A Visual transformation method of SysML model to Modelica model

Chao Fu1,a, Jihong Liu1,b*, HongYan Yu2,c, WenTing Xu2,c,
1 School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China
2 Beijing Institute of mechanical and electrical engineering, 100191 Beijing, China
asuperfurun@163.com, b*ryukeiko@buaa.edu.cn, cadsupport@163.com

Abstract—The core idea of Model based systems engineering (MBSE) is to model all the information related to the system design starting from the beginning of the system development in the integrated form, and serve as a central system model throughout the product system life cycle. In view of the lack of executable capability of the standard system modeling language (SysML) within MBSE, the system design-simulation model transformation method was studied. Visualizing the transformation process based on SysML could contribute to the low complexity of the transformation. Finally, an example of 3D printer is used to verify the proposed method.

1. INTRODUCTION
MBSE use model instead of document to build system design model, and supports system design, analysis, verification and validation activities through continuous evolution and iteration of the model, which is also the potential of MBSE in the future [1]. But at present, SysML model does not have the ability of execution, so we need to complete the analysis and verification of system design by the transformation of SysML model to physical simulation model.

At present, QVT[2,3] (Query-View-Transformation and TGG[5] (Triple Graph Grammars) is the main method for system design simulation model transformation. Johnson [6] proposed a method to use model transformation technology to realize the integration of design simulation. In this method, TGG was selected as the model transformation language. In the first step, the modeling elements of SysML are extended to build the analysis model, and the corresponding relationship between the extended elements and the simulation language is established. Finally, the transformation process is completed by VIATRA. Among them, the extended language of SysML-Modelica and SysML4Modelica are of great reference significance to the follow-up research. In addition, TGG is a graphical language, which can directly and explicitly describe the corresponding relationship in the process of transformation. Paredis et al. [7] put forward the general process of integrating domain specific languages with SysML. By constructing meta models of the two languages, a meta model level mapping relationship was established. Then, the transformation rules and their constraints are expressed by the executable model transformation language, and the transformation is completed by special external transformation tools.

At present, many kinds of physical modeling and simulation languages, such as Modelica, Simscape, Simulink, have been included in the research scope of the transformation from modeling language to simulation model with SysML as the source model modeling language. However, this kind of research is mostly targeted. How to form a general physical modeling method based on SysML and integrate it
with various types of domain specific languages is still a problem to be studied. So the system
design-simulation model transformation method was studied. Visualizing the transformation process
based on SysML could contribute to the low complexity of the transformation. Finally, an example of
3D printer is used to verify the proposed method.

2. VISUAL TRANSFORMATION METHOD OF SYSML MODEL AND MODELICA MODEL

2.1. Meta model of transformation rule

In order to realize model transformation, a set of modeling elements are abstracted from the actual
model transformation activities. By defining the transformation rule meta model (as shown in Figure 1),
developers can describe the mapping relationship between models in model transformation.

![Fig. 1. The forms of large-scale network survivability associations.](image)

In Figure 1, "Transformation" is an abstract transformation class. Its subclass is
"AutomicTransformation", which is a non separable transformation expression element. Multiple
atomic transformations constitute a transformation task. Each atomic transformation includes a set of
transformation rules, and each transformation rule has a modifier attribute to identify the type
information of the rule. In different transformation description languages, rules have different types,
and specific rule types are implemented by subclasses of rule classifiers. In addition to the type
modifier attribute, each rule is composed of several domains, which represent the components of rule
definition in different transformation languages. Each region also has modifiers that identify the type to
which it belongs.

2.2. Meta model of transformation activity

The process of model transformation can be regarded as the sequential execution of multiple
transformation rules, which can be realized by a set of orderly transformation activities. In this paper,
SysML activity diagram is used to complete the sequential expression of model transformation process,
and its meta model is shown in Figure 2.
In Figure 2, "Activity" represents the activity of model transformation, including activity node and edge. The active node can be instantiated as action node, object node and control node. The transformation action, a subclass of the action node, represents the transformation action to be performed and is associated with the transformation containing the corresponding transformation rules. The object node is represented as pin, which is the input and output of the transformation action. The control node can be subdivided into seven types: initialnode, finalnode (streamfinal and activityfinal), decisionnode, mergenode, forknode, and joinnode, etc. The edge of an activity graph is used to represent the execution sequence of an activity, which is divided into objectFLOW and controlFlow.

