Preprint

This is the submitted version of a paper presented at INCOSE International Symposium 28.

Citation for the original published paper:

Zhigang, L., Mengfei, Y., Qi, C., Qing, D., Ning, X. et al. (2018)
An MBSE Tool to Support Architecture Design for Spacecraft Electrical Power System
In:

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-225132
An MBSE Tool to Support Architecture Design for Spacecraft Electrical Power System

Zhigang Liu*
Beijing Institute of Spacecraft System Engineering
bitlzg@163.com

Mengfei Yang*
Beijing Institute of Spacecraft System Engineering
yangmengfei@cast.casc

Qi Chen
Beijing Institute of Spacecraft System Engineering
chenqi@cast.casc

Qing Du
Beijing Institute of Spacecraft System Engineering
duqing@cast.casc

Ning Xia
Beijing Institute of Spacecraft System Engineering
xianing@cast.casc

Jinzhi Lu*
KTH Royal Institute of Technology
jinzhi@kth.se

Copyright © 2018 by Zhigang Liu, Mengfei Yang, Ning Xia, Qing Du and Jinzhi Lu. Published and used by INCOSE with permission.

Abstract. Since difficulty and complexity of spaceflight missions are increasing, current electric systems in the spacecraft become growingly complicated challenging the spacecraft electrical power system (EPS) architecture development. By using traditional approaches, system architectures are managed by document collections including various reports, documents and data chats. Due to lack of a unified visualization of system architectures, consistency and traceability among data, documents and information are difficult to manage, particular in supporting architecture design and system verification and validation. This paper presents a model-based systems engineering (MBSE) tool with emphasis on domain specific modeling (DSM), architecture visualization, model transformation and automated verification and validation by using Modelica. The MBSE tool aims to develop the architecture models of EPS, to support MBSE-driven visualizations of architectural hierarchies inspired by related techniques of Google Earth and to realize architecture-driven verification & validation by using Modelica language. Finally, through a case study of EPS, we evaluate the feasibility of our tool.

1. Introduction

Electrical power systems (EPS) generate, store, supply, control and distribute electrical powers for spacecraft (Mengshoel et al. 2010). As a core component, EPS influences system performance, reliability and life-span of spacecraft. Since the difficulty and complexity of spaceflight missions are increasing, current EPS architectures become more complicated. This challenges EPS development, particular in architecture design, verification and validation (V&V) with high quality traceability and consistency management of related design information.

By using traditional systems engineering approaches, EPS architectures are described by a set of loop coupling document collections including documents and data chats, e.g. interface data sheet, wiring harness diagram, ground diagram and power supply diagram. In different stages of development life cycle, such documents and data chats allow stakeholders for obtaining and maintaining the relevant design information. Due to lack of integrated management of such document and data chats, consistency and traceability led by design changes are challenged. Moreover, architecture
descriptions of different views lead to difficulties to understand and communicate with each other among stakeholders. In addition, current EPS development process integrates with sets of V&V methods, e.g., simulation. High quality management of such simulation models and the related architecture concepts can promote the efficiency of EPS development.

In this paper, the goal is to adopt related techniques of Model-based Systems Engineering (MBSE) (Beihoff et al. 2014) to provide a tool to formalize, design and analyze EPS architectures and enhance management capabilities of the relevant technical resources, e.g., models and data. It requires us to overcome the challenges in the previous paragraph.

**Objective 1**-support for designing EPS architectures in a systematic and structural way: The EPS architectures need to be identified and described on an appropriate level of abstraction by using a model-based way.

**Objective 2**-support for communicating the related information of EPS architectures in a concise and graphical way: EPS architectures need to be visualized in a unified and friendlier environment.

**Objective 3**-support for implementing V&V of related architecture concepts automatically: Based on functional requirements of EPS, each architecture concept has the expected behaviors. Architecture-driven methods can support EPS developers to generate corresponding Modelica models and implement automated V&V of such behaviors automatically in order to promote traceability between architecture models and simulation models.

The rest of the paper is organized as follow: Firstly, we introduce related work in Section 2. In Section 3, we presented how our proposed MBSE tool supports model-based architecture design and automated V&V by using Modelica language. Then we evaluate the tool by using a case study in Section 4. Finally, we conclude our work with a summary in Section 5.

