ICAS 2020 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

Co-simulation of SysML and Simulink/Modelica Using FMI

Shen Jieshi, Zhou Qing, Li Bingfei, Chen Cong

Chinese Aeronautical Radio and Electronics Research Institute

Abstract

As the most commonly used modeling language for MBSE, SysML is often used in the conceptual design phase in the V model of the product lifecycle. At present, we have done a lot of mature research on the modeling method of SysML, but in the process from conceptual design to unit design, SysML model is often used as the specification of system function and architecture, which means SysML plays very little role in the unit design process.

In this paper, SysML model is Compiled into a generic model format as *.fmu* through FMI (functional mock-up interface). FMI is a tool independent standard proposed a solution for model integration with intermediate files. In this mode, the SysML model can integrate with other forms of models such as Simulink and Modelica, which play a role in unit design. With transferring a runnable SysML model, on the one hand, the role of SysML in product lifecycle is extended. It achieves the purpose of reusing the model. On the other hand, the FMU model from SysML can help to carry out Co-simulation ahead of time in unit design and implement the advance of the testing process.

This paper also studies the mapping relationship between SysML model and FMU model encapsulated by FMI. We also present a situation that SysML model participates in unit design and unit-level test through a typical case, which tests the electrical system status of cruise missile in the process of attack. The paper provides an effective and practicable method for using concept models efficiently.

Keywords: SysML, FMI, Co-simulation, Modelica

1. OVERVIEW

Currently, engineers make an effort to solve problem about Problem Domain and Solution Domain by using SysML to build models for complex system in Aeronautical and Astronautical engineering, which have already made progress. SysML model is expert in analyzing and organizing requirement of system and helps integrator express requirement of subsystem design to subcontractor precisely. However, SysML model itself can hardly play an important role in the following design. There are two main reasons. The first one is that SysML model for complex system would be huge and leads to high cost on learning the original model for engineers who should receive and utilize the model. To some extent, someone on border of the design only needs to focus on the inputs and outputs of subsystem, rather than the whole internal condition of subsystem. The other reason is that communication methods of model files limits the migration usage of SysML model on toolchain, so user should consider the implement ability of unit-design modeling tool on SysML model. Functional Mock-up Interface (FMI) realizes communication between multi-source heterogeneous models by exporting models produced by different modeling tools such as Rhapsody, Simulink and SimulationX to uniform format. FMU model becomes a bridge between conceptual design and unit design in system design, which changes the method in regular design using requirement and document as system information convention to a programmatic and trackable method. Indeed, tools and techniques are like the bricks of a house, whether and how to use them should be based on the consideration of structure of the house. Therefore, before using FMI protocol in MBSE, an appropriate MBSE method process should be designed on the basis of requirements and purposes of engineering project, and implement Life-circle Management on engineering elements and techniques.

2. SYSTEM DESIGN PROCESS BASED ON MODEL

To decide the model-driven system design process, the first step is to distinguish the purpose of work is to proceed Problem Domain or Solution Domain. Then, we should choose modeling method corresponding to the aim of modeling. Modeling for Problem Domain is always to capture and verify requirement. The modeling process using SysML language is to input coarse upper level demand. Through analyzing application scenes and activities, more detailed requirements are organized to obtain requirement items which could be used for system design in practice directly. For Solution Domain process, top-level design of system structure and function is carried out according to detailed requirement items to verify requirement and define system prototype. Modelling design of Problem Domain and Solution Domain is relative, we can repeat the process of decomposing requirement and verifying on the basis of complexity of system. The work of the paper is focusing on Solution Domain. It is because work in Solution Domain helps us to obtain primary conceptual design, which allows usage of more professional modeling language and tools to realize unit design on sub modules in system. In the process, there are two problems. One of the problems is whether the value of SysML will reduce because of the changes on layers, language and tools of modeling. The other one is that considering production relations, subcontractors are responsible for unit design, but integrator will rarely provide full system solutions. Therefore, because subcontractor finish unit design only based on requirement and lack methods to verify feasibility associating with integrator in advance, errors are produced in system integration process and consistency of system design cannot guarantee.

Aiming at solving two problems mentioned above, we establish the process method on building, transferring SysML model and joint validation. The method could maximum the value of model and provide restraints on system subcontractor as many as possible to reduce the problem in system integration process.

The system design process based on model is shown in the figure below. Referring to V-model, left part of the product lifecycle is design process, at the bottom is manufacturing process and the right part is testing and integration process. Our work focuses on conceptual and unit design stages and the connection process.