By instantiating the meta model of the above transformation activities, the constraint relationship between different transformation activities including multiple transformation rules is established, and the execution process of all transformation activities is finally realized. Taking TGG, a commonly used model transformation technology, as an example, figure 3 shows the equation transformation rules conforming to the TGG specification. The transformation activity diagram is used to express the implementation process of the transformation rules.

For the transformation of inheritance relationship, the transformation is carried out in the order of first parent and then child, as shown in Figure 4. It takes two steps to convert the constraint attributes of SysML model to the equation representing the behavior of Modelica model: 1) ensure that the transformation from parent module to parent component is completed, which is the premise of transformation between subclass constraint and subclass equation; 2) This example includes two parts: condition judgment and parameter transformation. By analogy, any complex transformation activity can be decomposed until it can be directly executed.

2.3. Process of model transformation
The static description of the transformation content can be realized by using the transformation rules, and the dynamic behavior of the transformation can be defined by the transformation activity diagram, so as to express the execution order of the transformation rules, that is, the transformation process. The transformation action is used to execute the transformation rules associated with it. Each action contains
an executable engine to judge whether the execution conditions are met or not, and pass the executed data flow or control flow to the next associated action node to ensure the continuity of the transformation process. Until the final activity is found, the control node can end the entire transformation activity.

![Diagram](image)

Fig. 4. Equation transformation rules defined by activity diagram

### 3. CASE STUDY OF 3D PRINTER DESIGN

Firstly, the 3D printer is modeled by SysML, and the system function and behavior model can be described completely. The printing function mainly includes three input streams and two output streams. The input streams include filamentous solid materials, low-voltage electric energy and control signals. The output stream includes solid materials and heat of fixed shape.

In order to verify whether the 3D printer design scheme meets the technical requirement, such as printing speed, precision, molding weight, etc... The corresponding Modelica simulation model is adopted for system analysis. Firstly, as shown in Fig. 5 the static diagram of system transformation is established according to the system structure level, and the subsystem level transformation model can be further decomposed according to the component composition of the subsystem.

![Diagram](image)

Fig. 5. Model transformation diagram of 3D printing system
Secondly, the subsystem transformation activities can be completed in parallel, and the transformation sequence and type are shown in Figure 7. Block2model is the main transformation activity, and the sub transformation process included in this activity is shown in Figure 8. The transformation from module to model is completed according to the general transformation sequence of name, stream object declaration, inheritance relationship, variable parameter, constant parameter, connector, initial equation and equation change.

Fig. 6. transformation model of 3D printer system

Fig. 7. activity diagram of Sysml2modelica model transformation

4. CONCLUSIONS
Aiming at the lack of simulation capability of MBSE, the system design is verified by system design simulation model transformation. Based on the idea of model transformation based on meta model system, this paper proposes a visual transformation method of system design simulation model based on SysML, which defines two transformation meta models based on SysML extension abstractly and establishes the mapping relationship table between SysML and Modelica meta model layer, realizes the visual transformation process by instantiating the transformation meta model. In fact, to realize the transformation from SysML model to Modelica model, we need to develop a specific transformation engine, which can realize the automatic transformation between models according to the established visual transformation model. This is the next step of our work.

ACKNOWLEDGMENT
The authors acknowledge the financial support from (JCKY2017204B011) and (41401030301).

REFERENCES
[1] Holt J, Perry S. SysML for Systems Engineering[M]. Institution of Engineering and Technology, 2008
[2] Bast W, Murphree M, Michael L, et al. MOF QVT final adopted specification: meta object facility (MOF) 2.0 query/view/ transformation specification.[J]. 2005
[3] Czarnecki K, Helsen S. Classification of Model Transformation Approaches[C]. The Workshop on Generative Techniques in the Context of Model-Driven Architecture. pp 217-219, 2003.

[4] Mens T, Gorp P V. A Taxonomy of Model Transformation and its Application to Graph Transformation[J]. Vacuum, 101(3):189-192, 2010.

[5] Rr A. Specification of Graph Translators with Triple Graph Grammars[C]. Graph-Theoretic Concepts in Computer Science, International Workshop, Wg '94, Herrsching, Germany, June 16-18, 1994, Proceedings. DBLP, pp 151-163, 1994.

[6] Johnson T A, Jobe J M, Paredis C J J, et al. Modeling Continuous System Dynamics in SysML[C]. ASME 2007 International Mechanical Engineering Congress and Exposition. pp 197-205, 1994.

[7] Paredis C J J, Johnson T. Using OMG's SysML to support simulation[C]. Simulation Conference, Wsc. IEEE, pp 2350-2352, 2008.