Architecting commercial aircraft with a Domain Specific Language extended from SysML

Haomin Li*, Chao Zhan1, Haomin Wu1, Miao Yu1, Jie Dai2 and Wenbin Zou2

1 Shanghai Aircraft Design and Research Institute, Shanghai, 201210, China
2 VisionMC, Shanghai, 201203, China
*Corresponding author’s e-mail: lihaomin@comac.cc

Abstract. Architecting commercial aircraft has been always challenging due to the complexity of the product. A Model-Based Systems Engineering (MBSE) method was employed in this paper to handle this challenge, and a widely used general-purpose systems modeling language - SysML - was adopted to support the architecting work. To ensure compatibility between the modeling language and the specific aircraft architecting methodology, we explored construction and application methods of a Domain Specific Language. A new Domain Specific Language was created, in which several extensions to SysML were made, including Function & Components, Port, Exchanged Item and Relation. Benefiting from this new Domain Specific Language, an overall aircraft architecture model was precisely constructed.

1. Introduction
As one of the most complex industrial products on the planet, the design of commercial aircraft has always been a challenging task. Systems engineering is an effective means to deal with this complex challenge, which includes a series of technology and management processes [1]. Among them, aircraft architecture design is an important content in the early stage. In traditional design methods, the results of aircraft architecture design are often described and communicated through documents. Due to the random nature of natural language description, it often causes misinterpretation during information transmission; when the design changes, the designer needs to manually modify the corresponding document content, which often leads to incomplete modification and inconsistency. The complexity of commercial aircraft makes these problems even worse.

In order to cope with the deficiencies of the above design methods, a new technology called “Model-Based Systems Engineering” [2] (MBSE) emerged. MBSE uses "models" to describe the design. These models can be understood, analysed and transmitted by computers. To ensure consistency, Object Management Group (OMG) released a new modeling language specifically for systems engineering - SysML [3,4], which is used to standardize the description of system design. SysML is a general-purpose description language, it can be and has been widely used in the design of various complex systems, including aviation, aerospace, automobiles, etc.

Although SysML provides a general expression of system design, when we need to accurately describe some specific concepts in a specific field, SysML is often not competent [5]. For example, in SysML, "Block" is a general modular unit of structure, it can be used to represent a system, a sub-system, a component, a software module, an abstract or concrete entity, etc. But in some cases, we need to distinguish between these concepts. If one concept is used to describe all types of information, it could be confusing for designers, especially those who are not familiar with the SysML language.
Therefore, in order to more accurately describe and distinguish the concepts, it is often necessary to expand SysML for special needs in a specific field. SysML provides a “Profile-Based” mechanism to realize the concept extension[6]. In this article, we utilized this mechanism to build a new Domain Specific Language (DSL), in which concepts of commercial aircraft architecture design are included. On the basis of this new DSL, we carried out aircraft architecture modeling.

2. Modeling methodology and the domain specific language
Methodology is an essential part of MBSE practice. It provides processes and methods that an enterprise or organization needs to follow when designing a system. There are a variety of MBSE methodologies, each with a different focus and scope of application[7], such as INCOSE Object-Oriented System Engineering Method (OOSEM), IBM Rational Telelogic Harmony-SE, JPL State Analysis, Dori Object-Process Methodology (OPM)[8-12], etc. In the selection process of methodology, the most basic principle is that the methodology selected must be able to cover the design concepts and processes in the field. The current publicly released methodologies often need to consider versatility, leading to the need to adjust the general content when applied in specific fields.

After choosing a specific modeling methodology, the concepts in that methodology need to be implemented on modeling software, that is, concept extensions based on the standard modeling language.

In the following, we specifically describe the methodology used in commercial aircraft architecture modeling, the standard System modeling language – SysML, and the conceptual extension of SysML language to support the implementation of this methodology.

2.1. Methodology to support aircraft architecting
In the commercial aircraft architecture modeling methodology, according to the stages of aircraft design, we divide the architecture into four levels: Functional Architecture, Logical Architecture, Logical Redundancy Architecture, and Physical Architecture.

The Functional Architecture focuses on describing the functional composition of the aircraft, the decomposition relationship between the functions, the interface (i.e., port) connection relationship between the functions, and the functional exchanged items transmitted between the functional interfaces - matter, energy, and signals.

Logical Architecture is the realization of "Functions", but at a relatively "abstract" level. The Logical Architecture focuses on describing the logical composition of the aircraft, including systems, subsystems, equipment, etc.; the interface connection between logical components, and the logical exchanged items transferred between logical interfaces.