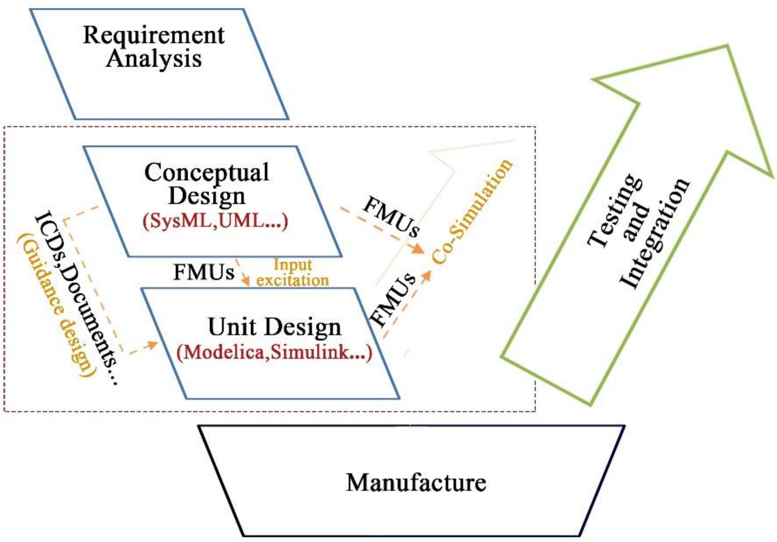


Figure 1 Models transfer in system design process.

The responsibilities and connection forms of each stage are:

- **Requirment analysis:** Rely on certain requirements engineering methods to decompose the needs of stakeholders and obtain functional requirements and non-functional constraints, which can directly guide the system design.
- **Conceptual design:** Use SysML/UML or other modeling language to organize the requirements of the system, preliminary construct a system structure and system function set which can meet the requirements. Realize the validation of requirements and the distribution of requirements to

structure components.

- **Unit design:** According to the subsystem's requirement input, function description and interface definition, the professional discipline modeling language is used to realize the subsystem design. Common unit design modeling languages including Simulink and Modelica are used to design and verify the simulation on the electromagnetic, mechanical, control, and thermal aspects of the system.
- **Conceptual design to input of unit design:** The conceptual design process mainly produces a design framework, ICD, requirements documents for the entire system and a visual SysML model. Due to the closure of the tool, SysML mainly uses its graphical description methods to provide a supplementary description of the system design framework and the requirements document when passing down. We hope to keep the logical features of the SysML model to be used as test cases. Those tests cases could be applied as incentive input of the unit design model to drive the advancement of the simulation validation process.
- **Integrated validation process in unit design stage:** The model-based method pre-produces the prototype of the system in the form of a virtual prototype. In this way, the integration test of the system before manufacturing can confirm the satisfaction degree of the system, coordinate the interface coupling problem and expose problems, reducing the risk and cost of the following implementation.

3. MODEL INTEGRATION SCHEME BASED ON FMI PROTOCOL

In order to solve the integration problem of interdisciplinary modeling tools, the European Development Information Technology Program (ITEA2) proposed the MODELISAR project, the vision is to create virtual products assembled by a set of models. Each model represents the combination of components, each model follows physical laws and is control system model digitally assembled. These models are connected by a common interface to cover the modeling, simulation, validation, and testing activities of the system. The interface standard is finally the FMI (Functional Mock-up Interface) standard.

The FMI interface supports the way in which the results of modeling tools are exported as the intermediate file FMU, which saves costs on the plug-in/interface development between tools. The figure below shows the logical structure of an FMU model unit. The data information that the model unit can receive from the simulation environment includes the real values, integers, Booleans, string parameters, and initial values of the simulation time for calibration, in addition to various input variables. The data content that the model unit feeds back to the simulation environment includes all internal variables, states, and output variables that modelers are allowed to expose.

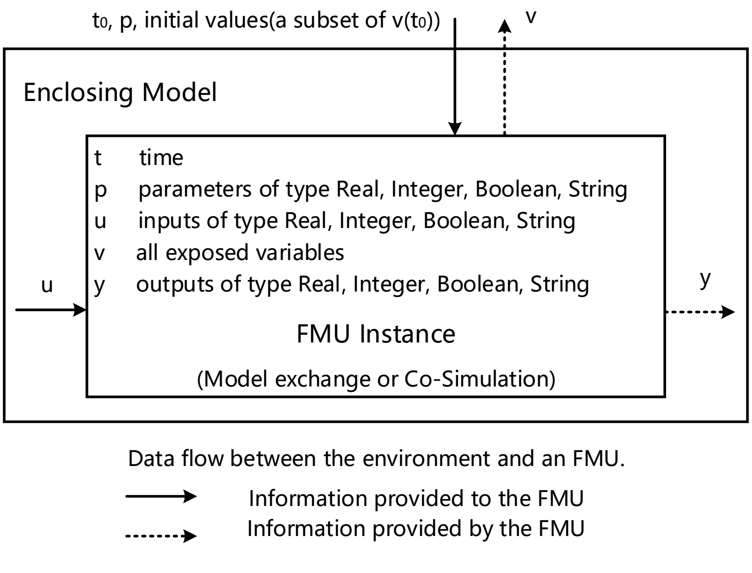


Figure 2 logical structure of an FMU.

