design evolves from requirement analysis, use case and conceptual design to system functionality realization through software and hardware. As an early-stage work, the Software Communications Architecture (SCA) [15] [16] revolutionized the modeling approach of a communication system and has been proven to be a useful infrastructure. But it mainly considered software specifications and did not provide insights into hardware. Another study [17] was later initiated to develop an architectural specification for satellite communication software and systems using a model-based approach. The Integrated Communication System Model (ICSM) [18] developed a baseline and high-level modeling approach for communication systems. It used a modeling tool to describe the function, behavior, architecture and other elements in a single paradigm.

To realize unified modeling in one single framework, multiple object-oriented (OO) modeling languages and design tools were brought out which coalesced to the Unified Modeling Language (UML) standard [19]. The limitations of applying UML to system engineering is the difficulty to realize system and software within a single modeling paradigm. This brings the extension of UML to the system modeling language (SysML) [20] for system modeling. Multiple design perspectives can thus be reflected based on a single meta-model. System decomposition, use cases and functional blocks of a communication system model using SysML was developed in [21].

It has been presented that reconfigurable SysML models are required to reduce expansion cost and cycle time [22]. In this framework, SysML metamodels and profiles are combined and applied to integrate multiple viewpoints and maintain consistency. The mechanism developed realizes faster design cycles and easier design information maintenance. In addition, separation of concerns in the form of multiple viewpoints is regarded as a possible way to tackle the complexity of a problem [5]. Domain specific languages are developed to facilitate particular sets of views. In general, viewpoints are defined based on the problem concerned and they establish the conventions to construct and analyze corresponding views. Various types of modeling framework based on viewpoints have been developed, such as the Zachman's framework [23] and the reference model for open distributed processing in [24]. Viewpoints can also be specified through projective approach based on principle of orthogonal viewpoints [25].

Model-based approach should be incorporated with simulation for quantitative architecting decisions. Complex system modeling with multiple stakeholders uses *ad-hoc* methods which have no standard structures to follow can result in various forms of system presentation and reduce modeling efficiency. The work in [6] developed a SysML mechanism for model driven simulation based on profiles and create model libraries using the language extensions. A number of works have been done to connect SysML constructs to various simulation software [26]. The research in [8] provided the ability to generate simulation code for typical discrete-event simulation software. Collaborative modeling and simulation interface has been raised in [27] which use SysML as the front-end for orchestrating MBSE at early system development stage. Allowing a decision process with more degree of freedom, Pareto front analysis for objective tradeoffs have been employed based on SysML analysis models in [9]. Integration of mechanical domain specific simulation models into descriptive SysML models was done in [10] to close the gap between system architects and domain model creators.

The works of connecting SysML modeling and simulation mentioned above have not considered multi-level design and optimization parameters. In satellite communication network domain, system architecture design with cross-level parameters, such as capacity in network level and energy consumption in sub-system level, is frequently encountered. Reusable SysML models should be developed with tailored design-simulation interface for supporting simulation execution. Multi-level and multi-objective modeling and optimization of a satellite system was investigated in [11], where power allocation (satellite platform level) and system capacity (network level) objectives are formulated together and solved with a two stage Pareto optimization. Antenna slewing time [28] and onboard processing capability (component level) are considered with an ETEG network-level model [30] to optimize system capacity. Power distribution based on satellite platform avionics are modeled together with network-level routing in [13] to maximize satellite battery lifecycle. In addition, mission-aware network performance has been analyzed with satellite resource utilization (subsystem level) [12] [14]. However, the works mentioned above lack a viewpoint of satellite design optimization based on application scenarios with reusable mechanism throughout the satellite production lifecycle. This viewpoint aims at improving efficiency and reconfigurability in satellite system manufacturing, which will be investigated in this paper.

III. METHOD OVERVIEW

OOSEM-based satellite network modelling and simulation [3] provides a complete system engineering procedure for satellite system architecture optimization. The elements developed during this procedure can be divided into requirements, functions, architectures and analysis. This is illustrated in Fig. 1 as the lower left block titled with “OOSEM Model”.

To improve efficiency and reconfigurability of the model development, concept reusability should be considered first. Satellite network domain specific language should be investigated. To enable reusable SysML extension to support satellite network domain specific concepts, metamodeling as shown at the top left of Fig. 1 should be used. There are four steps in a typical metamodel [1]:

- *Domain language formulation* identifies key concepts, relationships and constraints based on targeting application scenarios of the specific domain.

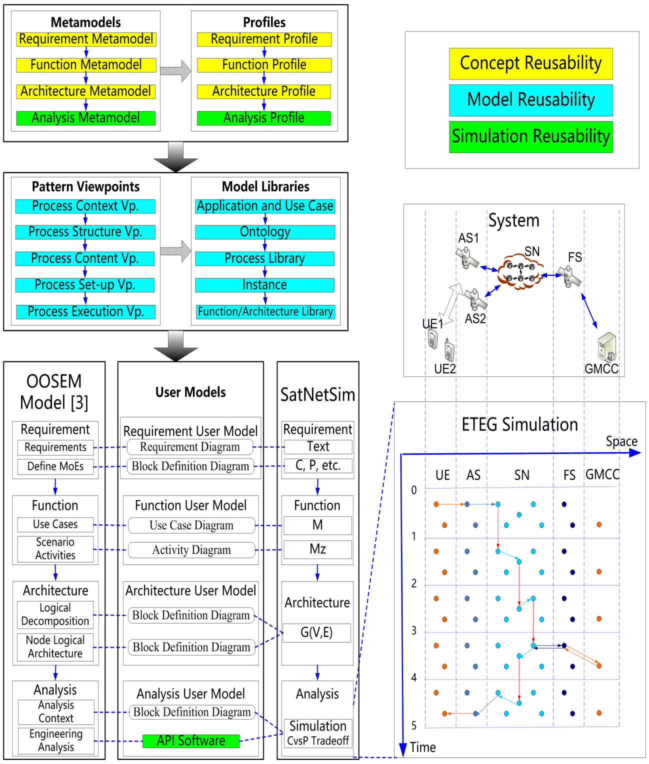


FIGURE 1: Method Overview.

- *Abstract syntax definition* describes the concepts above with a set of well-formedness rules explaining how the concepts can be put together. The abstract syntax is described using a *metamodel*. The Meta Object Facility (MOF) version 2.0 of the OMG standard is used to define metamodels.
- *Concrete syntax definition* maps the abstract syntax concepts to graphical symbols on diagrams, which visualizes the concepts and makes the concepts usable during the modeling procedure.
- *Semantics specification* gives the meaning of the language concepts by mapping them to concepts in the domain. It is mostly described using natural language.

It can be seen that domain language formulation is the foundation of the metamodeling procedure, which determines elements in the metamodel. The other three steps provide descriptions for these elements. The model elements obtained can thus be used for system modeling.

The next step is developing profiles in the specific application domain of satellite network. *Stereotypes* are utilized to add new language concepts, which are grouped together in special packages called *profiles*. This is shown at the top of Fig. 1 as the “Profiles” block next to the “Metamodels” block. A stereotype is extended based on one or more metaclasses in a reference metamodel. To choose suitable metaclasses for a stereotype, a language designer examines characteristics of the new concepts and looks for metaclasses which best match the concepts.

Reusability in the domain specific language definition stage is revealed here. As a powerful reuse mechanism, stereotypes extended based on an abstract metaclass is equivalent to those extending all the concrete metaclass specializations. In this work, metaclasses, either intrinsic or defined by domain language designer, closest to the domain concepts are chosen to create stereotypes. Based on this, a collection of stereotypes in the given satellite network simulation domain forms a profile under the name SysML4SatNetSim.

A second-stage reusable mechanism is to develop a model framework using elements from metamodels and profiles in the reusable concepts. As illustrated in Fig. 2, system is abstracted by user model which are formed with a number of views. Each view is manifested in one or more SysML diagrams. The user model, however, should not just be a random collection of diagrams, but with a rigorous structure. The structure of the model is defined by the model framework. The views in the user model should conform to viewpoints which are parts of the model framework.

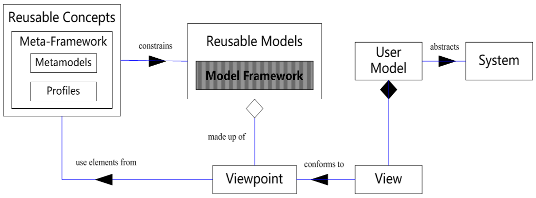


FIGURE 2: Model Reusability and Other Elements.

The model framework provides the basis of the model by identifying and defining a number of viewpoints representing model structure and contents. These viewpoints can be seen as templates of the views. During the modeling procedure, system engineers discover that certain types of problems occur over and over again, such as interface definition or process behavior. Then the core of the solution to those problems can be described in a way that the solution can be used many times over. Therefore, the idea of a pattern [4] has been adopted in the model framework. A pattern has a number of viewpoints and one or more patterns are used in a framework with intended purpose and application. Pattern viewpoints and model libraries developed based on them forms the model reusability mechanism as shown in the middle blocks at the left of Fig. 1.

The third-stage, simulation reusability, is facilitated by analysis modeling in SysML and its interaction with multi-disciplinary simulation tools, shown as the green blocks in Fig. 1. When simulation is required to optimize system architecture in the model, analysis metamodels and profiles should firstly be developed to define data exchanged before, during and after simulation. Analysis modeling also makes use of requirement, function and architecture models in pattern viewpoints to establish relations between system design and simulation parameters such as objectives, constraints and design variables. These parameters, together with optimization model presented in SysML, are delivered to multi-disciplinary tools for simulation. Multi-objective optimization results obtained in these tools such as Pareto set and relevant design variable values are returned to the modeling tool. These results are further analyzed in SysML to get suggested architecture point to update model values. Analysis models and interface between modeling and simulation tools should be reusable in that, when more parameters are added to reflect more complex system characteristics, system architecting relying on model-simulation interface can easily be extended without setting up the interface again.

IV. CONCEPT REUSABILITY: DOMAIN SPECIFIC LANGUAGE DEFINITION

A. METAMODELING

Domain specific metamodel in this paper is developed in EcoreTools following the MOF 2.0 rules. Satellite network metamodel consists of requirement, function, architecture and analysis metamodels. The first three are presented as follows while the analysis metamodel will be discussed in later sections.

1) Requirement Metamodel

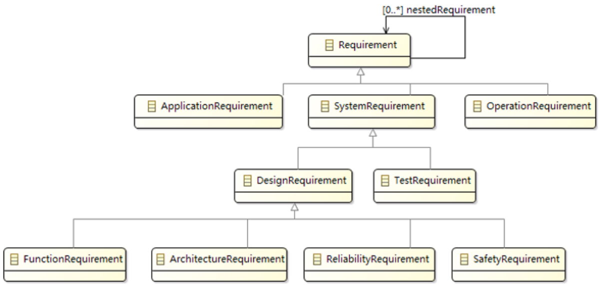
Requirement metamodel describes methods to model system requirements as shown in Fig. 3 (a). Requirement can be divided into application requirement, system requirement and operation requirement. System requirement can further be divided into design requirement and test requirement where the former can include function, architecture, reliability and safety requirements. The development of a certain requirement type depends on the problem to be solved at hand. Architecture design and optimization is the purpose of this research so design requirement is divided into its sub-requirements. Others can be specified on the metamodel already developed when other tasks are raised, which makes the DSL reusable. Specifications of the requirements are listed in Tab. 1.

Relationships among requirements are shown in Fig. 3 (b). One requirement can include a few sub-requirements. System requirement and operation requirement can be derived from application requirement.

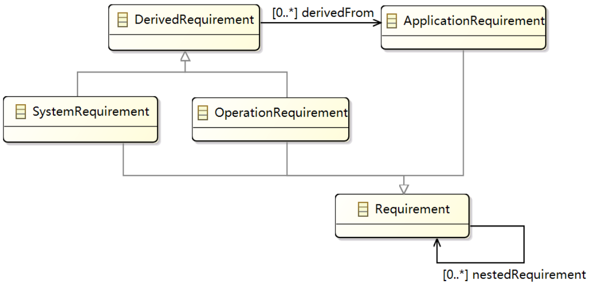
2) Function Metamodel

Function metamodel describes system function modeling mechanism. As seen in Fig. 4, system function can be divided into basic function, auxiliary function, specific function and evolving function. Function specifications are listed in Tab. 2.

In the following discussion, functions are realized as activities in SysML. Flow is used to deliver message between function activities. Flow information represents the information exchanged in the flows among functions. Flow information contains energy flow, material flow and signal flow. Signal flow can further be divided into wired and wireless flows. Flow information specifications are listed in Tab. 3.



(a) Concept



(b) Relationship

FIGURE 3: Requirement Metamodel.

TABLE 1: Requirement Specifications.

MOF Metaclass	**Specifications**
Application Requirement	Describes functions or measurement of effectiveness which satellite network system users are concerned based on application scenarios. For example, system capacity gained or time delay spent when satellite communication or broadcasting application scenarios happen.
System Requirement	Requirements to be satisfied when system builders are designing and manufacturing the system.
Operation Requirement	Requirements to be satisfied when system operators are using and maintaining the system after the system finishes manufacturing.
Design Requirement	Requirements to be satisfied when system designers are designing the system.
Test Requirement	Requirements to be satisfied when system is at the test stage.
Function Requirement	Requirements for the system to have certain functions.
Architecture Requirement	Requirements for the system to comply with certain architecture constraints.
Reliability Requirement	Requirements for the system to perform tasks within certain time period and conditions.
Safety Requirement	Requirements for the system to avoid harmful effects to the product, human health and environment.

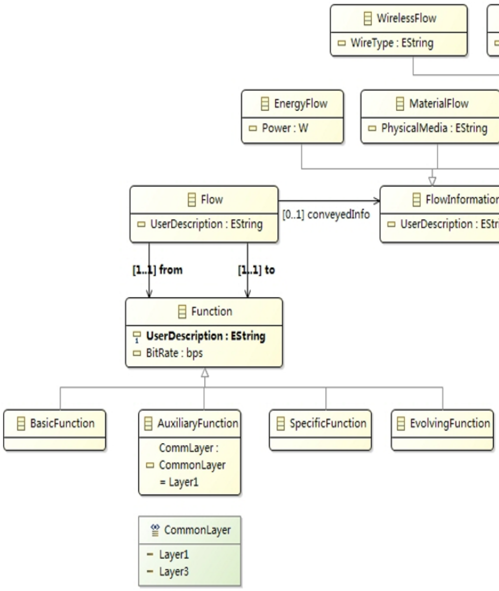


FIGURE 4: Function Metamodel.

TABLE 2: Function Specifications.