### 2. Related Work

Among the existing MBSE methods, system modeling languages, e.g., SysML, UML (Friedenthal et al. 2008) and domain-specific modeling (DSM) languages (Mannadiar & Vangheluwe 2010) are used to formalize the system architectures. System developers use a series of visual notations and relationships in these languages to build the models describing system architectures and capturing the related information. These architecture models can promote dependency, traceability and consistency of the architecture description in different hierarchical layers. Compared with system modeling models, DSM models can capture domain-specific concepts that are created specifically to model some problem domain. For example, META program led by The Defense Advanced Research Projects Agency (DARPA) specifically aimed to promote efficiency of the product development and deployment through model-based design and manufacturing. META program researchers developed the META language, called CyPhy as a model integration language. Cyphy included a sublanguage named ADML (Architecture Design Modeling Language) representing hierarchical component architectures and typed interfaces.

MBSE methods supporting complex system V&V are widely used in industry. NASA's Jet Propulsion Laboratory (JPL) used the related techniques to develop the aerospace system in practice. The main work included the extension of SysML with domain concepts to design DML, a DSM language formalizing and designing specifications and architectures of the electrical flight system (McKelvin, Jr. & Jimenez 2012). Moreover, they integrated DML implementations with V&V using Modelica that a model transformer was developed between a Modelica tool, MapleSim and a SysML modeling tool, MagicDraw (Schamai et al. 2012). Another approach suggested by OMG™ extended SysML and allowed for using Modelica language concepts directly in SysML by SysML4Modelica profile (Votintseva et al. 2012). For example, Georgia Institute of Technology and Ford Motor Co. explored to describe the system architecture based on SysML for the automotive architecture design.
First of all, SysML was used to define the reference architecture models and then a model transformer is used to generate Modelica and Simulink model templates, which are converted from SysML architecture models, automatically. These templates are used to verify the system performance of architecture models automatically without manual modeling processes. The authors put forward a framework for modeling automotive architectures by using SysML and the corresponding simulation models and develop a model transformation tool supporting automated generations of simulation models to promote consistency of the whole development process.

Visualizations can support system developers to analyze domain specific concepts. Traditionally, current modeling tools adopt static architecture views in different hierarchical levels to represent architecture descriptions. For example, Block Definition Diagrams (BDD) and Internal Block Diagram (IBD) in SysML are used to define system structures. If stakeholders want to browse compositions in one block in BDD, a set of manual operations are needed to reach the static view related to IBD. Nowadays, in order to capture, analyze and promote understandings of development information between stakeholders, MBSE-driven visualization is provided to transform related system models and related data to a virtualized way for different stakeholders (Jackson & Wilkerson 2016).

Based on literature reviews of related work, we proposed an MBSE tool by using DSM, architecture-driven V&V by Modelica and MBSE-driven visualizations to support EPS architecture design. Compared with the traditional systems engineering methods of architecture design, our tool can support EPS architecture design by using a model-based way and MBSE-driven visualization and realize automated V&V to promote traceability between different views, particularly in the bottom-up process.

3. Model-based Architecture Design and Verification and Validation

3.1 Overview

We make use of a DSM approach in our tool to formalize, describe and design EPS architectures. A DSM model refers to an architecture model we define in this paper, indicating a set of structure diagrams of EPS in different hierarchical layers. An MBSE-driven visualization inspired by the relevant techniques of Google Map can be performed in our tool to create a unified visualization environment of architecture models. Modelica language is used to support V&V of the related architecture models. A model transformer is developed to transform DSM models in our tool to related Modelica models performing the expected behaviors of different architecture concepts.

3.1.1. Domain-specific Modeling Approach
Figure 1 Domain specific modeling supporting architecture design and V&V

In Figure 1, a DSM solution used in our tool is introduced. Our tool can support architecture design and V&V within a development process of EPS. Developers make use of DSM to support architecture design and Modelica language to support system V&V in our tool. In order to support definitions and implementations of DSM and Modelica models, four modeling levels are used to construct model concepts in our tool.

- **M3** (Referring to Meta-meta-model and Modelica specification)
  Meta-meta-models (Simpson et al. 2001) in our tool are defined based on GOPPR meta-modeling language (Juha-Pekka Tolvanen n.d.). The meta-meta-models include graph, object, relationship, role, property and port to support definitions of meta-model of EPS architecture models. The Modelica specification is used to build Modelica model libraries.