From the perspective of the FMI protocol, the digitized model is a series of mappings of logical relationships or mathematical relationships between inputs and outputs. Therefore, all functional, logical and mathematical descriptions of the model are encapsulated in the form of differential equations, and the relevant parameters are initially set for solving calibration of the differential equations. Using FMU form as a model encapsulation method can solve the monopoly situation of the previous modelling tools on the solver. In addition, because FMU model is a black box model form, the system design can be kept confidential before the product is put into production, which effectively solve the Interest disputes between subcontractors and integrators.

Based on the characteristics of the FMU model, this paper offers two application forms in the model-based system engineering process.

1) SysML can be used as a constraint model of the unit test model to guide the design of the unit design level (such as mathematical model, physical model, structural model, etc.) and as a driven model (test module) on modelling platform for the unit design model to provide signal and data input and output for the original model built by other formal languages;

2) After subcontractors complete the design of the sub-system unit, the model of the unit design level is encapsulated in the FMU model and then provided to the integrator to perform the virtual prototype test of the whole machine, which is to verify whether the overall design conforms. demand. This paper presents case design for this application form.

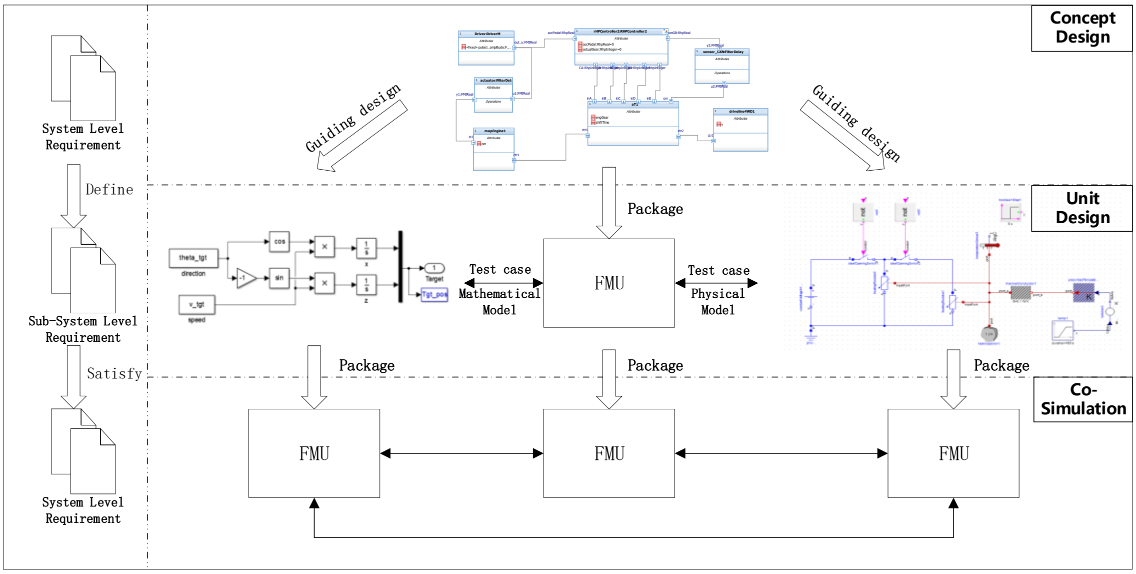


Figure 3 Application form of FMU model in model-based design.

4. CASE: MISSILE ELECTRICAL SYSTEM SIMULATION

4.1 Concept design

In this paper, the reliability of the missile power supply design is verified by the model design for the cruise missile electrical system. It is concerned whether the electrical system can provide power and transmit signals for the electrical equipment of each sub-system in a timely, correct, safe and reliable manner. The simulation analysis mainly covers the following requirements:

Requirement Item	Requirement Content
Structural Analysis of Electrical System	Capture the pivotal electric equipment on the aerodynamic missile and organize the topological logic relationship of the electric equipment for the aerodynamic missile.
Full-elastic electrical load analysis	Capture the power consumption of various electric equipment during the flight attack of the aerodynamic missile, identify the power consumption status of the battery at different stages, and verify the peak power consumption design of the electric equipment.
Overall Power Consumption Analysis	Capture the main power capacity design requirements by analyzing power consumption in combating process. The purpose is to prevent excessive battery capacity in the design which would increase volume and mass of batteries.

Table 1 Requirements of Simulate Missile electrical system.