MOF Metaclass	Specifications
Basic Function	Common functional aspects which all communication systems should have.
Auxiliary Function	Data transmission and processing functions following the OSI seven-layer protocol.
Specific Function	Special data transmission types besides bi-directional communication. For satellite communication it can be broadcasting from ground management and control center (GMCC) through satellite network to user equipments (UEs), or data collection from UEs through satellite network to GMCC.
Evolving Function	Other functions in addition to those mentioned above. This serves as an extension mechanism for technology advancement in the future.

TABLE 3: Flow Information Specifications.

MOF Metaclass	Specifications
Energy Flow	Energy exchanged between functions, e.g. heat or electricity. Attribute Power represents the power that energy flow consumes.
Material Flow	Matter exchanged between functions. Attribute PhysicalMedia represents the physical media that material flow transmits in.
Signal Flow	Electronic signal exchanged between functions. Attribute BitRate represents the transmission rate of the signal.
Wireless Flow	Wireless signal flow exchanged between functions. Attribute TransmissionFreq represents the communication frequency used by the transmission.
Wired Flow	Wireless signal flow exchanged between functions. Attriibute WireType represents the type of wire through which signal is transmitted, e.g., optical fiber or copper wire.

3) Architecture Metamodel

Architecture metamodel gives system architecture modeling methods. Satellite network system architecture is formed by two types of elements, segment and component as shown in Fig. 5. Segment of satellite network contains user segment, space segment and ground segment. Component is formed by antenna, communication processor and channel. As the lower three layers in the OSI seven-layer protocol is considered in this paper, Communication processor is further divided into L1RadioFreq, L1Baseband, L2DataLink and L3Network. Tab. 4 gives the specifications of architecture MOF metaclasses.

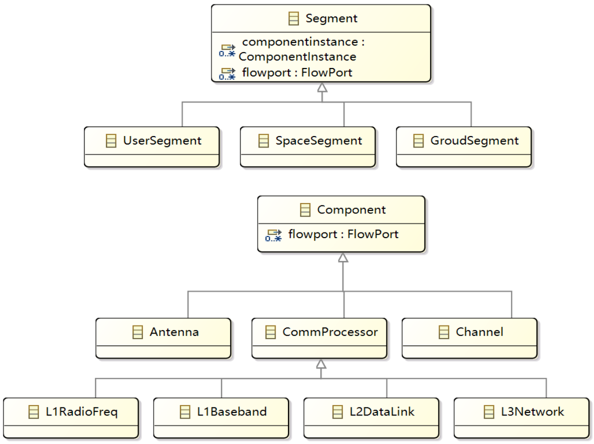


FIGURE 5: Architecture Metamodel.

Like functions have flows, architecture segments or components have ports. Flow ports are available for segments or components in a satellite network as an interface for them to communicate with other segments or components. Port metamodel is defined in Fig. 6. The information type transmitted through the port is defined by its port type, which can be divided into signal port and energy port. Signal port can be divided into wireless signal port and wired signal port. Port specifications are shown in Tab. 5.

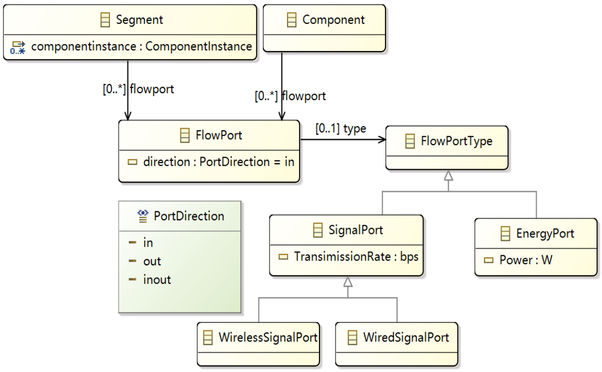


FIGURE 6: Architecture Port Metamodel.

TABLE 4: Architecture Specifications.

MOF Metaclass	Specifications
Segment	Part of a satellite network system containing a number of similar functioning objects.
User Segment	Part of a satellite network system containing application users.
Space Segment	Part of a satellite network system containing satellites forming a network which provides data transmission for users.
Ground Segment	Part of a satellite network system containing service management and control for all the users. It also serves as a service and application provider.
Component	Part of a satellite network system providing a specific capability in the satellite transmission procedure.
Antenna	Part of a satellite network system which transfers computer information into signal transmitted with microwave propagation.
CommProcessor	Part of a satellite network system which processes signal according to a prespecified communication protocol.
Channel	Part of a satellite network system serving as a signal transmission media.
L1RadioFreq	A CommProcessor providing physical layer (layer one) radio frequency processing. A satellite is known as transparent forwarding if only L1RadioFreq is available.
L1Baseband	A CommProcessor providing physical layer (layer one) baseband processing. A satellite is known to have onboard processing if L1Baseband module is available in addition to L1RadioFreq.
L2DataLink	A CommProcessor providing data link layer (layer two) media access and switch functions.
L3Network	A CommProcessor providing network layer (layer three) routing functions.

TABLE 5: Architecture Port Specifications.

MOF Metaclass	Specifications
Energy Port	Interface to deliver energy, e.g. heat, electricity. Attribute Power represents the power delivered through the port.
Signal Port	Interface to deliver electronic signal. Attribute TransmissionRate represents the transmission rate of the signal.
Wireless Port	Interface to deliver signal wirelessly.
Wired Port	Interface to deliver signal with wire.

Ports are related to architecture elements in Fig. 7. Segment is composed of instances of specific components. Both segment and component should be connected by ports.

B. PROFILE

The profile SysML4SatNetSim has four sub-profiles corresponding to the metamodels, which are requirement modeling profile, function modeling profile, architecture modeling profile and analysis modeling profile. Model elements in the traceability metamodel can reuse those in the language itself and thus do not need to be extended. Profiles and their relationships are illustrated in Fig. 8. Analysis profile will be presented later in the section considering simulation reusability.

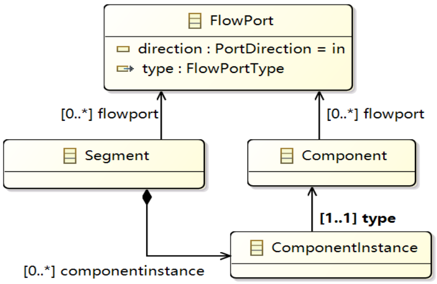


FIGURE 7: Architecture Block and Port Relationships.

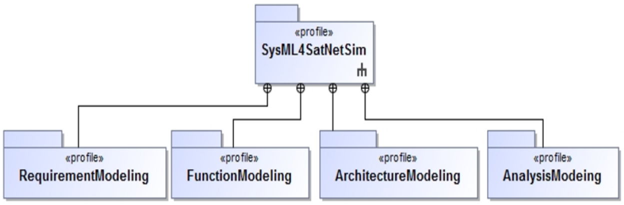


FIGURE 8: SysML4SatNetSim Profile.

1) Requirement Profile

Metaclasses in the requirement metamodel has the highest semantic similarity with the Requirement stereotype, based on which the requirement is extended. In addition, derivation between requirements can reuse the derivedReqt relationship in SysML. Requirement profile is shown in Fig. 9.

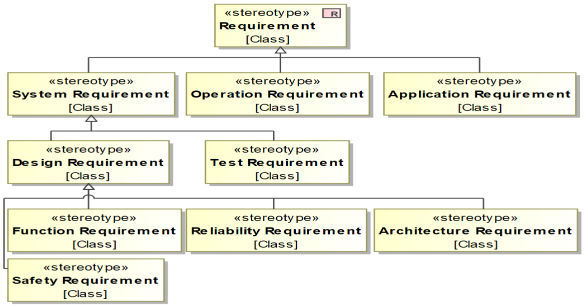


FIGURE 9: Requirement Profile.

2) Function Profile

Derived from metamodeling concepts, function profile is composed of three parts: functions and their subclasses representing various types of functions, flows between functions representing the information exchange mechanism, and flow information representing information exchanged in the flows between functions. In SysML, the most similar metaclass to function is Activity. Thus, it is seen in Fig. 10 that function and its subclasses are extended from the Activity metaclass. Object Flow is used to transmit information between Activities; it is then straightforward to extend Object Flow to obtain the stereotype flow.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

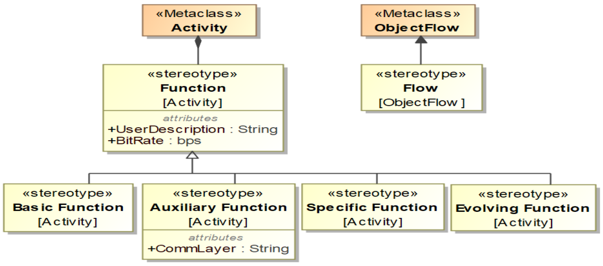


FIGURE 10: Function and Flow Profile.

In SysML, information transmitted in an Object Flow is expressed by its attribute Conveyed Information which can be realized by a Block. Therefore, Flow Information can be inherited from Block stereotype and divided into sub-stereotypes to identify different types of flow information as shown in Fig. 11. There are some attributes in the stereotypes to describe information conveyed by the flow. These attributes are defined as Tags of the stereotypes. For example, the stereotype MaterialFlow has PhysicalMedia as a Tag to describe the physical media used for material transmission.

It should be noted that these attributes are defined as Tags of stereotypes rather than value properties of the blocks. This is determined by how these elements are used. In an activity diagram, flow information inherited from a block should be assigned to Conveyed Information in an Object Flow rather than an instance of the block. Therefore, if the attribute of an information flow is defined as a value property of the block, the value assignment can then only be done in a block instance rather than the block. Hence Tag is used to define attributes of information flow rather than Value property.

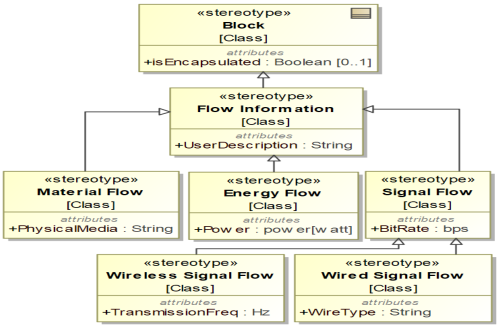


FIGURE 11: Flow Information Profile.

3) Architecture Profile

Architecture profile consists of three parts. The first one is "component" and "segment" and their subclasses. The second is flow port which supports information transmission. The third one is flow port type and its subclasses representing the type of information that can be transmitted through port. The best model element to match system segment and component is Block. Therefore, as in Fig. 12, "component" and "segment" and their subclasses are inherited from Block.

It should be noted that user models can be defined to describe system architecture based on these stereotypes. These user models have attributes for detailed description which should be defined as Value property of Block rather than Tag as in the function profile case. This is because these attributes belong to instance of the model rather than the model itself.

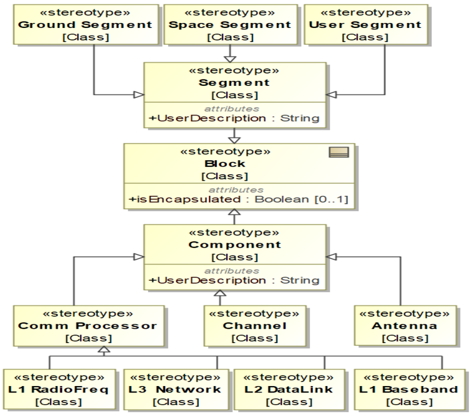


FIGURE 12: Block Profile.

In SysML, Port is used to exchange information between Blocks. Therefore, as in Fig. 13, flow port is extended from Port. In addition, flow port type is generalized from SysML stereotype Block. Signal port and electrical port are then defined which have Tags to describe Port attributes for the information transmitted.

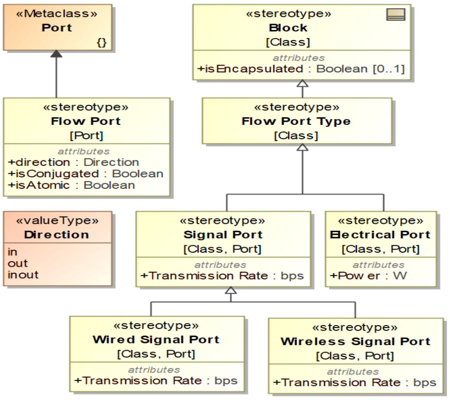


FIGURE 13: Port Profile.

V. MODEL REUSABILITY: FRAMEWORK WITH PATTERN VIEWPOINTS AND MODEL LIBRARIES

One of the cornerstones of model reusability is that of consistency and this is true of pattern definition just as much as it is of framework or model definition. A pattern should be clearly defined using an approach that is consistent with other patterns. In this section, a framework-based approach is outlined and key issues of relevant patterns are considered to ensure the consistency and reusability of viewpoints and model libraries.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

A. MODEL FRAMEWORK WITH PATTERNS

The "Seven Views" Framework of process modeling in [4] considers processes with activities together with their application context and execution behavior. This is conceptually similar to the satellite network simulation framework and thus is chosen as a baseline of our reusable model framework. Before modifying the framework accordingly, the "Seven Views" Framework is illustrated in Fig. 14 and its idea is outlined as follows.

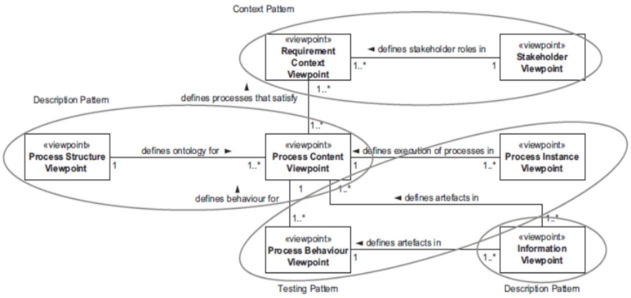


FIGURE 14: The "Seven Views" Framework for Process Modeling (Reproduced from Fig. 16.4 of [4]).

Process modeling usually has three inclusions. Firstly, there must be a mechanism for the process context to be identified and defined. Then, a mechanism for the definition of the process itself must be allowed, which must meet the needs and be consistent with the context defined in the previous point. Thirdly, based on the above two, there should be a mechanism in place to validate and test the process to see whether it is traceable against the needs and context.

The "Seven Views" Framework projects the above three aspects to elements of three well-defined patterns. Each of the viewpoints in Fig. 14 is a pattern viewpoint. Context pattern allows the needs and context to be expressed. It consists of stakeholder viewpoint and requirement context viewpoint, where the first viewpoint defines stakeholder roles in the second. Description pattern allows elements in the process to be defined. It has process structure viewpoint, process content viewpoint and information viewpoint in the framework. The structure viewpoint defines ontology for the process contents; and information viewpoint defines artefacts in the process contents. Test pattern allows the process to be tested and validated. It has process behavior viewpoint and process instance viewpoint, where the first one defines behavior for the process content and the second defines execution of processes in the process content.