- **M2** (Referring Meta-models and Modelica library)
  We define and implement meta-models (Simpson et al. 2001) in M2. These meta-models are used to build DSM models for architectures models in different hierarchical layers. Meta models are instantiations of meta-meta-models. Modelica model libraries are built based on Modelica specifications.

- **M1** (Referring to DSM and Modelica models)
  DSM models indicate architecture models describing EPS architectures in different hierarchical layers. Modelica models are multi-domain models used for V&V of the related architecture concepts.

- **M0** (Referring to modeling purposes)
  In this paper, DSM models are used to describe the architecture of the real system, EPS. Modelica models aim to verify and validate EPS architecture concepts.

### 3.1.2. MBSE Workflow during Architecture Design
By using our tool, an MBSE solution can support architecture design, MBSE-driven visualization and architecture-driven V&V. In Figure 2, an MBSE workflow how our tool supports architecture design of EPS is introduced. Based on the functional requirements of EPS, a set of work tasks are implemented to design hierarchical architecture models. We first develop meta-models and Modelica model libraries in different hierarchical layers: system layer, device layer and circuit layer. Then developers build related DSM models referring to architecture concepts in different layers through meta-models. Afterward, system developers generate related Modelica models supporting V&V automatically by using a model transformer in the tool.

The detailed functionalities that our tool supports are introduced as follow:

1) **Architecture Design**: Based on functional requirements of EPS in different layers, DSM is used to construct system architecture models. A definition of *System Level* is used to represent hierarchical layers of architecture models (introduced in the next section). In architecture models in different *System Levels*, interfaces in each block including inputs and outputs are defined to connect to other blocks. Each architecture model can be decomposed into more detailed architecture models.

2) **MBSE-driven visualization**: By using MBSE-driven visualization techniques, hierarchical architecture models can be browsed through zooming freely. In addition, simulation results of V&V can be presented and linked with the corresponding architecture models in order that stakeholders can capture architecture descriptions from different views - macro, holistic, static and dynamic views. The visualization techniques improve the consistency of stakeholders’ understandings about architectures and enhance the traceability between architecture models and V&V.

3) **Architecture-driven system V&V**: Based on architecture models and Modelica libraries for EPS, a model transformer is used for generating Modelica models from architecture models in different *System Levels*. These Modelica models can be implemented automatically for performance analysis, functional analysis and reliability analysis.
3.2 Architecture Design

3.2.1 Hierarchical Architecture Design

We make use of System Level to define hierarchical layers of EPS architectures. System Level 0 is the top layer of EPS. As the System Level is increasing, the hierarchical layers of related architecture models are higher (more detailed). The architecture models in the low System Level define the extra-interfaces, inter-connections and compositions of architecture models in higher System Level. For example, as shown in Figure 3, the architecture model of power supply and distribution system is in System Level 0; In System Level 1, a more detailed architecture model including battery and solar panel is presented; more detailed architecture models of devices and circuits are shown in System Level 3 and System Level 4. The detailed information is introduced as follow:

1. **System Level 0-Top level**
   As the top level of EPS, the architecture model defines high-level interfaces to Environment, e.g., satellite orbits and illuminations and other interfaces to other high-level systems, e.g., interfaces to guidance navigation & control (GNC) or Measurement control.

2. **System Level1-System level**
   In System Level1 (System level), based on constraints of interfaces in System Level 0, an architecture model is defined including required devices and energy flows between them. For example, in Figure 3, in order to satisfy the functional requirements of EPS, devices including battery, power control and distribution unit (PCDU) and solar panel are defined. At the same time, interfaces in such devices and connections between them are defined.

3. **System Level2-Device Level**
   In System Level2, components and the correspondent connections required in devices of System Level1 are defined. For example, components, e.g., battery charging regulators (BCR), battery discharge regulator (BDR) and shunt regulation (SR) are used to construct PCDU. Therefore, an architecture model of PCDU in System Level2 is defined by using these components and connections.

4. **System Level3-Circuit Level**
   In this level, based on circuit units in DSM library, e.g., resistance, capacitance and diode can be defined to construct components in detail. Particularly, loading equipments are defined as black boxes which are connected with other circuit units.