Through the analysis of the typical subsystem conditions in the missile strike engineering, we used SysML to build the main structure of the cruise missile. The battery module is used as the main module of the power supply to provide voltage and current for each electrical equipment. The power system and the integrated processing system of the missile simulate the flight and strike process of missile and realize the flight state control and position control of the missile. The bus module simulates the missile bus, connects and controls electrical devices and changes their working state. The electrical devices on the missile are generally classified into pulse-type electrical equipment, step-type electrical equipment, and random-type electrical equipment according to the type of electricity usage. In this case, some typical electrical equipment was selected for power consumption simulation analysis. The missile steering engine is a typical random load structure whose power consumption will change continuously according to the flight attitude of the missile during the flight. The radar is a typical rated load, which has two states, one is standby state whose power consumption is low and the other state is the radar-on state whose consumption is significantly higher than the standby state. The igniter is a typical instantaneous heavy load, which means a large current is generated only at the ignition instant to consume the battery. The topological structure model of the missile using SysML is shown in the figure below.

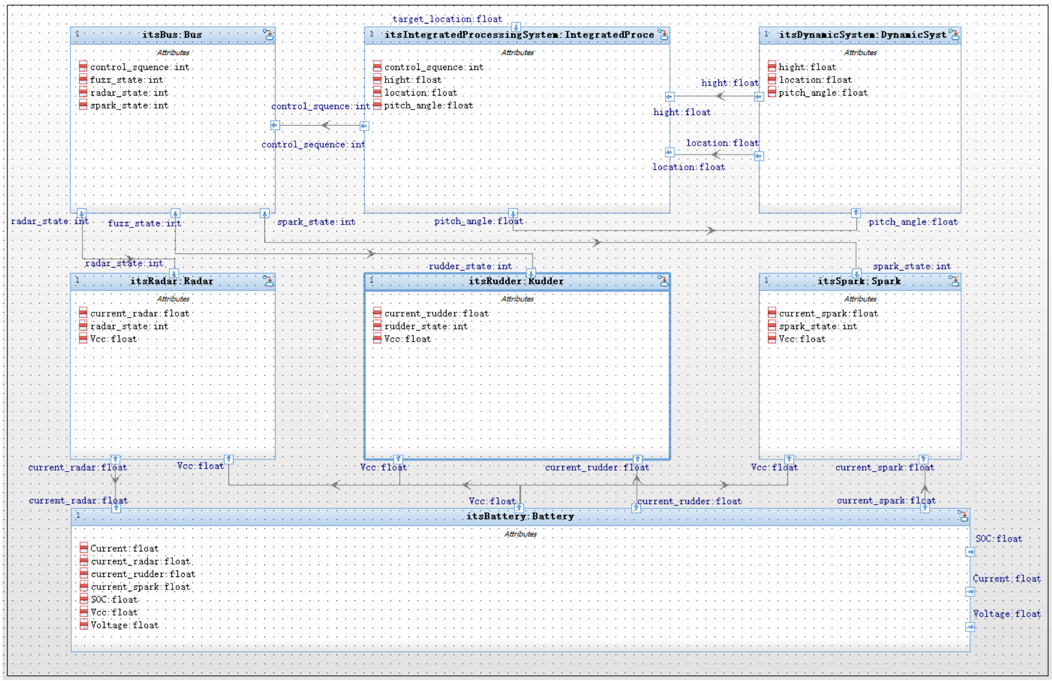


Figure 4 SysML model of missile electrical system.

In the interface relationship, the dynamics equation of missile inputs the coordinate position as the ballistic condition for scene control module, which helps control module to determine the position of the missile and output the missile pitch angle to change the missile flight attitude and the power state control sequence of the power device to switch electricity usage condition at each stage. The battery module provides a voltage and current interface to output the circuit simulation module. The circuit simulation module combines the voltage and current with the state interface to provide the power connection and control signal input terminal for the power device.

4.2 General model transformation based on FMI

FMU is encapsulated by an XML file and compiled by C. FMU contains model description XML documents and DLL binary file. XML is a model description file, which mainly describes the static parameters of the model, including input, output, parameters and other model information, while DLL binary file mainly contains the dynamic information of the model, which is mainly composed of ordinary differential equations.

To transform the model into the FMU model, we should firstly establish an appropriate controller, input and output ports and parameters, where the controller and input ports are necessary. Then generate the *model.c* and *model.h* files of C through the automatic code generation technology of the tool. Then, we could compile the header file and source file through makefile based on the FMI protocol to generate the model.xml and model.dll files. Finally, the model.xml, model.dll file and other files are encapsulated to obtain a *model.fmu* file that can be integrated and simulated on an environment with FMI protocol.