However, not all the viewpoints are suitable or necessary for establishing the satellite network simulation application so the framework needs to be modified. The modification is illustrated in Fig. 15 and explained in the following discussions.

First consider the context pattern. The stakeholder viewpoint identifies the stakeholder roles and classifies stakeholder needs. This is similar to the functionality of needs and requirement modeling based on the requirement metamodel and profile and hence this viewpoint is not considered at this level. The requirement context viewpoint uses the context description viewpoint of the context pattern [4]. It defines the context of the process and identifies a number of use cases. This corresponds to the use case definition in satellite network model using OOSEM [3] and is indispensable in the framework. The name is changed to Context Viewpoint as in Fig. 15.

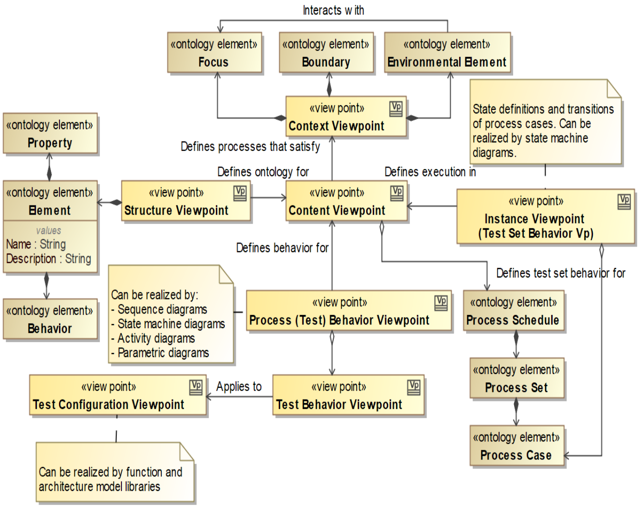


FIGURE 15: Modification of Process Modeling.

Next consider the description pattern. The structure viewpoint specifies concepts and terminology of a process model in the form of an ontology. It mainly borrows the idea from the element structure viewpoint of the description pattern [4]). This is a core modeling step and it is under the name structure viewpoint in our framework. It defines ontology for the content viewpoint, which identifies the actual processes and shows the activities carried out. Another viewpoint relevant to the description pattern in process modeling is the information viewpoint. It identifies the artefacts produced and consumed by activities within a process. This viewpoint is integrated into other viewpoints to make the framework compact and will be made clear later on.

In Fig. 14, part of the test pattern is adopted as instance viewpoint and process behavior viewpoint. The instance viewpoint shows instances of processes. In test pattern [4], however, more viewpoints are available for test set-up. Test set-up viewpoint is composed of test structure viewpoint and test set behavior viewpoint. The former, identifying the process schedule, set and case information, can be regarded as part of the content viewpoint. The latter is similar to the instance viewpoint. The process behavior viewpoint shows how a process behaves in terms of order of activities, flows through the process, resource usage and so on. In test pattern [4], it is part of the test case viewpoint. The process behavior viewpoint is applied to test configuration viewpoint, which can be realized with function and architecture model libraries based on their corresponding metamodels and profiles.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

1) Viewpoints overview

To make the framework clear and concise, details in Fig. 15 are omitted and we are left with viewpoints overview in Fig. 16. Context viewpoint is the origination of the framework and content viewpoint defines processes that satisfy the context viewpoint. Structure viewpoint defines ontology for the content viewpoint. Set-up viewpoint defines execution of processes in the content viewpoint. Execution viewpoint defines behavior for the content viewpoint. The behavior defined is applied to configuration viewpoint, which is actually function and architecture model libraries in the next subsection. Before discussing the libraries, some of the key viewpoints are investigated as follows to make their connections to the satellite network domain clear.

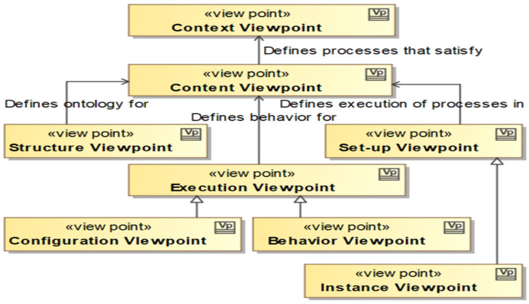


FIGURE 16: Viewpoints Overview of the Model Framework.

B. VIEWPOINTS REALIZATION AND MODEL LIBRARIES

1) Structure Viewpoint (Ontology)

The structure viewpoint defines ontology for the content viewpoint by describing model elements in terms of their properties and behaviors. It also defines, for one element, the relationship with other elements and how it breaks down into parts. The ontology elements and their relationships for the satellite network simulation framework is shown in Fig. 17.

The context viewpoint in Fig. 15 with ontology elements focus, boundary and environmental elements defines the contexts for the framework. The context represents the need for the process schedule in the content viewpoint as in Fig. 17. Additional relationships of applications and use cases with the content viewpoint is defined. The process schedule realizes applications and satisfies use cases. The use cases refine the applications. The process case, as the concrete element of the process schedule, has execution defined by process instances. Activity and artefact ontology elements compose of the process case, where the behavior for the former is defined by process behavior. The behavior is applied to process configuration. The models developed in the satellite network simulation framework should comply with this set of ontology definition and their relationships.

2) Content Viewpoint (Process Library)

This viewpoint works as a core of the whole framework as it defines the processes to be executed in various stages of the simulation and thus can be regarded as a process library.

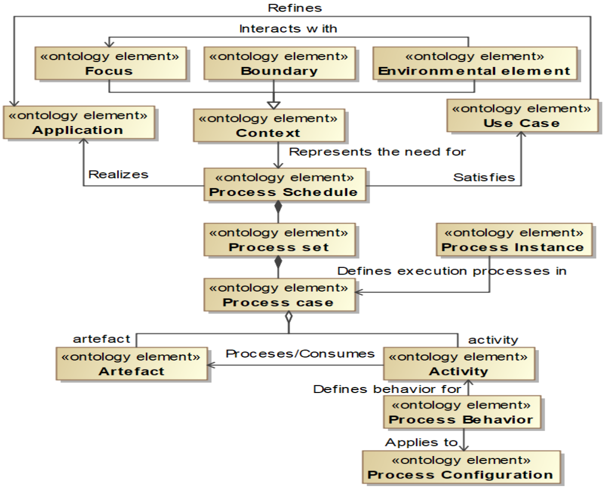


FIGURE 17: Structure Viewpoint.

It is modified from the test structure viewpoint of the test pattern from [4] to account for satellite network simulation characteristics. As shown in Fig. 18, the overall process schedule is divided into effective and performance process sets.

The effective process set has three types of process cases. Application process cases considers processes of satellite network user applications. Bidirectional communication has narrowband (handheld) and wideband (vehicular) processes; one-directional transmission includes data collection and broadcasting processes. Context process cases identifies processes with system context, including focus, boundary and environmental processes. Use case processes enumerates cases based on service priority with non-real-time, real-time and emergency processes.

The performance process set has two types of process cases. Function process cases contain all the basic operations for a communication procedure, including send, transmit, receive and onboard processing capabilities complying with the OSI seven-layer model. Artefact process cases are elements transferred from the information viewpoint of the original "seven-views" framework in [4]. It is known that artefacts are produced or consumed in certain components of the architecture and thus the process cases in this subset models the production and disappearance of artefacts during the process.

3) Instance Viewpoint

This viewpoint is modified from the test set behavior viewpoint in the test pattern of [4] and gives the execution sequence of the process cases in the content viewpoint. In the effectiveness process set, application process cases are firstly carried out to account for stakeholder needs, followed by context and use case process cases. In the performance set, function and artefact process cases are executed and simulated sequentially. This is illustrated in Fig. 19.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

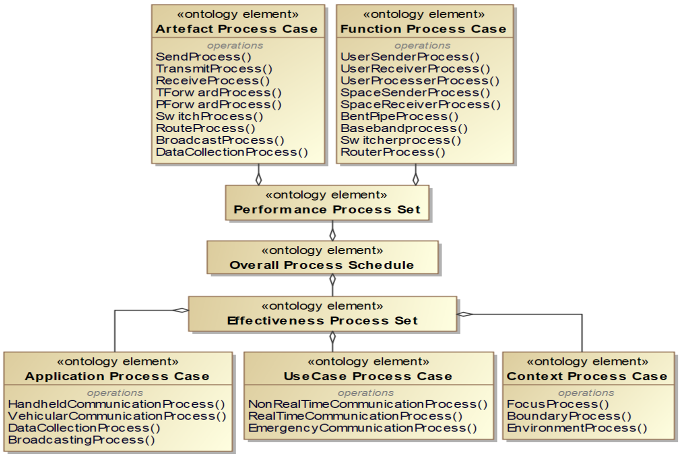


FIGURE 18: Content Viewpoint.

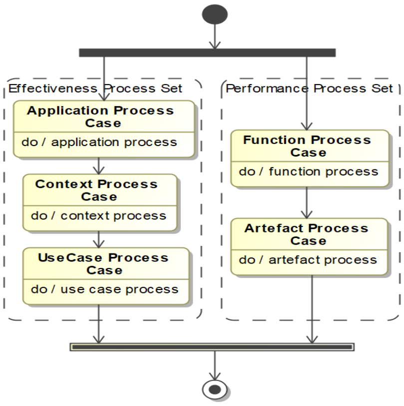


FIGURE 19: Instance Viewpoint.

4) Behavior Viewpoint

This models process behavior and can be realized with a number of SysML behavior diagrams. In particular, state machine diagram is used for defining high-level scenarios; activity diagram models execution of activities in the simulation procedure; sequence diagram is used for detailed process description in critical design and simulation stage; parametric diagram can be used for defining analysis and performance tradeoff scenarios. Examples of behavior models using the above diagrams can be found in [3].

5) Configuration Viewpoint (Function and Architecture Libraries)

As mentioned before, this viewpoint is realized as function and architecture model libraries. This has direct and fundamental connection with the concept reusability presented in the last section. The domain specific semantics in the metamodel and profile provides reusable mechanism for function and architecture modeling and connect, in a multi-view framework, the model reusability and the concept reusability. This provides a mechanism to preserve reusability when complex system details should be reflected in models. At the same time, multiple viewpoints give comprehensive and well-structured understanding of the system.

Function model library and architecture model library are presented in Fig. 20 and Fig. 21, respectively. It can be seen that elements in them are constructs based upon relevant profiles. These libraries can be utilized directly by satellite network simulation system modelers to preserve reusability and speed up their modeling procedure.

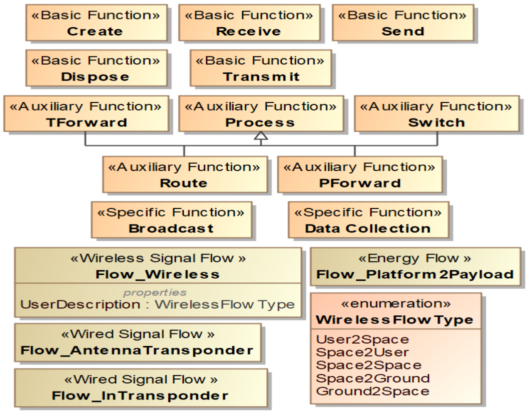


FIGURE 20: Function Model Library.

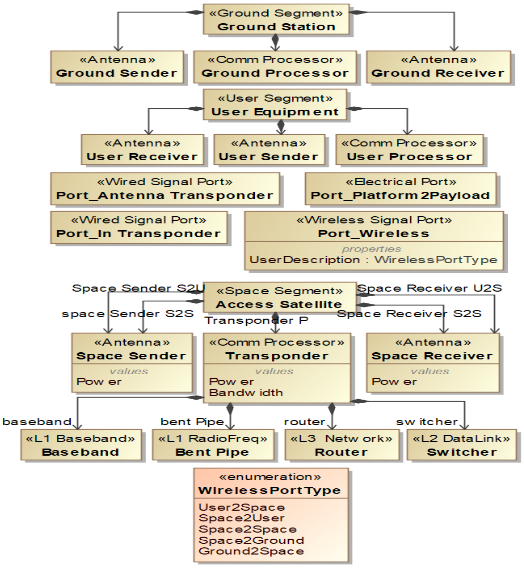


FIGURE 21: Architecture Model Library.

VI. SIMULATION REUSABILITY: ANALYSIS MODELING AND DESIGN-ANALYSIS INTERACTION

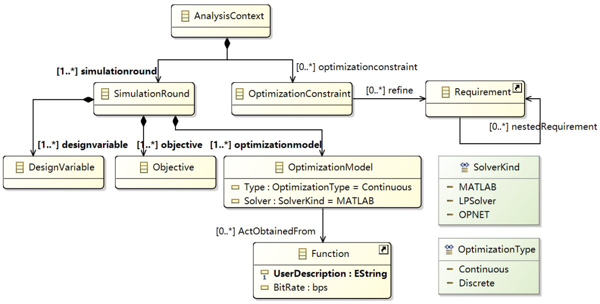
A. ANALYSIS MODEL

1) Analysis Metamodel

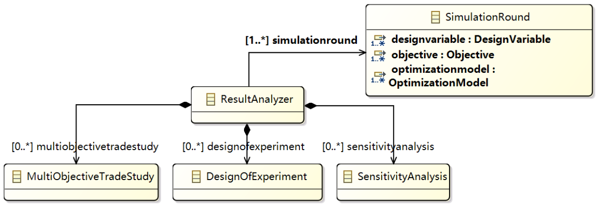
Analysis metamodel gives modeling methods for analysis as shown in Fig. 22. Analysis context should be properly defined before and during simulation. Optimization constraints should be consistent with requirements and hence there is a Refine relation in the figure. Optimization model used during simulation should obtain activities from function models.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

Thus, a newly defined relation ActObtainedFrom is used for this type of connection. For modeling elements after simulation, a number of result analyzer tools are defined and connected among simulation rounds.



(a) Before and During Simulation



(b) After Simulation

FIGURE 22: Analysis Metamodel.

2) Analysis Profile

Analysis profile in Fig. 23 relates design in SysML and simulation in multi-disciplinary software. The procedure can be divided into three steps. First, before simulation begins, task to be performed should be established as analysis context. Regarding the architecture design as an optimization problem, objectives, constraints and design variables should be setup in this stage. Then, during simulation, optimization model contains the mathematical model and simulation activities used in multi-disciplinary software. It is the mapping of satellite network models in network simulation software into elements in SysML modeling software. SolverKind enumerates simulation software used. Third, after simulation, results (not the raw data but the one after processing) should be returned to SysML modeling software for architects to make design decisions. These results include design of experiment, multi-objective trade study and sensitivity analysis.