![Figure 3 Hierarchical levels in EPS Architecture](image-url)
3.2.2 Domain-specific definitions of meta models

In order to design and implement DSM models of EPS, we make use of the domain specific concepts to define meta-models in different System Levels. In Table 1, meta-models, related descriptions and the corresponding meta-meta models are illustrated. Each meta-model has their own graphic syntax corresponding to the representations of the related electronics system as shown in Figure 4. For example, in level 2, the green rectangle with specific component information represents related component of the EPS.

<table>
<thead>
<tr>
<th>Domain specific concepts (Meta-models)</th>
<th>Description</th>
<th>Meta-Meta model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Level 0</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Block</td>
<td>Spacecraft subsystems, environment of EPS can be treated as different systems</td>
<td>Object</td>
</tr>
<tr>
<td>Power Port</td>
<td>External system interfaces</td>
<td>Port</td>
</tr>
<tr>
<td>Power Connection</td>
<td>Energy flows between systems</td>
<td>Relationship</td>
</tr>
<tr>
<td>System Property</td>
<td>System characteristics</td>
<td>Property</td>
</tr>
<tr>
<td>Environment Connection</td>
<td>Connections between system and environment</td>
<td>Relationship</td>
</tr>
<tr>
<td><strong>System Level 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Block</td>
<td>Devices constructing EPS</td>
<td>Object</td>
</tr>
<tr>
<td>Electrical Port</td>
<td>External device interfaces</td>
<td>Port</td>
</tr>
<tr>
<td>Electrical Connection</td>
<td>Device connections</td>
<td>Relationship</td>
</tr>
<tr>
<td>Device Property</td>
<td>Device characteristics</td>
<td>Property</td>
</tr>
<tr>
<td><strong>System Level 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Block</td>
<td>Component constructing devices</td>
<td>Object</td>
</tr>
<tr>
<td>Electrical Connector</td>
<td>Connections between components</td>
<td>Relationship</td>
</tr>
<tr>
<td>Component port</td>
<td>External Component interfaces</td>
<td>Port</td>
</tr>
<tr>
<td>Component Property</td>
<td>Component characteristics</td>
<td>Property</td>
</tr>
<tr>
<td><strong>System Level 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circuit components</td>
<td>Circuits constructing device</td>
<td>Object</td>
</tr>
<tr>
<td>Wire</td>
<td>Circuit connections</td>
<td>Relationship</td>
</tr>
<tr>
<td>Pin</td>
<td>External circuit interfaces</td>
<td>Port</td>
</tr>
<tr>
<td>Circuit Property</td>
<td>Circuit characteristics</td>
<td>Property</td>
</tr>
</tbody>
</table>
3.3 MBSE-driven Visualization of Architecture

Inspired by Google Map, an MBSE-driven visualization technique is used to present EPS architecture concepts in different System Levels. Elements instantiated from Object in M3 in the DSM models refer to landforms and connections between elements refer to roads. EPS Architectures in different System Levels and related information can be browsed through free zooming as shown in Figure 5.

In detail, our MBSE tool proposed in a unified 3D browsing environment for MBSE-driven visualization which architecture models in different System Levels are represented. As Figure 5 shows, in the visualization environment, stakeholders can control the Z-axis to browse different hierarchical structure of architectures which link to the verification results generated by Modelica models. Moreover, the tool also can represent architecture descriptions of different views: views in different levels, views for dynamic performance of EPS and views of energy flow of EPS (details shown in Figure 9 and Figure 10).
3.4 Architecture-driven Method for Automated V&V

In our tool, an architecture-driven method is implemented to support automated V&V. A model transformer is developed to generate Modelica models from DSM models describing EPS architectures. In this section, we first introduce Modelica libraries supporting V&V and then the model transformation supporting automated V&V.

3.4.1 Modelica Model Library based on EPS

According to meta-models defined based on domain specific concepts, related Modelica models supporting automated V&V are developed. These Modelica models are built in different System Levels referring to specific systems, devices, component and circuits, as shown in Figure 6.

In System Level 0, high-level systems of spacecraft are built. There are three types of model in this level: power supply system model, distribution system model and load system model. The parameters of these models include voltage, current, power, etc.