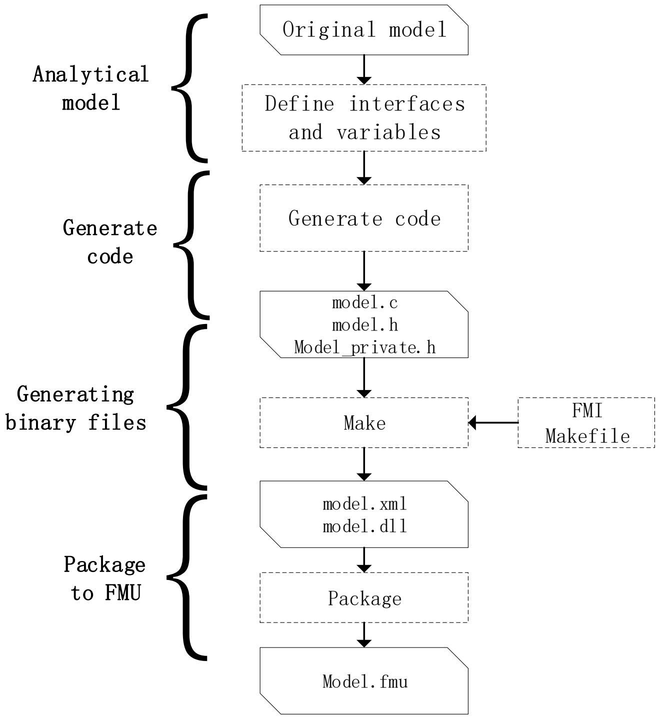


Figure 5 Process of Generate FMU.

Not all features of the SysML model can be preserved after being transformed into FMU models. The model description file of FMU contains the parameters, variables, data types and ports of the model. These elements come from the Attribute, FlowAttribute and Flowport in the Block of the SysML model. The binary DLL file includes the dynamic running information of the model which is from the StateMachine and Operation in the Block of the SysML model. Other elements of SysML such as Activity, Sequence and etc. are not preserved during the transformation to the FMU model.

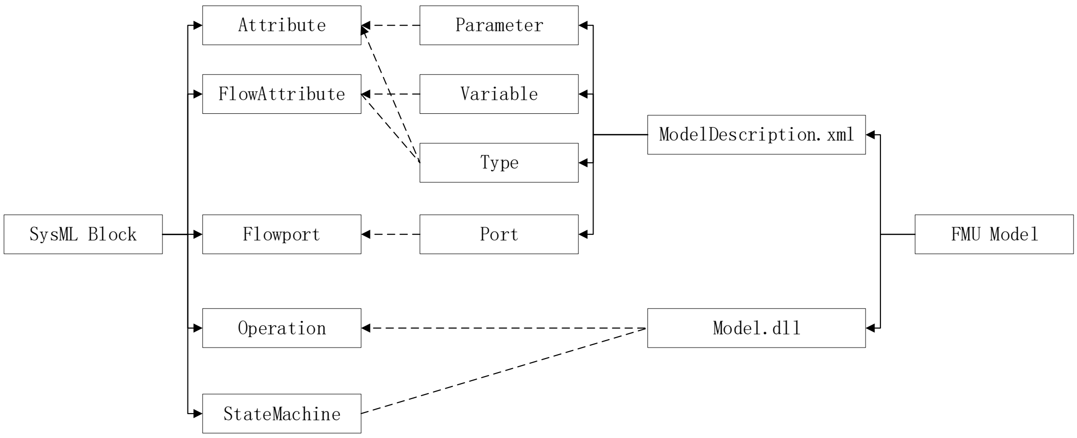


Figure 6 Element mapping between SysML Block and FMU.

4.3 Unit Design

For a simulation modeling design of a complex system, modeling language and tools is used would hardly be agreed. On the one hand, the model is not the system itself, the model is just a representation of a certain aspect of the system. Therefore, the modeler needs to select the most appropriate modeling language and tools to represent the system according to the characteristics of it. On the other hand, due to the wide range of systems design, there are different tools to support the different fields of mechanics, control, thermals, electrics, hydraulics, and network. Different software provide more professional solutions to equations in different fields and they also force the modeler to use the corresponding tools to finish the modeling of the system.

In this case, for the topological characteristics of the missile electrical system, the model is divided into three types in the unit design, which are logical model, physical model and mathematical model. The logical model has obvious logical features, such as state, switch and other structures, or is generally implemented in software in the system, involving control, network and other fields. Physical models have obvious physical structures, such as mechanical structure, involving mechanics, electrics and other fields. Mathematical models have obvious mathematical characteristics, which could generally be abstracted into equations and directly reflect the mathematical characteristics of certain aspects of the system. According to the characteristics of each aerodynamic missile system, the integrated process system and bus module are distributed into logical models, whereas the dynamic system, igniter, radar and battery are defined as mathematical models and the steering gear is defined as physical model.