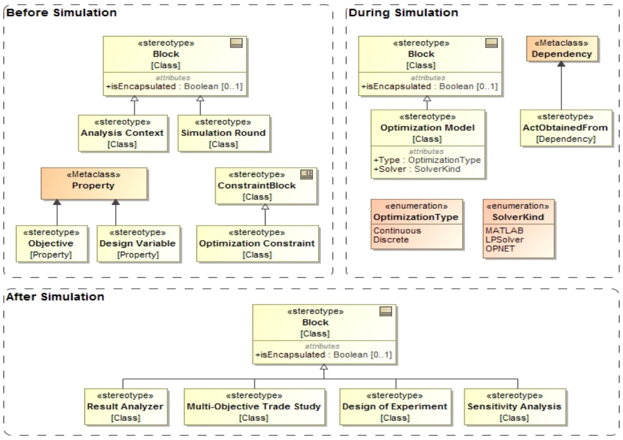


FIGURE 23: Analysis Profile.

B. REUSABLE ARCHITECTURE DESIGN MECHANISM

The reusable mechanism for iteratively simulating more complex system characteristics is illustrated in Fig. 24. From round 1 to subsequent rounds, number of parameters passing between modeling and simulation activities increase to reflect more system views and elements. Analysis user model developed based on analysis profile has relations with requirement, function and architecture models. Specifically, it gets objective and design variable requirements from requirement user model; it also gets optimization model from function user model. These values are input into simulation tools for the execution of multi-objective optimization. The obtained Pareto set and design variable values are sent back to the system modeling tool.

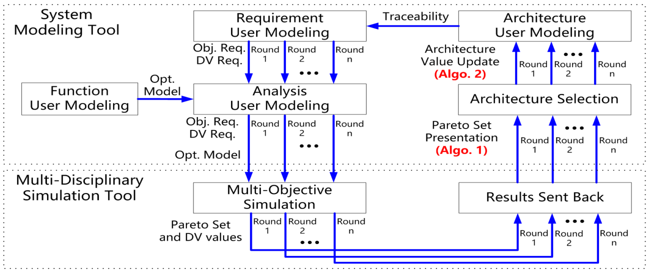


FIGURE 24: Design-Simulation Iteration.

Before proceeding to the reusable mechanisms, related terminologies for SysML and its application programming interface (API) [20] are presented as in Tab. 6. They will be used in subsequent algorithms.

TABLE 6: SysML API Terminologies.

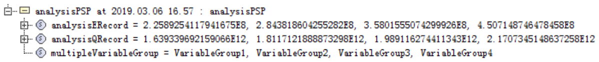
No.	Term	Explanation
1	Tree, Node, Object	Hierarchical structure in SysML database
2	Package	Element for storing a set of data
3	Property	Element for storing architecture parameter
4	SlotValue	Value returned from other tools to be processed in SysML
5	InstanceTable	Table in SysML to visualize SlotValues
6	ValueSpec	Value specification of SlotValues
7	ValueType	Value type in architecture model for SlotValue assignment

1) Reusability Among Multiple Simulation Rounds

There are two ways to process and visualize returned data to SysML. The first one is to take values directly from SysML database tree structure as shown in Fig. 25(a). But this is inconvenient for further data analysis and gives little

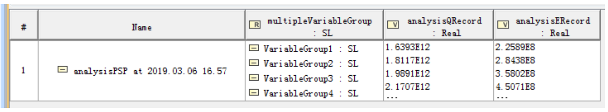
This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

visualization of data relations. The second one is to generate instance table using default SysML function as in Fig. 25(b). However, this gives instance table with all data presented together without row separation, which is inconvenient for further data processing. In addition, the type and amount of returned data are increased as multiple simulation iterations take in more objectives and/or design variables. Hence in a reusable manner, instance tables should be set as can be extended to any number of columns.



```
analysisPSP at 2019.03.06 16.57 : analysisPSP
    analysisERecord = 2.2589254117941675E8, 2.843818604255282E8, 3.580155507429926E8, 4.507148746478458E8
    analysisQRecord = 1.639339692159066E12, 1.8117121888873298E12, 1.989116274411343E12, 2.1707345148637258E12
    multipleVariableGroup = VariableGroup1, VariableGroup2, VariableGroup3, VariableGroup4
```

(a) Tree Structure for PSP

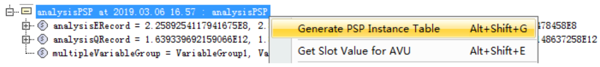


#	Name	multipleVariableGroup : SL	analysisQRecord : Real	analysisERecord : Real
1	analysisPSP at 2019.03.06 16.57	VariableGroup1 : SL VariableGroup2 : SL VariableGroup3 : SL VariableGroup4 : SL	1.6393E12 1.8117E12 1.9891E12 2.1707E12 ...	2.2589E8 2.8438E8 3.5802E8 4.5071E8 ...

(b) Auto-Generated SysML Instance Table

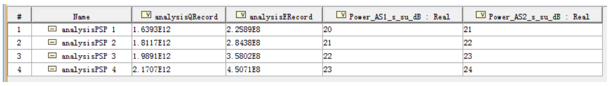
FIGURE 25: Default Pareto Set Presentation.

To solve the above problems, plugins are developed via SysML API to generate reusable Pareto set presentation as in Fig. 26(a). Algorithm 1 gives the method to implement this in an API tool. The generated reusable tabular format is shown in Fig. 26(b). In addition to optimization objectives Q and E, the table also presents values of two design variables corresponding to the objectives. These can be used for further analysis in SysML to select architecture points. The tabular format is designed to be extended to any number of objectives and DVs to support reusability between multiple optimization rounds.



```
analysisPSP at 2019.03.06 16.57 : analysisPSP
    analysisERecord = 2.2589254117941675E8, 2.
    analysisQRecord = 1.639339692159066E12, 1.
    multipleVariableGroup = VariableGroup1, Va
```

(a) Plugin Developed via SysML API



#	Name	analysisQRecord	analysisERecord	Power_AS1_s_su_dB : Real	Power_AS2_s_su_dB : Real
1	analysisPSP 1	1.6393E12	2.2589E8	20	21
2	analysisPSP 2	1.8117E12	2.8438E8	21	22
3	analysisPSP 3	1.9891E12	3.5802E8	22	23
4	analysisPSP 4	2.1707E12	4.5071E8	23	24

(b) Reusable Tabular Format for PSP

FIGURE 26: Reusable Pareto Set Presentation.

2) Reusability Within One Simulation Round

After Pareto set is presented with proper tabular format, architecture selection can be performed by manipulating and visualizing data in the table. Within one simulation round, this table is reusable to select architecture points based on different criteria. For example, for the tradeoff of minimizing one objective and maximizing another, the criteria can be choosing the architecture point, from the Pareto set, with medium objective 1 or maximum objective 2. This depends on user needs, preferences and requirements. SysML architecture model should be updated to account for these variable requirements with reusable and automatic updating mechanism. This brings the architecture value update mechanism shown in Fig. 24. Slot value is obtained from the Pareto set presentation using the plugin "Get Slot Value for AVU" in Fig. 26(a). Algorithm 2 gives the method to implement this in an API tool.

algorithm 1 Pareto Set Presentation (PSP).

```
Get $Tree$, $Node$, $Object$;
While $Object$ is of type $Package$
    Create element selection dialog;
    Select elements;
    Create session for PSP using $InstanceTable$;
    Repeat
        Create new $InstanceTable$;
        Create $Property$ (for columns) of $InstanceTable$;
        Add $InstanceTable$ columns for objectives;
        Add $InstanceTable$ columns for design variables;
        Calculate $InstanceTable$ row number N;
        Record $Property$-$ValueSpec$ relationship;
        Obtain $SlotValue$;
        Add $SlotValue$ to corresponding columns based on
        $Property$-$ValueSpec$ relationship;
    Until Session for PSP using $InstanceTable$ is closed;
EndWhile
```

algorithm 2 Architecture Value Update (AVU).

```
Get $Tree$, $Node$, $Object$;
While $Object$ is of type $Property$
    Create element selection dialog;
    Select elements;
    Create session for setting architecture value;
    Repeat
        Obtain $SlotValue$ for design variables;
        Record $ValueSpec$ property;
        If $Object$ contains $ValueType$
            Set $SlotValue$ for $ValueType$;
        Else
            Set default value;
    Until Session for setting architecture value is closed;
EndWhile
```

VII. MULTI-OBJECTIVE SATELLITE NETWORK SIMULATION

A. EXTENDED TIME-EVOLVING GRAPH FOR SATELLITE COMMUNICATION NETWORKS

Satellite communication network simulation is within the system engineering framework discussed in previous sections where detailed optimization are carried out across multiple layers of the system. The time-varying topology of LEO satellite network due to orbit movement relative to the earth surface can be characterized by predictable connections among satellites [29]. In this context, the network model can be expressed by using the (ETEG) [30] which characterizes network resources in both spacial and temporal dimensions.

A general satellite communication network consists of a set of user equipments $U = \{u_1, u_2, ..., u_L\}$, satellites $S =$

$\{s_1, s_2, ..., s_M\}$ and ground stations $G = \{g_1, g_2, ..., g_N\}$. A time-slotted system [30] divides the topology and real-time connections of the network elements into consecutive time slots indexed by $t \in \Gamma = \{1, ..., T\}$. Each time slot has a duration of $\Delta\tau$ and the network topology and connection is fixed during each time slot. The ETEG representation is thus defined as follows.

Definition 1: ETEG Representation for Satellite Communication Network. Define a satellite communication network by a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The set of vertices is denoted as $\mathcal{V} = \mathcal{V}_u \cup \mathcal{V}_s \cup \mathcal{V}_g$ where $\mathcal{V}_u$, $\mathcal{V}_s$ and $\mathcal{V}_g$ correspond to the replicas of user equipments, satellites and ground stations, respectively. The set of edges is denoted as $\mathcal{E} = \mathcal{E}_l \cup \mathcal{E}_b$ where $\mathcal{E}_l$ is the set of link edges and $\mathcal{E}_b$ is the set of processing edges. $\mathcal{E}_l$ can further be divided as $\mathcal{E}_l = \mathcal{E}_{su} \cup \mathcal{E}_{ss} \cup \mathcal{E}_{sg}$ where $\mathcal{E}_{su}$, $\mathcal{E}_{ss}$ and $\mathcal{E}_{sg}$ models the user links, inter-satellite links (ISLs) and feeder links, respectively.

The graph elements should be expressed with time-slotted notations for modeling and calculation:

- The sets of vertices can be expressed as $\mathcal{V}_u = \{u_i^t \mid u_i \in U, 1 \le t \le T\}$, $\mathcal{V}_s = \{s_i^t \mid s_i \in S, 1 \le t \le T\}$ and $\mathcal{V}_g = \{g_i^t \mid g_i \in G, 1 \le t \le T\}$.
- The edges in $\mathcal{E}_{su}$ model the links between user equipments and the satellite network in each time slot, i.e. $\mathcal{E}_{su} = \{(s_i^t, u_j^t) \cup (u_j^t, s_i^t) \mid 1 \le t \le T, u_j \text{ within } s_i \text{ coverage in } t\}$. The edges in $\mathcal{E}_{ss}$ model the ISLs between satellites within the satellite network in each time slot, i.e. $\mathcal{E}_{ss} = \{(s_i^t, s_j^t) \cup (s_j^t, s_i^t) \mid 1 \le t \le T, s_j \text{ within } s_i \text{ coverage in } t\}$. The edges in $\mathcal{E}_{sg}$ model the links between the ground station and the satellite network (feeder link) in each time slot, i.e. $\mathcal{E}_{sg} = \{(s_i^t, g_j^t) \cup (g_j^t, s_i^t) \mid 1 \le t \le T, g_j \text{ within } s_i \text{ coverage in } t\}$.
- Processing edges are modeled to represent the capability of satellites to store data between consecutive time slots. The set of such edges is denoted as $\mathcal{E}_b = \{(v_i^t, v_i^{t+1}) \mid v_i^t \in \mathcal{V}_s, 1 \le t \le T-1\}$.

Information passing through the network should be encapsulated according to certain protocols such as the open system interconnection (OSI) model [31]. Satellite payload can be designed as "transparent" to perform only physical layer (layer 1) functions, or with onboard processing to include data link layer (layer 2), network routing (layer 3) and upper layer capabilities. The following definition is to classify payload functionalities into different protocol layers.

Definition 2: Layer Label. Label $y \in \{1, ..., Y\}$ represents the highest layer of the information to be processed in a satellite payload following a predefined communication protocol, where Y is the highest layer in the protocol.

Example 1: Non-real time mission flows can be transmitted in the satellite network only in the physical layer with $y = 1$ and higher layer access and routing are left to ground station. Real-time missions requires that the payload should be equipped with network routing with $y = 3$ to process the information and find routing within satellites as it may be a waste of time to go back to the ground stations. Protocols following the OSI standard have $Y = 7$.

It is necessary to treat information flows from various layers separately since they bring different capability (transmission capacity, satellite network level) or requirement (power, satellite platform level) to the model. The layer label is thus regarded as a shared parameter among network, satellite and payload/platform level MBSE models which can further be related to parameters of these levels. As seen in the previous example, layer label represents mission classification in network level. It also guides platform power allocation to the payload as higher level processing requires more power.

Let M_z be a mission carried out by the satellite network with $1 \le z \le Z$, where Z is the total number of missions for the application scenario concerned. Each mission M_z can be divided into a number of flows to be transmitted in the network.

Definition 3: Flow and Flow Element. Define the k-th flow of the z-th mission with label y as $f_{k_z}(y)$. In the ETEG framework, a flow is characterized within each time slot t and allocated to certain user equipment i where $1 \le i \le L$. Define flow element as $f_{k_z}(i, t, y)$, which allocates $f_{k_z}(y)$ to the i-th user equipment (started by user i) in the t-th time slot.

It can be deduced that the set of flows corresponding to a user equipment is $f_{k_z}(i, y) = \{f_{k_z}(i, t, y) \mid 1 \le t \le T\}$. The set of flows originated in certain time slot t is $f_{k_z}(t, y) = \{f_{k_z}(i, t, y) \mid 1 \le i \le L\}$. Thus, a flow is thoroughly captured by its flow elements in all the related UEs and time slots, i.e. $f_{k_z}(y) = \{f_{k_z}(i, t, y) \mid 1 \le t \le T, 1 \le i \le L\}$. The formal definition of a mission is hence as follows.