In System Level 1, according to the functional requirements, the device compositions, control algorithm of the devices, and specify the power flow and key property of the devices are built. For example, Modelica model library in the device level includes models of solar array, battery, etc.

In System Level 2, internal components of the device are built, e.g., battery cells.

In System Level 3, internal circuits are built.
3.4.2 Model Transformation Supporting Automated V&V

Based on architecture models in different System Levels, a model transformer developed to implement automated generations of Modelica models for verification and validation in Figure 7 from related DSM models. The model transform is developed based on the existing Modelica libraries supporting EPS simulations. The workflow in the model transformer is as follow:

1. **STEP1: Generating Modelica models in System Level 1**
   Based on the used meta-models referring to devices in System Level 1 and developed Modelica model library, related Modelica models in System Level 1 are generated. The Modelica models include block, parameter, interface and equation definitions.
   - **Block definition:** Modelica blocks related to devices are generated with the same meta-model name. For example, a DSM model includes a battery block built based on a meta-model of battery. After transformation, a Modelica model is generated with the name, `battery` according to the battery block in DSM model.
   - **Parameter definition:** Based on properties in the DSM models, parameters in Modelica models are declared. For example, parameter `Modelica.SIunits.Voltage` in Modelica models refers to the corresponding `voltage` in DSM models.
   - **Interface definition:** Based on connections in the DSM models, interfaces of Modelica models can be generated. For example, in a DSM model, there is an interface called X01.
connected to another interface called [16], a Modelica interface is generated and named as 
**Interfaces.Special.connector X01[16]**.

- **Equation definition**: Based on Modelica specification, equations of related domain specific concepts indicated by meta-models and equations of connectors are generated.

```model Load1 "1.装置模型"

parameter Modelica.SI.Units.Power PowerLoad = 20 "输入功率";
parameter Modelica.SI.Units.Voltage V_rated = 29 "额定电压";
parameter Modelica.SI.Units.Voltage V_est = 29 "额定电压";

//电能输入
Interfaces.Special.X01 X01[16]
annotation (Placement (transformation: extent = ((-26.18, -26.18), (26.18, 26.18)), rotation = 90, origin = (-106, 100)));

//控制开关
Modelica.Blocks.Interfaces.BooleanInput control "true => p--o connected, false => switch open"
annotation (Placement (transformation: extent = ((-20, -20), (20, 20)), rotation = 270, origin = (0, 100)));

end Load1;
```

**Figure 7 Generating Modelica models of device**

2. **STEP2: Generating Modelica models in System Level 0**

Based on DSM models referring to DSM models in System Level 0, related Modelica models are generated based on the developed Modelica model library and the generated Modelica models of related devices in **STEP1**, as shown in Figure 8. These models include external connections of EPS definition, device definitions, environment definitions, and connector equation definitions.

- **External connections of EPS definition**: Define the external interfaces of EPS, for example, *SI.Voltage V_Bus* and *SI.Current I_Bus* refers to refers to bus voltages and currents to systems or environments in System Level 0.

- **Device definition**: Define devices in EPS. Based on existing blocks in Modelica model library mentioned in 3.4.1, Modelica models of the related devices are generated. If the related Modelica models are not defined in the Modelica model library, **STEP1** is implemented to define Modelica blocks for the related devices.

- **Environment definition**: Define environment blocks based on Modelica model library. For example, *Environment.Environment* refers to environments of EPS.

- **Connector equation definition**: define connector equations of connections between devices. For example, *Port 4 in Connection X01 in Device A connects with Port 2 in Connection X03 in Device B*. Based on Modelica specification, Modelica code, "connect(A.X01[4],b.x03[2]); " refers to the related connector equation is generated.
4. Case study and Evaluation

In this section, we make use of a real case, an application used in spacecraft development to illustrate how our tool supports architecture design, MBSE-driven visualization and automated V&V by using Modelica. The case study is an EPS of a returning spacecraft for exploration missions of deep space.

The returning spacecraft is composed of the service module and the return module. The service module adopts full adjustment bus power supply mode, with two bus in the EPS. Each bus is connected to one solar array, one hydrogen-nickel battery, and one power supply controller. Two bus pass through the main controller, and are connected in parallel to form a unified common bus to supply power to the loads.