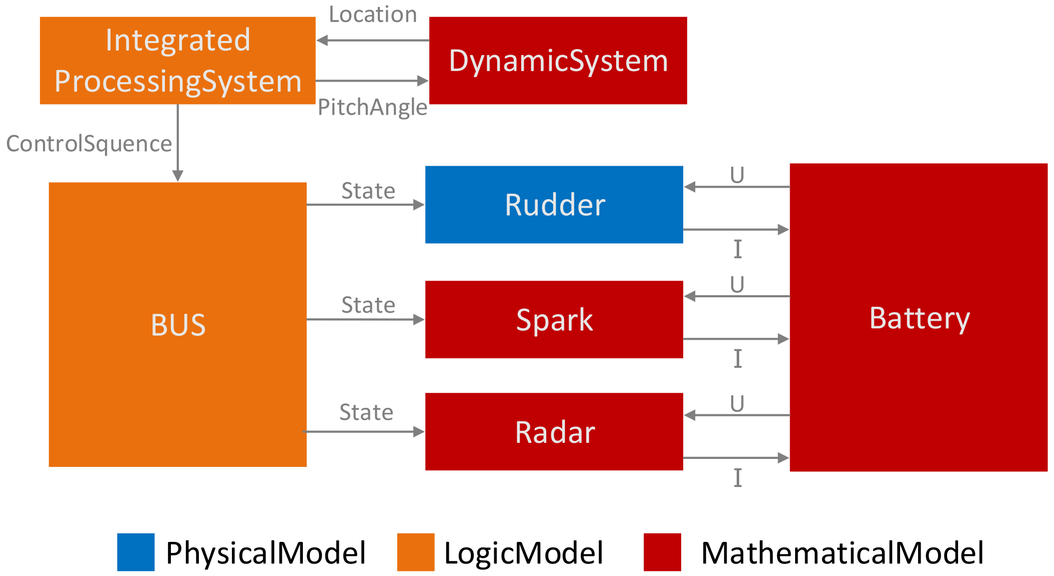
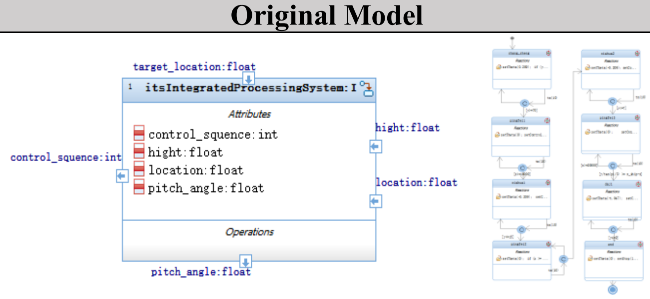
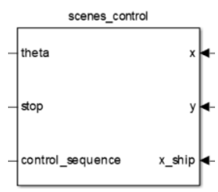


Figure 7 Select the corresponding modelling language based on the object type.

As for the selection of modelling languages among these three models, the SysML language uses state logic to describe system functions that can be used to describe logical models, while Modelica is an open, object-oriented and equation-based language that can be used for complex physical systems such as mechanics, power, electronics, hydraulics, heat, and control. In addition, Simulink has a rich library of mathematical functions and a reliable solver that can well describe the mathematical characteristics of the system.

This case uses the corresponding modelling language to design the sub-systems of the aerodynamic missile electrical system. The table below shows some of the modules in the original model and FMU model.

Model Name	Original Model	FMU
IntegratedProcessingSystem		

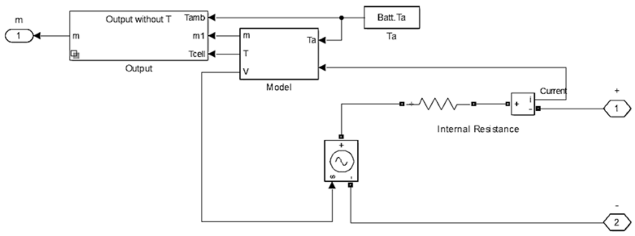
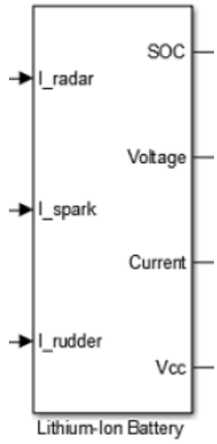
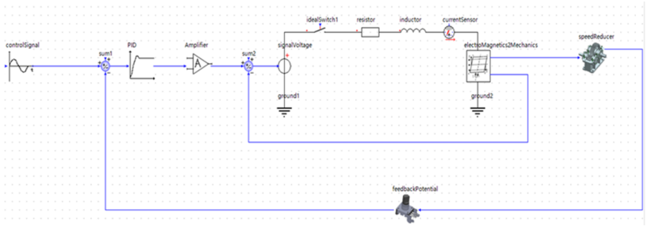
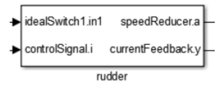
Battery		
Rudder		

Table 2 Original Model and FMUs of some modules.