Definition 4: Mission. A mission is defined as $M_z = \Psi_{1 \le k_z \le K_z} f_{k_z}(y)$, where Ψ represents the arrangement of the K_z flows of the mission M_z.

The set of all the missions for the application scenario with Z missions in total is represented as $\mathcal{M} = \cup_{1 \le z \le Z} M_z$.

B. MULTI-OBJECTIVE OPTIMIZATION

Non-geostationary orbit (NGSO) satellite constellation has time-varying network topology because of the orbit movements of satellites. The resulting intermittent but predictable connections between satellites make the ETEG model defined previously useful in characterizing network behavior. The model discussed here has been extended from the one for remote-sensing networks in [12] to satellite communication networks. Definition 1 and the discussions followed basically modifies the remote-sensing satellite network carrying one-direction data stream from user equipment (UE) to ground management and control center (GMCC) to one that can handle bi-directional data streams supporting communications among users within the coverage of the satellites.

In addition, it has also been pointed out in [12] that resources onboard satellites are limited for data transmission and the number of connections that can be supported in one satellite is limited. The link capacity hence the power needed is also time-varying due to various propagation channel conditions of user-satellite links and inter-satellite links. The authors in [12] formulated this problem as a single-objective problem optimizing the network profit with power and contacts as constraints. However, this model is only tractable for network with a few satellites and the complexity becomes higher when increasing the network size. Taking the point of view of optimizing the satellite design based on the performance it can provide to users at the network level, it has been shown that satellite power consumption can be included as an additional metric to form a multi-objective optimization problem [11]. Transmission capacity requirements is met together with efficiently using satellite resources.

1) SysML-Model-Driven Simulation

Network optimization can thus be solved using heuristic techniques. In contrast to single-objective problems, multi-objective optimization provides tradeoff between two conflicting functions. The tradeoff achieved among the two gives a set of non-dominated solutions known as the Pareto optimal solutions [32]. Fuzzy Pareto front is often obtained by using heuristic algorithms, determining a set of near-optimal solutions for designers to choose rather than one single optimal solution. This is useful in system engineering while architects try to decide among multiple design variables. The complexity problem when increasing the network size can hence be avoided without directly using the one-objective optimization with traditional ETEG model.

It has been suggested that network performance should take mission differentiation into account [11] [12]. Different mission sources and purposes may result in diverse values of data delivery. However, the models so far only considered differentiated data volumes. Considering network modeling and simulation connecting with SysML models, it is necessary to characterize missions based on user requirements, use cases and application scenarios. In particular, flows should further be divided based on satellite communication service types. For example, services for handheld or ship-mounted user equipments transmit different types of flows. In addition, for each type of UE, various service categories such as non-real-time, real-time or emergency requires satellites to provide different resources to support the services. The layer label in Definition 2 gives a mechanism to describe different service processing capabilities in the satellite. Furthermore, network simulation should be connected with SysML models in that the two should have corresponding parameters with the same granularity that can be interchanged. The flow and flow element in Definition 3 offer this mechanism. In this way, the application-oriented SysML model and performance-oriented simulation of a particular subject can be integrated.

2) Objectives and Design Variables

A number of application scenarios are generated and input into the satellite network to simulate capacity transmitted and satellite power consumed. Power consumption should consist of those from data transmission in the wireless channel and onboard processing, which corresponds to the link edges and processing edges in the ETEG model, respectively. Transmission capacity is computed, after UEs connect to the network and arrive their access satellite with certain arrival rates. Data forwarding or routing within the satellite network is computed in a centralized manner which routing information are computed in ground stations and distributed to each satellite in the network.

Advanced techniques such as flexible payload can optimize power consumption within the satellites [13]. The model in this paper supports this flexibility in that, through the tradeoff of power distribution among various payload functionalities, a non-dominated set of Pareto optimal solutions is obtained. This can be feeded back to the SysML model and serves as a design guide for the flexible payload [33] at satellite manufacturing stage.

When considering single satellite design, the component consuming the most power is the user downlink TWTA rather than the ISL transmission parts [13]. Power distribution variables are thus set up based on this with power consumption related to various services. This is seen as the extension of the fixed data rate in [13] to account for multiple service types. The model is thus formulated as follows.

3) Optimization Problem Formulation

A number of notations are firstly defined to be used in the optimization problem.

TABLE 7: Notations for Satellite Network Model.

No.	Symbol	Definition
1	$i-s-su$	Satellite i payload to send signal from satellite i to users (through user link)
2	$i-s-ss$	Satellite i payload to send signal from satellite i to other satellites (through ISL)
3	$i-r-us$	Satellite i payload to receive signal from users to satellite i (through user link)
4	$i-r-ss$	Satellite i payload to receive signal from other satellites to satellite i (through ISL)
5	$i-p$	Satellite i payload to do onboard processing

The multi-objective optimization problem aims at maximizing system capacity C while at the same time minimizing total power consumption in the satellite network:

$$\max C = \max \sum_{f_{k_z}(y)\in\mathcal{M}} \sum_{s_i:(s_i,u_j)\in\mathcal{E}_{su}} R(P_{i-s-su}) \quad (1)$$

$$\begin{aligned}\min P = \min \sum_{f_{k_z}(y)\in\mathcal{M}} \sum_{s_i\in\mathcal{V}_s} (P_{i-s-su} + P_{i-s-ss} \\ + P_{i-r-us} + P_{i-r-ss} + P_{i-p})\end{aligned} \quad (2)$$

For system capacity maximization, the rate can be calculated as

$$R(P_{i-s-su}) = B_0 \log(1 + SNR_i) \quad (3)$$

with

$$SNR_i = \frac{g_i^2 P_{i-s-su}(OBO)}{N_0 B_0} \quad (4)$$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

where (according to [34])

- $P_{i-s-su}(OBO)$ is the power allocated to the $i-s-su$ link with TWTA output backoff OBO;
- g_i is the link gains and losses including satellite beam antenna gain towards the intended coverage area, gain of the receiving antenna, free space loss and other losses;
- N_0 is the noise power spectral density depending on the receiving antenna and equivalent noise temperature;
- B_0 is the signal bandwidth.

For power consumption minimization, it is known that the payload sending signal from satellite i to ground user equipment is the transmission rate bottleneck among all the payload links. Thus, the rate calculated from P_{i-s-su} is regarded as the rate the satellite can transmit for the UE service. Power consumption by other payload functionalities can be calculated from P_{i-s-su} as:

$$P_{i-r-us} = \alpha_{r-us} \cdot R(P_{i-s-su}) \tag{5}$$

$$P_{i-s-ss} = \alpha_{s-ss} \cdot R(P_{i-s-su}) \tag{6}$$

$$P_{i-r-ss} = \alpha_{r-ss} \cdot R(P_{i-s-su}) \tag{7}$$

$$P_{i-p} = \mu \cdot (R(P_{i-s-su}))^{\gamma} \tag{8}$$

where the values of α_{r-us}, α_{s-ss}, α_{r-ss}, μ and γ can be determined based on the model in [13].

It can be seen that the calculation of system capacity only takes the downlink data stream of $(s_i, u_j) \in \mathcal{E}_{su}$ into account whereas the calculation of power consumption considers all the satellites in the network $s_i \in \mathcal{V}_s$. This is because the routing and relevant data sending/receiving consumes power of every satellite the data stream passes by.

4) Problem Transformation

Services originated from ground UEs arrives at access satellites randomly according to certain arrival rate. Meanwhile, an NGSO satellite moves relative to the earth surface so there is only a periodical fixed time duration that the satellite can serve the ground UEs. Therefore, capacity calculation can be transformed to data quantity Q transmitted for services generated when UEs are within the satellite coverage area. Similarly, power calculation can be transformed to energy E consumed in the network for transmitting the service. The optimization problem can thus be rewritten as

$$\begin{aligned} \max Q \quad &= \max \sum_{f_{k_z}(y)\in\mathcal{M}} \sum_{s_i:(s_i,u_j)\in\mathcal{E}_{su}} \\ &\qquad R(P_{i-s-su}) \cdot t_{dur}(P_{i-s-su}) \qquad (9) \\ \min E \quad &= \min \sum_{f_{k_z}(y)\in\mathcal{M}} \sum_{s_i\in\mathcal{V}_s} \\ &\qquad [(P_{i-s-su} + P_{i-r-us}) \cdot t_{dur}(P_{i-s-su}) \\ &\qquad + (P_{i-s-ss} + P_{i-r-ss} + P_{i-p}) \cdot T_{min}] \qquad (10) \end{aligned}$$

where $t_{dur}(P_{i-s-su})$ represents the time duration with service arrivals when UEs are within the coverage area of the satellite. It is a function of P_{i-s-su} since the power level of satellite downlink transmission determines the duration that UEs can connect to the satellite. T_{min} is the minimum simulation time for the service transmission in the network when satellite power levels for all the functionalities are above a pre-specified threshold. $i-s-ss$, $i-r-ss$ and $i-p$ links consumes energy for the service no matter whether the ground link is connected or not; so the time used when calculating their energy consumption is the whole simulation time T_{min}. On the other hand, $i-s-su$ and $i-r-us$ links only consumes energy when the satellite and UEs are connected.

5) Design and Simulation Interaction

The multi-objective optimization problem proposed above involves tradeoffs between two conflicting objectives to determine a set of design variables, where the power allocation proportion among the payload functionalities of a single satellite is obtained based on the fuzzy Pareto front architecture points [3]. The correspondence between this analysis problem and the SysML model can be formed. The objectives are at the system level where the "system" capacity and "system" resource usage is optimized. When carrying out the simulation, activities for application scenarios should be executed which is 'above' the system level and corresponds to the requirements and use cases in the SysML model. After the simulation, the balancing of design variables is to choose power distributions which is 'below' the system level. This corresponds to the architecture model in SysML after logical decomposition when determining among alternative physical architectures.

As mentioned before, this separation of engineering analysis and simulation components into corresponding parts of the SysML model integrates design and simulation process. Based upon the interface developed in Section VI, interaction between the multi-disciplinary SysML models and the simulation of a particular subject area forms an iterative loop to consistently optimize the system architecture. This is essential in architecture design of complex systems such as satellite networks, especially when multiple levels (network, satellite, payload, TWTA component, etc.) of design variables are considered within the problem. This, together with the extension schemes based on metamodels and pattern viewpoints, makes the system still tractable when the system models are expanded along the manufacturing stages.

VIII. CASE STUDY AND RESULT ANALYSIS

A. SYSML USER MODEL: BEFORE SIMULATION

SysML user model is developed based on domain specific profiles, pattern viewpoints and related model libraries. Before simulation, requirement and function user models should be set up and related to corresponding elements in analysis user model.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

1) Requirement User Model

It classifies requirements based on requirement profiles. Top level system requirement is to develop a satellite network simulation system as in Fig. 27. Application requirements include quantitative objective of network capacity and possible application supported such as handheld and vehicular communication, data collection and broadcasting. Function requirements describe relevant functions the system has, including non-real-time, real-time and emergency communications. Architecture requirements raise design variables to be decided from simulation, which are power allocation to user downlink in access satellite 1 and 2[1], and average bandwidth in payload transponder. Model reusability is demonstrated in that the first two design variables are optimized in the first round where the third variable is added in the second-round optimization by *reusing* analysis models and simulation software interface. Operation requirement gives the maximum resource that can be used. Calculation of quantitative requirement will be made clear in subsequent discussion.

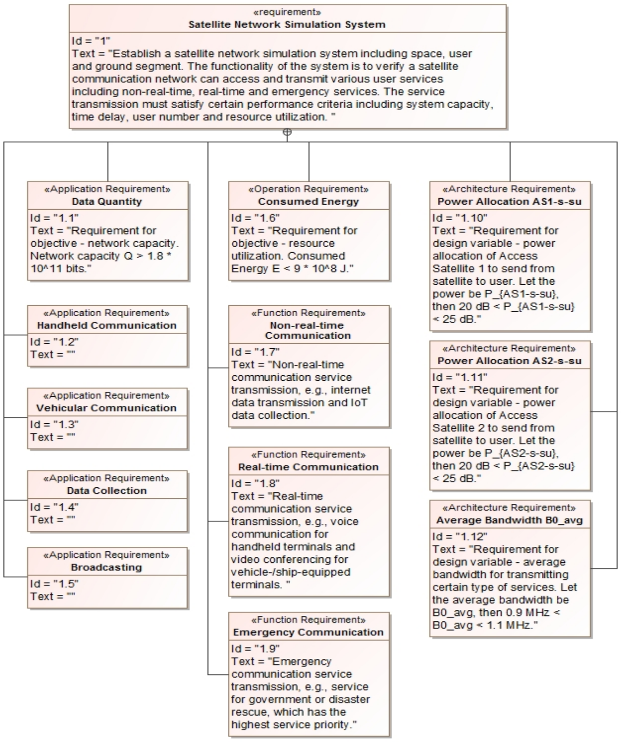


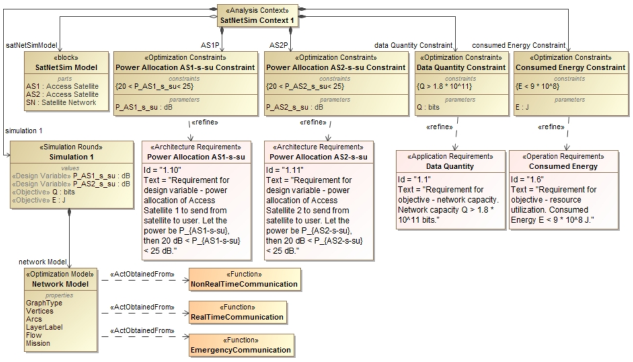
FIGURE 27: Requirement User Model.

2) Function User Model

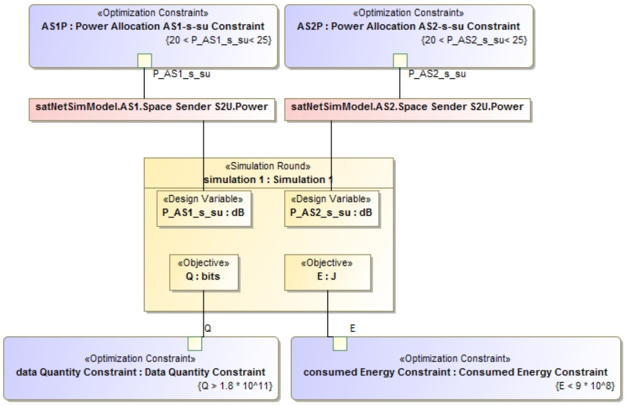
This is formed with process cases of content viewpoint and function model library of process configuration viewpoint. These models are executable and serve as a mechanism to carry out simulation. Interaction with simulation software can be done through SysML diagrams such as state machine, activity and sequence diagrams. Function user models are omitted here and examples for satellite network can be found in [3].