We adopt our tool to design the architecture of the returning spacecraft in an up-down process and construct DSM models for different hierarchical architectures, as shown in Figure 9. It includes almost 150 electronic devices and more than 10000 connections. During architecture design, amount of existing models can be reused and extended.

In order to support model transformation from architecture models to Modelica models, meta-models and related Modelica model library in different System Levels are built, as shown in Figure 6. Based on the architecture models, we analyzed the energy balance and power flow of the returning spacecraft and verified the design compliance with the requirements constraints. Figure 10 presents four views of EPS architectures in our MBSE tool. The DSM model indicates an architecture model of spacecraft. The space environment defines parameters in Modelica models of related EPS architecture concepts. Simulation Results can be visualized at the same time.
Figure 9 Visualizations of a returning spacecraft

Figure 10 DSM Models and related Modelica models mapping to DSM models of a returning spacecraft.
Current MBSE tool supports development of architecture models of EPS and visualizations of architectural information. Based on the provided meta-meta models, several meta-models are developed for building the architecture models. Compared with existing languages like SysML and UML, such meta-models are more simplified and less time-consumption when engineers start their MBSE industry practices. Different stakeholders can make use of the MBSE tool to browse the architectures of EPS which can help stakeholders to understand views of others and promote the communications between the whole team.

There are several limitations about the MBSE tool: (1) The links between requirements and architecture models are missing. Current MBSE tool can support to build the architecture models of EPS based on requirements from stakeholders. In the future, two solutions will be done to connect requirements to the architecture models. Firstly, based on the meta-meta models, stakeholder can develop their requirement models in the MBSE tools and then link such requirement models to the architecture models. Secondly, a function to link each meta-model to a specific URI will be developed. The URI can be generated from requirement management tools, such as Doors. (2) The model transformer in current tool is limited. In the future, a code generator editor will be developed in order to transform architecture models to other DSM models. (3) In the future, our MBSE tool can support to develop meta-models based on existing languages such as SysML and AADL.

5. Conclusion

In this paper, an MBSE tool has been proposed to support architecture design, MBSE-driven visualization and automated V&V using Modelica language for EPS. A DSM technique is adopted by the tool to design related architecture models. Moreover, the MBSE-driven visualization in our tool can represent system architecture in multiple views. Furthermore, a model transformer is developed to support architecture-driven V&V which can generate related Modelica models from architecture models automatically. Finally, through a real case of EPS in spacecraft, our tool indeed promote the effectiveness of system developers when they design architectures.

References


Paredis, C.J.J. et al., 2013. Integrating Analytical Models with Descriptive System Models: Implementation of the OMG SyML Standard for the Tool-specific Case of MapleSim and
Zhigang Liu is a senior engineering of Beijing Institute of Spacecraft System Engineering(ISSE). He is charged with the Electrical Power System design of the Lunar Sampling and Recycling Project and the Mar exploration project in China, and he is experienced in system engineering using MBSE for Spacecraft power system design. He holds Ph.D degree from Beijing Institute of Technology.

Mengfei Yang is a Professor of China Academy of Space Technology and Fellow of International Academy of Astronautics(IAA). He is the Chief designer of the Lunar Sampling and Recycling Project of China. He is experienced system engineer having dealt with wide range of disciplines ranging from computer science, control theory and engineering, and complex electro-mechanical system design.

Chen Qi is the vice minister of Beijing Institute of Spacecraft System Engineering(ISSE). He is responsible for the planning and construction of the technical capability of space information and electronic technology. He is an expert of the remote sensing satellite power system design.

Qing Du is a senior engineer of Beijing Institute of Spacecraft System Engineering(ISSE). He focuses on spacecraft power system design and MBSE application.

Ning Xia is an engineer of Beijing Institute of Spacecraft System Engineering(ISSE). He focuses on spacecraft power system design and MBSE application.

Jinzhi Lu, ph.d student at KTH Royal Institute of Technology, Mechatronics Division (70%). Meanwhile he works in Suzhou Tongyuan Software and Control Technology Company as a project manager (30%). His research interest is MBSE tool-chain design and MBSE enterprise transitioning. He is senior member of China Council on Systems Engineering (CCOSE), China Council on Systems Engineering