4.4 Co-simulation Test

After completing the detailed design, we built a virtual prototype of the system and were going to integrate the model results. The interface of the FMU model can preserve the topological information of the system and recover the system topology quickly. We import the generated FMU module on the Matlab tool for interface connection. The topology connection relationship is shown in the figure below.

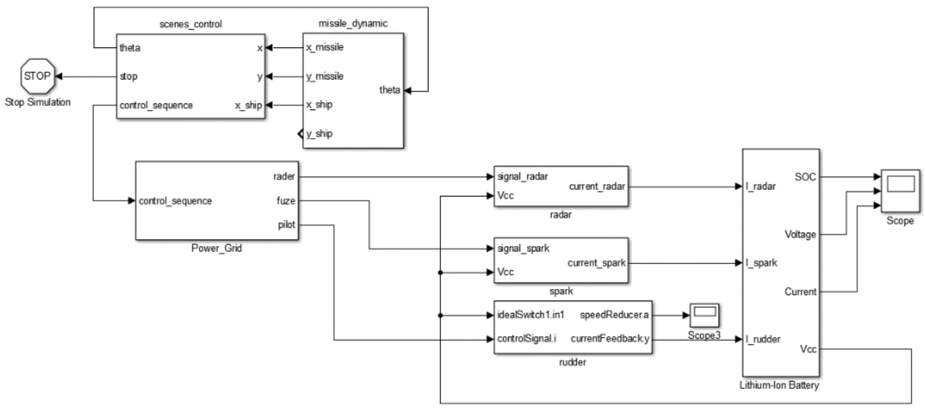


Figure 8 Missile electrical system in FMUs.

After the connection is established, the prototype can be run. The flight process of the missile designed in this case is shown in the figure below. The missile climbs to a height of about 70 meters above sea level(A) after launch, then the steering gear controls the missile attitude to keep the missile in a cruise state(K). After a 10-km flight, the missile starts to reduce the altitude to about 20 meters above sea level and keep on level flight until it reaches the location with a distance about 10 km toward target and turns on the radar(K3). After identifying the enemy target, the missile reduce its altitude again(C) to about 5 meters above sea level. When approaching the target with distance within 1 km, the steering gear adjusts attitude of missile and missile strike the target.

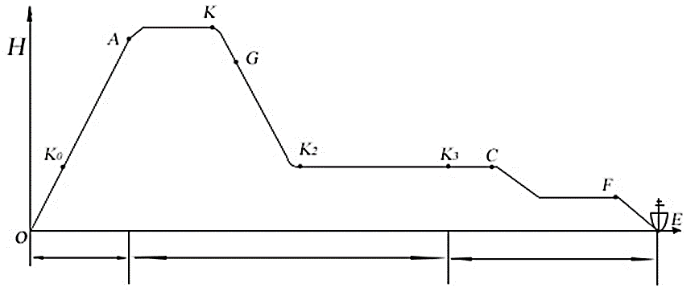


Figure 9 Designed flight path of cruise missile.

The missile trajectory curve obtained through the simulation is shown in the figure below. From the results, the missile flight trajectory trend is almost in line with the program ballistic design, and strikes the enemy ship after a 31.7 km flight.

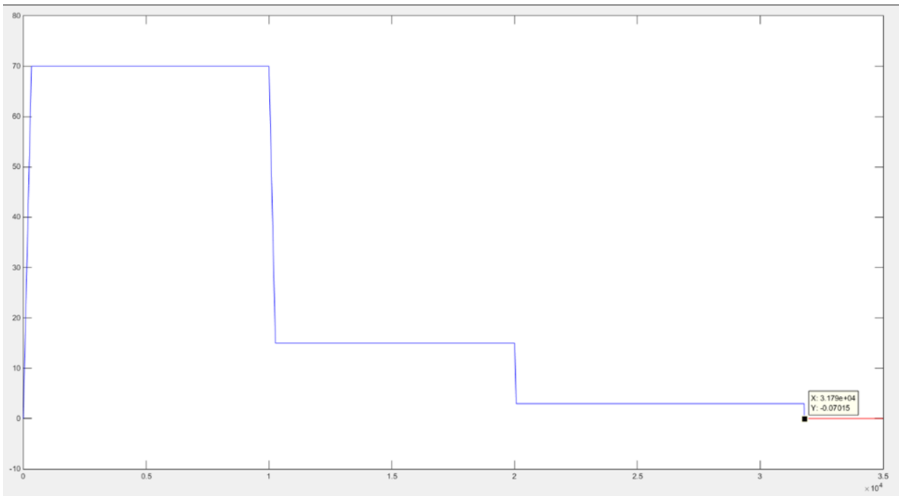


Figure 10 Co-Simulation results of cruise missile flight path

The change of battery characteristic value during the strike process is displayed through the reserved battery parameter port where the following curve result is obtained. It can be seen from the curve that the battery power consumption is about 65% from the launch to the strike process and the acceleration is decreasing after the radar turns on. The voltage value released by battery changes with the decrease of the battery margin. Therefore, in order to ensure that each system works properly, the battery capacity needs to be controlled to keep it above the voltage normal operation required. From the instantaneous current, when missile changes the flying state, such as climbing or flying down, the instantaneous current increase. Thus, balancing peak margin design through different flying state could help guarantee the reliability and economy of the design.