3) Analysis User Model Before Simulation

It is firstly set by relating objectives and design variables with their bounded requirements. In addition, design variables should also be related with architecture user model for setting architecture values after simulation. Then, network model carrying simulation parameters is related to function user model and used during simulation. These are illustrated in Fig. 28.



(a) Simulation Context



(b) Relations Among Parameters

FIGURE 28: Analysis User Model Before and During Simulation.

B. CASE STUDY SETUP

It has been discussed that optimization of objectives Q and E are considered with respect to optimizing design parameters within one satellite. This is different from network simulation which only optimizes network level parameter design and normally takes satellite as a black box. This is also different from the point of view of satellite design and optimization, which considers satellite payload, platform and spacecraft dynamics. The point of view is, by making use of the multi-level and multi-disciplinary modeling capability of MBSE,

---

[1] The reason for two access satellites is to separate the cases when two UEs get access from a single satellite and two. Energy consumption for access satellites are different in these two settings [35].

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

to incorporate multi-level parameters and solve key system design problems.

In the optimization setting of the previous section, Q is to maximize data quantity that can be transmitted during access time to a satellite. E is to minimize energy consumption for transmitting Q in access satellite and relay satellites. Design variables are power allocation to different payload functionalities in a satellite. Objective calculation should be simulated for service transmission under a number of application scenarios. In this case, application scenarios and user access to satellites are at the network level; data quantity through user link and energy consumed are at the system level; and power distribution to payload components are at the subsystem level.

The choice of missions for simulation should be complete in that all possible application scenarios can be formed from this finite set of missions. Three factors are considered in mission setup:

- *Access Point* for UEs connecting to satellites. Two UEs transmit data through the **same** satellite if both UEs are under the coverage area of that satellite, or through **different** satellites otherwise. More energy is consumed for this mission when one satellite transmits two UEs' data.
- *Protocol Level* of flows requiring different satellite resources. Satellite provides different resources to support different types of services. **Emergency** service has the highest priority and thus requires satellites to provide resources such as routing (level three) and switch (level two) in order to guarantee the transmission within required time. **Real-time** service has moderate priority and can use transparent forwarding for part of the signal transmission to save resource usage onboard the satellite. **Non-real-time** service has the lowest priority and all the signals can be transparently forwarded to the GMCC and thus can use least resources onboard.
- *Service Type* supported by different UE capabilities. For bi-directional transmission, narrowband and wideband services are transmitted using **handheld** and **vehicular** terminals, respectively. One-directional transmission has two types. The one transmits data from UEs through satellite network to GMCC is data **collection**. The one transmits data from GMCC through satellite network to UEs is **broadcasting**.

Twelve missions are presented to account for the above three factors and represent different applications. These missions are randomly generated following Poisson distribution with certain arrival rate and input into a LEO satellite network to simulate data quantity transmitted and energy consumed. Each of these twelve missions has multiple flows consecutively executed to implement certain transmission, e.g. user equipment to access satellite or satellite to ground station transmission. Each of these flows is decomposed into one or more time slots in the ETEG, depending on the transmission or processing time of the network operations. Arrival rate is determined following adopted traffic model that takes into account the population covered by LEO satellites and busy hours of certain area [36].

Simulation parameter values are as follows. Output power P_{i-s-su} is set within the range of 20 to 25 dBW. G/T for the first six and the last six missions are set to -27 and -16 dB/K, respectively. Other parameters are set following the results in [13] and [37]. α_{s-ss} is 0.05 W/Mbps. α_{r-us}, α_{r-ss} and μ are all set to 0.01 W/Mbps. γ is set to 1.4.

C. RESULT ANALYSIS

With pre-specified design variable ranges, multi-objective optimization simulation using genetic algorithm is performed to obtain fuzzy Pareto front architecture points. Population size is set to 1000 with Pareto fraction 30% and 200 generations. Non-dominated objective values are returned with corresponding design variable choices.

1) Round-One Results (Two DVs) and Comparison

Round-one simulation with two design variables is carried out and the Pareto front of two-objective tradeoff is illustrated in Fig. 29. Two baseline results using algorithms from literature is also included in the figure. Baseline 1 uses the GreenSR algorithm in [13] which considered only inter-satellite link energy consumptions and the traffic model did not distinguish different traffic types. Baseline 2 uses the OP-GMS method in [14] which allocated power with downlink channel status but did not take energy consumed in the space network into account. The results show that, by considering both user link and inter-satellite links, the model captures comprehensive system energy consumption. This is also interpreted that, at some locations, the Pareto front obtained with various traffic types (Round One) provides better performance for the one with uniform traffic (Baseline 2).

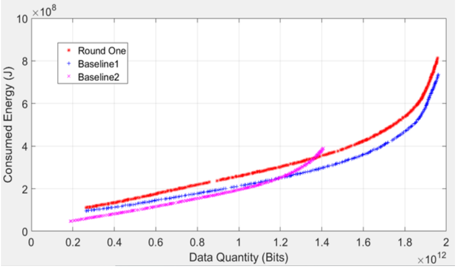


FIGURE 29: Objective Values in Fuzzy Pareto Front (Round One).

What is concerned is the effect of modeling to design variable choices. As illustrated by the round-one points in Fig. 30, most non-dominated points allocate more power in AS2 than in AS1. This is because AS1 executes more inter-satellite transmissions than AS2 which take some portion of the power to be allocated to downlink transmission. This shows the importance of making satellite design choices based on a power model considering all transmission links,

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426

which gives more practical suggestions for system architecture design.

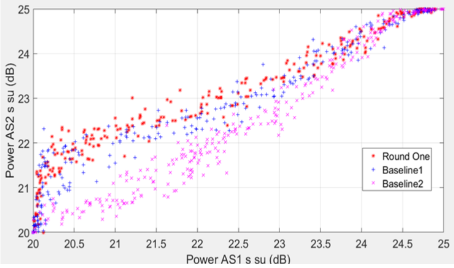


FIGURE 30: Design Variable Values in Fuzzy Pareto Front (Round One).

2) Round-Two Results (Three DVs)

In the second-round simulation, the number of design variables increases from two to three to reflect bandwidth design choices. Distributions of the three design variables for non-dominated architecture points is shown in Fig. 31. Most values of $P_{AS1-s-su}$ are in the 20 to 20.5 dB range while for $P_{AS2-s-su}$, values in 22 to 22.5 and 24.5 to 25 takes more than half of the set. Bandwidth is better chosen as the highest within the range though some with lower values can be considered in band-limited cases. Distributions of the objectives is shown in Fig. 32. It is seen that architecture points with high data quantity and low energy consumptions take most of the non-dominated set.

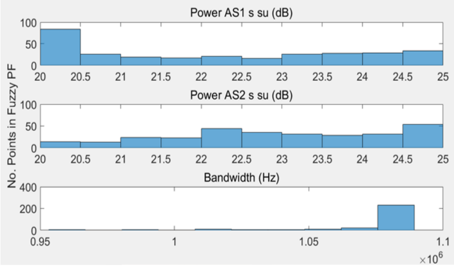


FIGURE 31: Design Variable Distributions for Non-Dominated Points (Round Two).

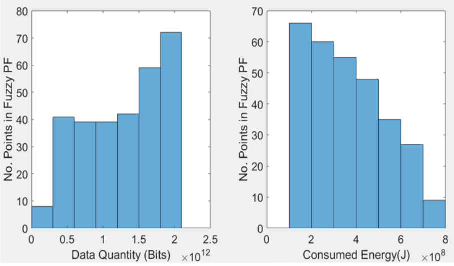
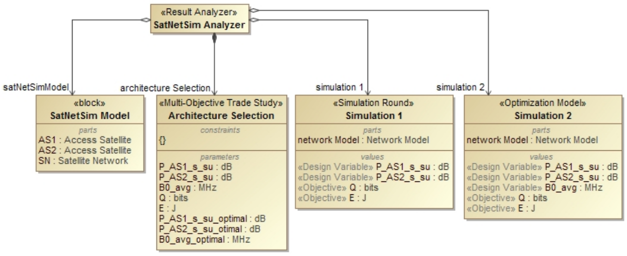


FIGURE 32: Objective Distributions for Non-Dominated Points (Round Two).

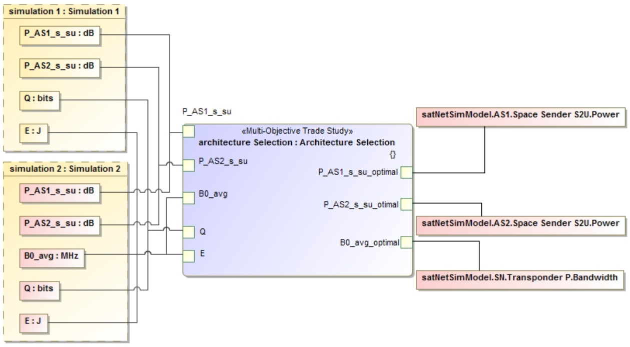
D. SYSML USER MODEL REVISITED: AFTER SIMULATION

1) Analysis User Model

Reusability should be preserved among multiple simulation rounds when more variables are added as the iteration proceeds. After simulation, Pareto set is returned to the SysML modeling software and displayed as an instance table. The part after simulation in analysis user model developed based on the analysis profile is shown in Fig. 33. Pareto set presentation in instance table based on the reusable architecting mechanism for simulation reusability is shown in Fig. 34.



(a) Simulation Analyzer



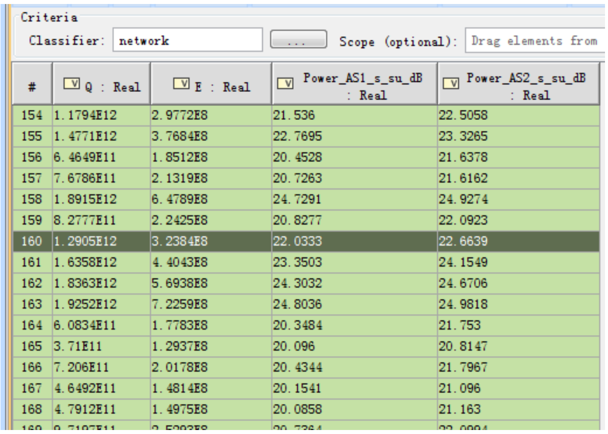
(b) Relations Among Parameters

FIGURE 33: Analysis User Model After Simulation.

2) Architecture User Model

Architecture selection procedure is performed to choose the one with average data quantity among non-dominated points and corresponding energy and design variable values. Then, the reusable mechanism of architecture value updates is carried out to update architecture user model with the selected values. The entire architecture model is shown in Fig. 35. It consists of space segment elements including satellite network routing and switching functionalities modeled in SN, and communication access for UEs to the network modeled in AS1 and AS2. The three design variables in Fig. 35 (red-colored) are updated and other variables can be decided based on them using formulae in the previous section. The variable Bandwidth is the one added in the second-round simulation. In this way, SysML architecture is optimized iteratively among simulation rounds and values are renewed

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3250426



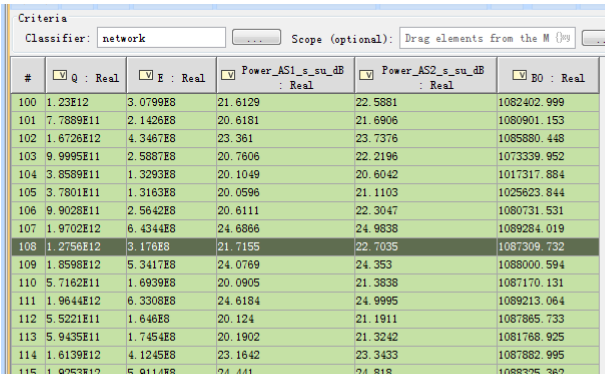
Criteria

Classifier: network

Scope (optional): Drag elements from

#	Q : Real	E : Real	Power_AS1_s_su_dB : Real	Power_AS2_s_su_dB : Real
154	1.1794E12	2.9772E8	21.536	22.5058
155	1.4771E12	3.7684E8	22.7695	23.3265
156	6.4649E11	1.8512E8	20.4528	21.6378
157	7.6786E11	2.1319E8	20.7263	21.6162
158	1.8915E12	6.4789E8	24.7291	24.9274
159	8.2777E11	2.2425E8	20.8277	22.0923
160	1.2905E12	3.2384E8	22.0333	22.6639
161	1.6358E12	4.4043E8	23.3503	24.1549
162	1.8363E12	5.6938E8	24.3032	24.6706
163	1.9252E12	7.2259E8	24.8036	24.9818
164	6.0834E11	1.7783E8	20.3484	21.753
165	3.71E11	1.2937E8	20.096	20.8147
166	7.206E11	2.0178E8	20.4344	21.7967
167	4.6492E11	1.4814E8	20.1541	21.096
168	4.7912E11	1.4975E8	20.0858	21.163

(a) Round One



Criteria

Classifier: network

Scope (optional): Drag elements from the M

#	Q : Real	E : Real	Power_AS1_s_su_dB : Real	Power_AS2_s_su_dB : Real	BO : Real
100	1.23E12	3.0799E8	21.6129	22.5881	1082402.999
101	7.7889E11	2.1426E8	20.6181	21.6906	1080901.153
102	1.6726E12	4.3467E8	23.361	23.7376	1085880.448
103	9.9995E11	2.5887E8	20.7606	22.2196	1073339.952
104	3.8589E11	1.3293E8	20.1049	20.6042	1017317.884
105	3.7801E11	1.3163E8	20.0596	21.1103	1025623.844
106	9.9028E11	2.5642E8	20.6111	22.3047	1080731.531
107	1.9702E12	6.4344E8	24.6866	24.9838	1089284.019
108	1.2756E12	3.176E8	21.7155	22.7035	1087309.732
109	1.8598E12	5.3417E8	24.0769	24.353	1088000.594
110	5.7162E11	1.6939E8	20.0905	21.3838	1087170.131
111	1.9644E12	6.3308E8	24.6184	24.9995	1089213.064
112	5.5221E11	1.646E8	20.124	21.1911	1087865.733
113	5.9435E11	1.7454E8	20.1902	21.3242	1081768.925
114	1.6139E12	4.1245E8	23.1642	23.3433	1087882.995

(b) Round Two

FIGURE 34: Pareto Set Presentation in Instance Table.

to satisfy requirements with reusable and automatic updating mechanism.