Logical Redundancy Architecture adds safety and reliability considerations to the Logic Architecture, and adds redundancy design components.

The Physical Architecture describes the realization of the Logical Architecture in the real world, and also includes the physical composition, the interface connection between physical components, and the physical exchanged items passed between the physical interfaces.

In summary, in this methodology, three types of concepts are used to describe the architecture:

- **Function and components** (logical components, logical redundant components, physical components)
- **Exchanged item**: matter, signal, energy
- **Port**: used to transfer exchanged items

2.2. System modeling language - SysML

2.2.1. Methodology concepts in standard SysML
SysML is a general-purpose modeling language that provides various modeling elements required to describe requirements, structure, parameters, and behaviour. Elements to express the three types of concepts required for architecture modeling are described briefly in the following.
- **Function and components**: SysML use Block to represent all types of function and components, including "abstract" (Function and Logical Components) and "concrete" (Physical Components).
- **Exchanged item**: The exchanged items are divided into three categories: matter, energy, and signal. In SysML, matter and energy are represented by Block, and signal is represented by Signal.
- **Port**: There're two types of ports in SysML - Proxy Port or Full Port. And the features of a port are captured and expressed through Interface Block (for Proxy Port) or Block (for Full Port) respectively. By binding Interface Block andproxy port (Block and full port), the port characteristic information can be expressed.

2.2.2. Profile-based extension mechanism
SysML provides a profile-based extension mechanism. In profile, modelers can extend the modeling concept by creating an element called "Stereotype"[4].

SysML extended by this profile has the ability to support the modeling of a specific "concept" in that domain. Thus, in this case, the language is no longer a standard SysML, but has become a domain-specific language.

2.3. Domain Specific Language for aircraft architecting – an extension to SysML
As mentioned above, in the aircraft architecture modeling methodology, we divide the architecture into four levels: Functional Architecture, Logical Architecture, Logical Redundancy Architecture, and Physical Architecture; and introduce three types of concepts to describe the architecture: function and components, exchanged item, port. In SysML, these three types of concepts are represented by Block, Signal, and Interface Block.

In order to describe the concepts in the methodology more accurately, we have extended the above three types of concepts in SysML. Moreover, in order to define or clarify the different parameters or characteristics of four levels, we created some attributes and relation for extended elements according to the actual design processes of commercial aircraft. This is also the top-level uniform rules of the aircraft architecture design. And we can also evaluate the architecture design results by checking the consistency of these attributes.

2.3.1. Function & components
In SysML, Function and Components are all represented by "Block". Here, we extend 4 Stereotypes based on Block – Function, Logical Component, Logical Component with R, Physical Component (see figure 1), which are used to represent the modular unit in four different architectures.

In addition, according to the design processes of commercial aircraft, we also define some attributes for Function, Logical Component, Logical Component with R and Physical Component, including ID, description, validation status, etc. These attributes represent the identification, design state and feature of different layer element.
2.3.2. Exchanged items
The exchanged items are divided into three categories: matter, energy, and signal. In SysML, matter and energy are represented by "Block", and signal is represented by "Signal". Based on these two elements, we extend nine new Stereotype. Each layer contains three types of exchanged items (the Logical Architecture and the Logical Redundancy Architecture use the same Stereotype), and the prefix of the corresponding level (i.e., Functional, Logical, Physical) is added before the name of the Stereotype (see figure 2).

According to three categories of properties, some attributes are defined, including general attributes and category attributes. Among them, general attributes include source, description and so on. And category properties are distinguished by the type to which they belong. For example, the exchanged item attributes of signal class include precision, range, delay, etc. The exchanged item attributes of matter class include pressure, temperature, etc. The exchanged item attributes of power class include power, voltage, etc. Moreover, the attributes of the same class are different among four layer.

Figure 2. Exchanged Items extension.

2.3.3. Ports
The port feature is captured and represented by "Interface Block" in SysML. In our practice, the ports are classified according to the level and the type of exchanged items transmitted. Based on "Interface Block", nine Stereotype of port types are constructed (the logical architecture and logical redundancy architecture use the same Port type Stereotype). A prefix of the corresponding level is also added before the name of each Stereotype (see figure 3).
2.3.4. Relation

In the real world, software needs to exist in hardware. To represent this kind of dependency, we created a new relation - "Hosted by". "Hosted by" can be used to connect a "Software" and a "Hardware", which means the "Software" physically exists in that "Hardware" (see figure 4).

3. Architecting aircraft using the domain specific language

This section describes the four-layer aircraft architecture model based on the domain-specific language.