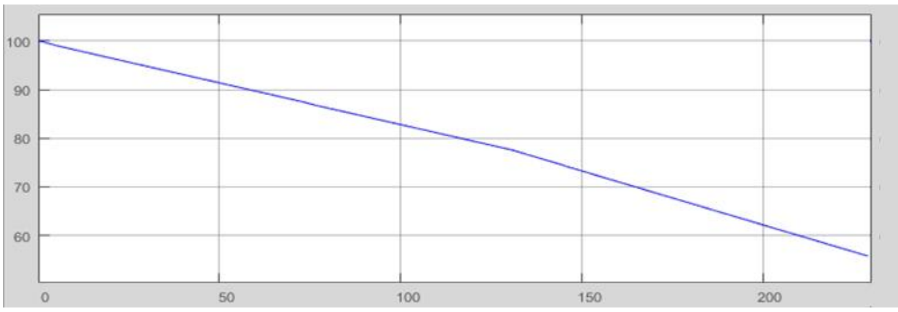


Figure 11 SOC simulation curve of battery.

Figure 12 Simulation Curve of battery voltage characteristic.

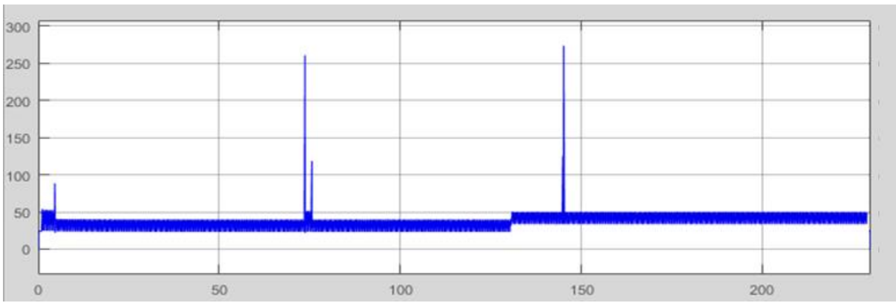


Figure 13 Simulation Curve of Instantaneous Current of Battery

5. CONCLUSION

In the project design and application based on model, risk in project implementation decreases by dividing work into stages and using testing process at key points to verify product design. Applying FMI protocol to generate intermediate files could break through the limitations of multi-modeling tools and multi-modeling languages, realizing the value extension of model results and promoting the communication between integrators and subcontractors. That contributes to rapid progress of product. The FMU model can be understood as a reduced order model (ROM: Reduced Order Model), which abandons the visual properties of the original professional modeling tools and the better professional solver for the platform. FMU model is a black box module with interface that only saves internal logic, making the application form of the model more flexible. SysML models are given interface attributes and behavioral logic when verifying the requirement, which means the value of model is not only the visualization function, but also the possibility of participation in unit design and virtual prototype integrated validation, realizing the consistent validation of system logic.

6. Contact Author Email Address

mailto: sjs609090240@gmail.com

7. Copyright Statement

References

[1] Hoffmann H P. Systems engineering best practices with the rational solution for systems and software engineering[J]. version 4.1. IBM Corporation, Somers, NY (US), 2011.

[2] Cantor M, Plug R U P. Rational Unified Process for Systems Engineering Part 1: Introducing RUP SE Version 2.0[J]. The Rational Edge (August 2003), 2003.

[3] Friedenthal S, Griego R, Sampson M. INCOSE model based systems engineering (MBSE) initiative[C]//INCOSE 2007 Symposium. 2007, 11.

[4] Yishai A. Feldman, Lev Greenberg, Eldad Palachi. Simulating Rhapsody SysML Blocks in Hybrid Models with FMI. Lund, Sweden:Proceedings of the 10th International Modelica Conference; March 10–12, 2014.

[5] Bastian J, Clauß C, Wolf S, et al. Master for co-simulation using FMI[C]//Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany. Linköping University Electronic Press, 2011 (63): 115-120.

[6] MODELISAR Consortium. Functional Mock-up Interface for Co-Simulation[J]. Accessed March, 2010, 1: 2013.

[7] Broodney1 H, Greenberg2 L, Hirose3 S, et al. Open Standards Simulation Flow for Heterogeneous System Models1[C]//INCOSE International Symposium. 2015, 25(1): 1-15.