IX. MODEL ASSESSMENT AND REQUIREMENT TRACEABILITY

In this section, we present how our proposed framework is validated via comparing with non-reusable models. In addition, the optimized efficiency of the proposed model is calculated through metrics such as viewpoint/ontology coverage and degree of requirement traceability.

A. VIEWPOINT AND ONTOLOGY COVERAGE

This subsection will discuss the assessment of the model framework. In chapter 20 of [4] the authors used patterns as a model assessment tool. A number of patterns were selected and compared with the model by assessing its ontologies and viewpoints. Results were presented as "mapping strength", which are percentage numbers and represents whether the mapping between patterns and models are strong or weak.

This work borrows this idea and extends the assessment method in [4] to a set of metrics. Setting the "Seven Views" Framework and its relevant patterns (including their viewpoints and ontologies) as a modeling standard, the assessment compares the reusable model proposed in this work and the OOSEM model in [3].

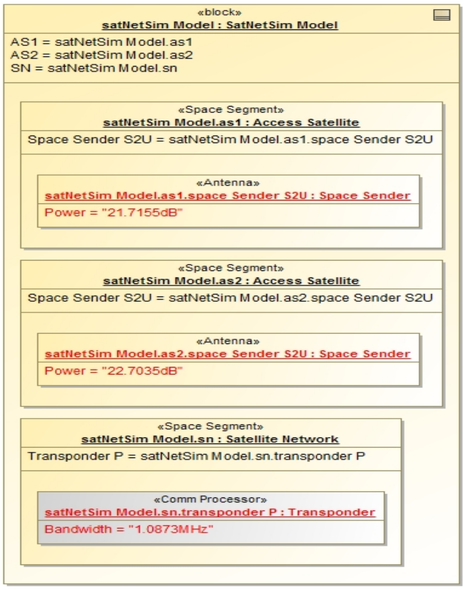


FIGURE 35: Architecture User Model and Value Update (Round 2).

Let the number of viewpoints and ontologies in a pattern framework be N_V and N_O, respectively. User-developed models have viewpoints, ontologies corresponding to the patterns, and other model elements which are separately presented in line with the viewpoints. Model viewpoints are assessed by calculating the percentage of elements in the pattern viewpoints, which is represented by P_{MV}. Each model ontology is assessed by calculating the quantity of ontology elements realized for each pattern ontology, which is represented by Q_{MO}. Besides viewpoints and their ontologies, there are model elements which may be developed before certain reusable framework is established and can also be reused separately. They can be measured by percentage P_{ME} and quantity Q_{ME}, corresponding to viewpoint and ontology, respectively.

Viewpoint coverage C_V and ontology coverage C_O are thus defined as:

$$C_V = \frac{1}{N_V} \cdot \sum_{i=1}^{N_V} [P_{ME}(i) + P_{MV}(i)] \tag{11}$$

$$C_O = \frac{1}{N_O} \cdot \sum_{i=1}^{N_O} [Q_{ME}(i) + Q_{MO}(i)] \tag{12}$$

Viewpoint coverage calculations of the reusable model of this paper and the OOSEM model in [3] are shown in Tab. 8. The column "pattern viewpoint" includes viewpoints required in the "Seven Views" Framework. The quantities are explained as follows.

- Context Identification Viewpoint. The reusable model reuses the concepts in the requirement profile. The OOSEM model has mission requirement to cover this.
- Context Description Viewpoint. Part of the Context Viewpoint in the reusable model is the context description. Mission requirement in the OOSEM model has descriptions to be reused here.
- Element Structure/Description Viewpoint. The reusable model has Structure Viewpoint. The OOSEM model has no reusable design for this one. Same reasons are for their relevant description viewpoints.
- Testing Context Viewpoint. The reusable model has Context Viewpoint (Shared coverage to Context Description Viewpoint above). The OOSEM model has handheld and vehicular but not all the applications or use cases.
- Test Structure, Test Schedule Behavior, and Test Set Behavior Viewpoints. In the reusable model, they are mostly covered by the Content Viewpoint, where the last one is also covered by the Instance Viewpoint. For Test Schedule Behavior, only the effective set was realized in the OOSEM model so a half is counted. For Test Set Behavior, only one fifth behaviors were realized.
- Test Behavior, Test Configuration and Test Record Viewpoints. They are realized as Behavior Viewpoint, Configuration Viewpoints and test results in the reusable model, respectively. The OOSEM model has all the test scenarios to cover these viewpoints so they are all covered as well.

TABLE 8: Viewpoint Coverage.

No.	Pattern Viewpoint (Required) *	Reusable	OOSEM
1	Context Identification	1	1
2	Context Description	0.5	1
3	Element Structure	1	0
4	Element Description	1	0
5	Testing Context	0.5	0.3
6	Test Setup - Structure	0.3	0
7	Test Setup - Schedule Behavior	0.3	0.5
8	Test Setup - Set Behavior	1.3	0.2
9	Test Case - Behavior	1	1
10	Test Case - Configuration	1	1
11	Test Case - Record	1	1

*Pattern viewpoints 1, 2 belong to the context pattern; 3, 4 belong to the description pattern; 5 to 11 belong to the test pattern [4].

Similarly, ontology coverage calculations are presented in Tab. 9. The column “ontology element” is added to represent ontologies required in the “Seven Views” Framework. The coverage numbers mainly represent model element quantities realizing certain ontology elements.

- Context Identification Viewpoint ontologies. The reusable model has application, context and use case to cover its ontology Context. The OOSEM model has mission requirement and use cases.
- Context Description Viewpoint ontologies. All the three ontologies are realized in the reusable model while the OOSEM model has only the Focus.
- Element Structure/Description Viewpoint ontologies. All the three ontologies are realized in the reusable model while the OOSEM model has only the Element. The reusable model has descriptions while the OOSEM model does not.
- Testing Context Viewpoint ontologies. For the testing need, the reusable model has four applications while the OOSEM model has only two (handheld and vehicular).
- Test Set-up Viewpoint ontologies. Under one test schedule, the reusable model has two test sets and five test cases. The OOSEM model has only one test set (functionally corresponding to the effective process set) and two test cases.
- Test Case Viewpoint ontologies. The reusable model has process configuration including function and architecture model libraries. The OOSEM model has logical activities, system context and logical decomposition. They both have sufficient element to cover the patterns.

TABLE 9: Ontology Coverage.

No.	Pattern Viewpoint (Required)	Ontology Element (Required)	R*	O*
1	Context Identification	Context	3	2
2	Context Description	Focus	1	1
3	Context Description	Boundary	1	0
4	Context Description	Environment	1	0
5	Element Structure	Element	1	1
6	Element Structure	Property	1	0
7	Element Structure	Behavior	1	0
8	Element Description	Description	1	0
9	Testing Context	Required System	1	0
10	Testing Context	Testing Need	4	2
11	Testing Context	Testing Boundary	1	0
12	Test Setup - Structure	Test Schedule	1	0
13	Test Setup - Schedule Behavior	Test Set	2	1
14	Test Setup - Set Behavior	Test Case	5	2
15	Test Case - Behavior	Test Behavior	4	4
16	Test Case - Configuration	Test Configuration	2	2
17	Test Case - Configuration	Testable Element	2	2
18	Test Case - Record	Test Record	1	1

*R for reusable model and O for OOSEM model.

Elements in the viewpoint and ontology coverage are related to the quantities in Equations (1) and (2). All the OOSEM model items are counted in P_{ME} and Q_{ME} as they are all developed before the reusable framework is established. For the reusable model, concepts in the Context Identification Viewpoints of the reusable model are counted in P_{ME} as it is developed before the reusable framework. The description and structure viewpoints items are counted in P_{MV} as they are developed under the reusable framework. Items for the last three test-related viewpoints are partly from before the reusable viewpoints so they are counted in both P_{ME} and P_{MV}. Context Identification Viewpoint ontologies are partly counted in Q_{ME} as some mission and use cases are developed before the reusable framework. Test-related ontologies are counted in both Q_{ME} and Q_{MO} for similar reasons as for the viewpoints.

Based on the above analysis, the reusable model and the OOSEM model has viewpoint coverage 0.82 and 0.545, respectively. It can be seen the former covers more than 80 percent of the viewpoints while the latter only more than a half. Similarly, they have ontology coverage of 1.94 and 1.06, respectively. Please note this metric is more about 'quantity' but not 'percentage'. Thus, the results show that the former covers the pattern ontologies in a wider and more frequent way and is thus regarded as better modeling.

B. DEGREE OF REQUIREMENT TRACEABILITY

Traceability metamodel describes how to relate system design results back to the requirements, as illustrated in Fig. 36. Three traceability relationships are defined. Satisfy relationship represents how architecture block design satisfies requirements. Refine relationship describes how use cases refines certain requirements. Verify relationship gives how test cases verify requirements.

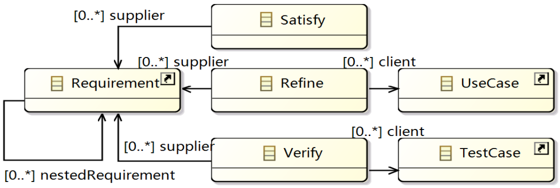


FIGURE 36: Traceability Metamodel.

Generally, architecture design and test procedure contain more system details than use cases. Thus, satisfy and verify relationships are mainly considered, which are presented as satisfy/verify requirement matrix [1].

Satisfy Requirement Matrix consists of:

- Row: items of certain architecture system, which is the client element of the Satisfy dependency. They are developed according to the architecture profile of Fig. 12 and are related to the architecture model library in Fig. 21.
- Column: requirement items that can be the supplier element of the Satisfy dependency. They are related to the requirement profile of Fig. 9.

Verify Requirement matrix consists of:

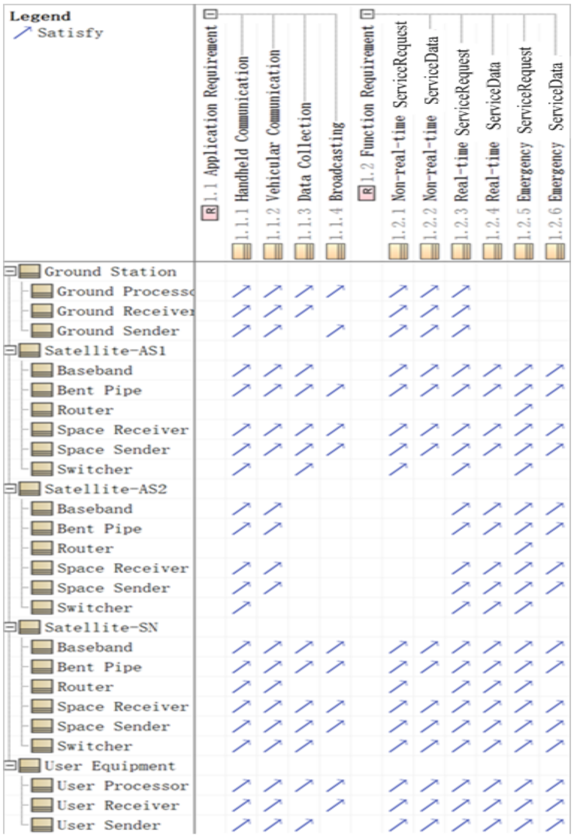
- Row: items of certain test scenario, which is the client element of the Verify dependency. They are developed based on the elements in the function model library of Fig. 20 and corresponds to the process sets of Fig. 18.
- Column: requirement items which can be the supplier element of the Verify dependency. They are consistent with the requirement profile of Fig. 9.

Based on these two matrices, traceability is quantified as follows. Let the number of items of certain architectural system or test scenario be N_I. Let the number of requirements be N_R. For requirement i, let the number of components satisfying/verifying this requirement be N_i. The Degree of Requirement Traceability (DoRT) is defined as

$$DoRT = \frac{1}{N_I} \cdot \sum_{j=1}^{N_R} N_j \tag{13}$$

Requirements are developed based on the requirement profile. The same satellite communication network setup as the case study in VIII-B is applied. Application requirements include handheld (H) and vehicular (V) communication, data collection and broadcasting. Function requirements describe relevant functions of the system including non-real-time (N-RT), real-time (RT) and emergency (E) communications. All these items are consistent with the process cases in Fig. 18 which are related to the function model library of Fig. 20 and are developed based on the function profile of Fig. 10.

Satisfy and verify requirement matrices are presented in Fig. 37 and Fig. 38, respectively. Similar as the case study set-up in previous sections, space segment is divided into satellite network (SN) and access satellite (AS) in Fig. 37. This is to distinguish functioning parts: SN mainly does information routing and AS transmits signals between ground users and the satellite. More on this modeling method can be found in [3].

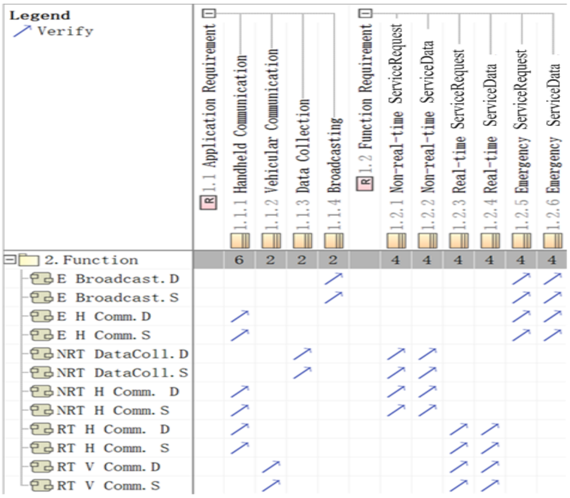


Legend
↗ Satisfy

	R 1.1 Application Requirement	1.1.1 Handheld Communication	1.1.2 Vehicular Communication	1.1.3 Data Collection	1.1.4 Broadcasting	R 1.2 Function Requirement	1.2.1 Non-real-time ServiceRequest	1.2.2 Non-real-time ServiceData	1.2.3 Real-time ServiceRequest	1.2.4 Real-time ServiceData	1.2.5 Emergency ServiceRequest	1.2.6 Emergency ServiceData
Ground Station												
Ground Processo		↗	↗	↗	↗		↗	↗	↗			
Ground Receiver		↗	↗	↗			↗	↗	↗			
Ground Sender		↗	↗		↗		↗	↗	↗			
Satellite-AS1												
Baseband		↗	↗	↗			↗	↗	↗	↗	↗	↗
Bent Pipe		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Router											↗	
Space Receiver		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Space Sender		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Switcher		↗		↗			↗		↗		↗	
Satellite-AS2												
Baseband		↗	↗						↗	↗	↗	↗
Bent Pipe		↗	↗						↗	↗	↗	↗
Router											↗	
Space Receiver		↗	↗						↗	↗	↗	↗
Space Sender		↗	↗						↗	↗	↗	↗
Switcher		↗							↗	↗	↗	
Satellite-SN												
Baseband		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Bent Pipe		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Router		↗	↗				↗		↗	↗	↗	
Space Receiver		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Space Sender		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
Switcher		↗	↗	↗			↗	↗	↗	↗	↗	↗
User Equipment												
User Processor		↗	↗	↗	↗		↗	↗	↗	↗	↗	↗
User Receiver		↗	↗		↗		↗	↗	↗	↗	↗	↗
User Sender		↗	↗	↗			↗	↗	↗	↗	↗	↗

FIGURE 37: Satisfy Requirement Matrix.