First, we created the Functions/Components in each layer, as well as corresponding ports for these elements. Next, we created the exchanged items in each layer. By supplementing the decomposition relationship, port connections, and the exchanged items passed between ports, the aircraft architecture model is completed. In addition, at the physical architecture layer, we also use the "Hosted By" relationship to express the dependency between software and hardware.

3.1. Function, components, exchanged item, and port

Figure 5 takes Function and Logic Component as examples to show the application of Function, Component, Exchanged Item and Port in DSL. The "Decelerate aircraft using Wheel Braking" function includes a port named "F_Power_in", and the port type (after the colon) is expressed by the "Functional Power Proxy Port" with the same name on the right. The flow property "F_Hydraulic Power" with the direction of "in" indicates that an exchanged item "F_Hydraulic Power" will flow in through this port. The stereotype of the exchanged item is "Functional Power Item", which is defined by DSL. Similarly, the three elements below indicate that the logical component named "L_Wheel Brake System" contains a port named "L_Signal_in", the port type is "Logical Proxy Port" with the same name on the right, and a Logical Signal Item named "L_WB Control Signal" is allowed to flow in via this port.

Usage of Logical Redundancy Component and Physical Component are the same as Logical Component, which will not be repeated here.
3.2. Aircraft architecture

Based on the created Function, Component, Exchanged Item and Port, the aircraft architecture can be built. Figures 6 and 7 take the logical architecture as an example, respectively showing the decomposition and interface (port) connection of the aircraft logical architecture.

Figure 6 describes several components of an aircraft, including Airframe, Wheel Brake System, Wheels, Landing Gear, Hydraulic System, Flight Control System. The prefix "L_" of each Logical Component name is used to indicate that the component belongs to the logical architecture level. Similarly, the component names of functional architecture, logical redundancy architecture, and physical architecture will use the prefixes "F_", "LR_", and "P_".

Figure 7 shows the port connection of the above-mentioned logical components and the exchanged items passed between the ports. The direction of the arrow in the figure represents the direction in which the exchanged item is passed. To keep the diagram concise, the port names and exchanged item names in the diagram are hidden, and only the exchanged item "L_WB Control Signal" is shown as an example.

3.3. Software hosted by hardware

In the Domain Specific Language, we created a new relationship "Hosted By" to express the dependency of software on hardware. Figure 8 shows the application of this relationship in model. Because we only
consider the specific implementation method (software or hardware) at the physical layer, the "Hosted By" relationship is only used to connect Physical Component.

![Figure 8. Software hosted by Hardware.](image)

4. Conclusion
This paper explored the construction and application methods of a Domain Specific Language in the field of aircraft architecture modeling, and achieved accurate modeling of aircraft architecture. Through the analysis of specific domain methodology concepts, the conceptual elements that need to be expanded are determined. Then the Domain Specific Language construction is realized with the help of the language expansion mechanism. The construction pattern of domain-specific language here is also applicable to architecture design in other domains, and the experience explored in this article can be used as a reference in other fields as well.

References
[1] Jackson, S. (2016) Systems Engineering for Commercial Aircraft. Routledge, Surrey.
[2] Wymore, A. W. (2018) Model-Based Systems Engineering. CRC press, Boca Raton.
[3] OMG (2019) OMG System Modeling Language. https://www.omg.org/spec/SysML/1.6/PDF.
[4] Friedenthal, S., Moore, A., Steiner, R. (2008) Omg systems modeling language (omg sysml) tutorial. INCOSE International Symposium, 18(1), 1731-1862.
[5] Weilkiens, T. (2007) SysML-The Systems Modeling Language. Systems Engineering with SysML/UML. Elsevier Inc.
[6] Friedenthal, S., Moore, A., Steiner, R. (2014) A practical guide to SysML: the systems modeling language. Morgan Kaufmann, California.
[7] Estefan, J. A. (2007) Survey of model-based systems engineering (MBSE) methodologies. INCOSE MBSE Focus Group, 25(8), 1-12.
[8] Fusaro, R., Ferretto, D., Viola, N. (2016) Model-Based Object-Oriented systems engineering methodology for the conceptual design of a hypersonic transportation system. IEEE International Symposium on Systems Engineering. IEEE.
[9] Authors, S. (2013) A survey of model-based systems engineering (MBSE) methodologies.
[10] Rao, B. H., Padmaja, K., Gurulingam, P. (2013) A brief view of model based systems engineering methodologies. International Journal of Engineering Trends & Technology, 4(8).
[11] Authors, S. (2013). INCOSE systems engineering vision 2020.
[12] Dori, D. (2002) Object-Process Methodology-A Holistic Systems Paradigm. Springer Berlin Heidelberg.