Tab. 10 gives the degree of requirement traceability calculation for satisfy and verify relationships. UE represents user equipment and GS is ground station. The architecture



Legend ↗ Verify	R 1.1 Application Requirement	1.1.1 Handheld Communication	1.1.2 Vehicular Communication	1.1.3 Data Collection	1.1.4 Broadcasting	R 1.2 Function Requirement	1.2.1 Non-real-time ServiceRequest	1.2.2 Non-real-time ServiceData	1.2.3 Real-time ServiceRequest	1.2.4 Real-time ServiceData	1.2.5 Emergency ServiceRequest	1.2.6 Emergency ServiceData
2. Function		6	2	2	2		4	4	4	4	4	4
E Broadcast. D					↗						↗	↗
E Broadcast. S					↗						↗	↗
E H Comm. D		↗									↗	↗
E H Comm. S		↗									↗	↗
NRT DataColl. D				↗			↗	↗				
NRT DataColl. S				↗			↗	↗				
NRT H Comm. D		↗					↗	↗				
NRT H Comm. S		↗					↗	↗				
RT H Comm. D		↗							↗	↗		
RT H Comm. S		↗							↗	↗		
RT V Comm. D			↗						↗	↗		
RT V Comm. S			↗						↗	↗		

FIGURE 38: Verify Requirement Matrix.

developed here is consistent with the architecture model library presented in Fig. 21.

TABLE 10: DoRT Calculation.

DoRT for Satisfy		DoRT for Verify	
UE	9.33	Handheld Application Only	Handheld & Other Applications
Sat-AS1	7.5		
Sat-AS2	4.67		
Sat-SN	9.17		
GS	6.33	1.5	3

Considering DoRT for satisfy, the left five numbers in Tab. 10 give the degree of the five parts in the system satisfying requirements. It can be seen that AS1 and SN have higher degrees than AS2. This is because AS1 and SN models implement more functions than AS2, which is only considered in scenarios for different access satellites [35]. UE has the most DoRT value which is reasonable as communication applications are mainly connected to various UE types. GS has a relatively low value as the current requirement set does not raise many on the ground segment.

It should be noted that to calculate DoRT for verify relationship, test scenarios in the rows are closely connected to the communication procedure. Therefore, tests covering more than one requirement should be from the same procedure. For example, the two non-real-time requirements service request and data transmission, shown in Fig. 38 as the two columns of "ServiceReq" and "ServiceData", are from the same communication procedure and thus can be covered with one test case (the first or the second row).

Model reuse is reflected in that different sets of test models can be combined for various requirements. This is shown in the two columns of "DoRT for Verify" in Tab. 10 as an example. The value is 1.5 for handheld applications only and 3.0 for handheld together with other applications. Other combinations can also be applied to represent different mission objectives or user communities.

Remark 1: The maximum value that N_j can take is N_I. Thus, the upper bound DoRT is N_R. Then $DoRT/N_R$ represents the percentage of satisfied or verified relationships in the whole system. However, the separate calculations in Tab. 10 is more practical as they reflects the measurement of each item design relating to requirements. This will be valuable as the system is extended or decomposed into subsequent levels.

X. CONCLUSIONS

Reusability for Practical Concerns. When the proposed framework is used in practice, it can be connected with models already developed using certain MBSE methodologies. As shown in Fig. 1, part of the user models have already been developed based on the OOSEM precedure. Some of the requirement, function and architecture models are already been developed and presented using SysML diagrams. When modelers build reusable models using the framework presented in this paper, they should firstly make sure the profiles contain ontologies in the specific domain, such that the legacy models are semantically within the scope of the reusable model. Then, the framework of viewpoints can be built based on Sec. V of this paper. They serve as a complete framework that a system model development procedure may include. When model libraries are built based on the patterns, older models can be classified into their related areas, such as requirements, functions or architectures. In this way, when users develop their own models, they will have a number of readily available models from the library and can also insert any new models as they like. In addition, the added models may not be presented in built-in SysML diagrams as the viewpoint mechanism provides flexibility in expressing ideas and outputs in their own way.

In conclusion, A SysML-based modeling and simulation approach for satellite network is presented in this paper. Reusability, reconfigurability and efficiency are seen as three levels of modeling and simulation aims. "Reusability" achieves the basic modeling and simulation. It is realized with profiles, model libraries and simulation interface. "Reconfigurability" is the concrete, detailed modeling and simulation. Based on "reusability" modeling, it is realized via reconfiguration of parameter features in metamodels and profiles and pattern viewpoints. "Efficiency" is concerned with practical implementation of large-scale framework. This can be realized via the extension mechanism of the framework and the design-analysis optimization. Our future plan will be extending the model to other domains and to subsequent manufacturing and operational stages such that the reusable modeling framework can support more domains and contribute to efficient product development.

REFERENCES

[1] S. Friedenthal, A. Moore and R. Steiner, *A Practical Guide to SysML: the Systems Modeling Language*, 2nd ed., Elsevier Inc., 2012.
[2] D. Steinberg, F. Budinsky, M. Paternostro and E. Merks, *EMF Eclipse Modeling Framework*, Pearson Education Inc., Boston, 2009.
[3] S. Gao, W. Cao, L. Fan and J. Liu, "MBSE for satellite communication

system architecting," *IEEE Access*, vol. 7, no. 1, pp. 164051-164067, Dec. 2019.

[4] J. Holt, S. Perry and M. Brownsword, *Foundations for Model-based Systems Engineering: from Patterns to Models*, The Institution of Engineering and Technology, London, 2016.

[5] A. Cicchetti, F. Ciccozzi and A. Pierantonio, "Multi-view approaches for software and system modeling: a systematic literature review," *Software and Systems Modeling*, vol. 18, pp. 3207-3233, 2019.

[6] O. Batarseh and L. F. McGinnis, "System modeling in SysML and system analysis in ARENA," *Proc. of the Winter Sim. Conf. (WSC)*, pp. 258:1-258:12, 2012.

[7] *SysML-Modelica Transformation*, Object Management Group, http://www.omg.org/spec/SyM/1.0/PDF, Nov. 2012.

[8] G.-D. Kapos, V. Dalakas, M. Nikolaidou and D. Anagnostopoulos, "An integrated framework for automated simulation of SysML models using DEVS," *Simulation*, vol. 90, no. 6, pp. 717-744, 2014.

[9] P. Leserf, P. Saqui-Sannes and J. Hugues, "Trade-off analysis for SysML models using decision points and CSPs," Software and Systems Modeling, http://doi.org/10.1007/s10270-019-00717-0, 2019.

[10] M. Friedl, A. Kellner and L. Weingartner, "Integration of domain-specific simulation models into descriptive system models by using SysML" *IEEE International Systems Engineering Symposium (ISSE)*, Vienna, Oct. 2017.

[11] A. I. Aravanis, B. Shankar M. R., P. Arapoglou, G. Danoy, P. G. Cottis and B. Ottersten, "Power Allocation in Multibeam Satellite Systems: a Two-Stage Multi-Objective Optimization," in *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3171-3182, Jun. 2015.

[12] D. Zhou, M. Sheng, X. Wang, C. Xu, R. Liu and J. Li, "Mission Aware Contact Plan Design in Resource-Limited Small Satellite Networks," in *IEEE Transactions on Communications*, vol. 65, no. 6, pp. 2451-2466, Jun. 2017.

[13] Y. Yang, M. Xu, D. Wang and Y. Wang, "Towards Energy-Efficient Routing in Satellite Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3869-3886, Dec. 2016.

[14] D. Zhou, M. Sheng, R. Liu, Y. Wang and J. Li, "Channel-Aware Mission Scheduling in Broadband Data Relay Satellite Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1052-1064, May 2018.

[15] *Software Communications Architecture Specification*, Joint Tactical Radio System Office, JTRS-5000SCA, version 2.2.2, 2006.

[16] J. Bard and V. J. Kovarik Jr., *Software Defined Radio: The Software Communications Architecture*, Wiley & Sons, 2007.

[17] T. Rittenbach, V. Kovarik, R. Krause-Aiguer and C. Stewart, "Complex terminal system design," *IEEE Military Comms. Conf. (MILCOM)*, San Jose, Oct. 2010.

[18] V. J. Kovarik Jr. and R. Muralidharan, "Integrated communications systems modeling: a model-based engineering approach," *WInnComm*, 2014.

[19] *Unified Modeling Language (UML)*, Object Management Group, 1997.

[20] *OMG Systems Modeling Language*, v1.5, Object Management Group, 2017.

[21] V. J. Kovarik Jr and R. Muralidharan, "Model-based system engineering: lessons learned from the joint tactical radio system," *J. Sign. Process. Syst.*, vol. 89, pp. 97-106, 2017.

[22] A. Abid, M. Hammadi, M. Barkallah, J. Y. Choley, J. Louati and A. Riviere, et al., "Generic framework for holonic modelling and multi-agent based verification of reconfigurable manufacturing systems,". *International Journal of Precision Engineering and Manufacturing*, vol. 19, no. 12, pp. 1793-1809, Dec. 2018.

[23] J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal*, vol. 26, 1987.

[24] A. Vallecillo, et al., "RM-ODP: The ISO Reference Model for Open Distributed Processing," *Softw. Eng.*, vol. 3, pp. 66-69, 2001.

[25] C. Atkinson, D. Stoll and P. Bostan, "Supporting View-Based Development through Orthographic Software Modeling," *Proceedings of the 4th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, pp. 71-86, Milan, Italy, May 2009.

[26] M. Nikolaidou, G. Kapos, A. Tsadimas, V. Dalakas and D. Anagnostopoulos, "Simulating SysML models: Overview and challenges," *10th System of Systems Engineering Conference (SoSE)*, San Antonio, TX, pp. 328-333, May 2015.

[27] M. Bajaj, D. Zwemer, R. Peak, A. Phung, A. G. Scott and M. Wilson, "SLIM: collaborative model-based systems engineering workspace for next-generation complex systems," *Aerospace Conference*, Big Sky, MT, pp. 1-15, Mar. 2011.

[28] R. Liu, M. Sheng, C. Xu, J. Li, X. Wang and D. Zhou, "Antenna slewing time aware mission scheduling in space networks," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 516-519, Mar. 2017.

[29] C. Joo and J. P. Choi, "Dynamic cross-layer transmission control for station-assisted satellite networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1737-1747, Jul. 2015.

[30] R. Liu, M. Sheng, K.-S. Lui, X. Wang, Y. Wang, and D. Zhou, "An analytical framework for resource-limited small satellite networks," *IEEE Commun. Lett.*, vol. 20, no. 2, pp. 388-391, Feb. 2016.

[31] *The OSI model's seven layers defined and functions explained*, Microsoft Support, 2015.

[32] E. G. Talbi, *Metaheuristics From Design to Implementation*, Wiley, Hoboken, NJ, USA, 2009.

[33] N. Font, C. Blosse, P. Lautier, A. Barthere and P. Voisin, "Flexible payloads for telecommunication satellites - a Thales perspective," in *Proc. 32nd AIAA Int. Commun. Satellite Syst. Conf.*, San Diego, CA, 4-7 Aug 2014.

[34] G. Maral and M. Bousquet, *Satellite Communications Systems: Systems, Techniques and Technology*, Wiley, Hoboken, NJ, USA, 2009.

[35] S. Gao and Z. Fang, "Multilayer feature-rich satellite network analysis: an application-oriented and time-evolving approach," *IEEE Trans. Network Science and Engineering*, vol. 8, no. 2, pp. 1316-1327, Apr.-Jun. 2021.

[36] X. Alberti, J. M. Cebrian, A. D. Bianco, Z. Katona and N. Alagha, "System capacity optimization in time and frequency for multibeam multi-media satellite systems," *Advanced Satellite Multimedia Systems Conference (ASMA) and the 11th Signal Processing for Space Communications Workshop (SPSC)*, Cagliari, Italy, pp. 226-233, Sep. 2010.

[37] Y. R. Tian, X. C. Lu and F. J. Huang, "Design and performance analysis of inter-satellite link in multilayer satellite network," (in Chinese), *J. Time Freq.*, vol. 33, no. 2, pp. 140-145, 2010.

SU GAO received the MSc and PhD degrees in telecommunication engineering from Imperial College London, UK. He is currently a senior research engineer at the Institute of Telecommunication and Navigation Satellites, China Academy of Space Technology (CAST). His research interests include satellite communication, system architecting and machine learning.

YUE CAO received the M.Sc. and Ph.D. degrees in Computer Science from Zhejiang University, Hangzhou, China. She is currently a Lecturer with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. Her research interests include model-driven mechatronic system design and model-based systems engineering.

YINQIAO LI received the Ph.D. degree in information engineering from Beijing Institute of Technology. He is currently a chief designer and research fellow in China Academy of Space Technology. His research interests include satellite and payload, next generation mobile and high-throughput satellite communication.

YUJUN CHEN currently serves as a principal investigator of model-based spacecraft system design in China Academy of Space Technology. His research interests include SysML-related model design and multidisciplinary optimization design.

SONG JIN received the Ph.D. degree in microwave engineering from Beijing Institute of Technology. He is currently a senior satellite designer at the Institute of Telecommunication and Navigation Satellites, China Academy of Space Technology. His research interests include satellite communication and payload design, resource allocation and management for the next generation satellite network.

SHIKUN XIE received the M.S. degree from Tsinghua University, Beijing, China. He is a visiting scholar at Nanjing University of Aeronautics and Astronautics and a Chief Engineer with Beijing iFlight Technologies co., LTD. His research interests include architecture design and multidisciplinary design optimization.

JIHONG LIU is a Professor in the School of Mechanical Engineering and Automation at Beihang University. He received his PhD degree in mechanical engineering from Tokyo Metropolitan University, Japan in 1996. He has published more than 150 journal and conference papers. His research interests include model-based systems engineering, complex product engineering, knowledge management and knowledge engineering, and artificial intelligence in design.

• • •