Modeling and Simulation of CPS based on SysML and Modelica(KG)

Fei Deng,Yunqiang Yan, Feng Gao, Linbo Wu
Institute of Computer Application
China Academy of Engineering Physics
MianYang,China
fax_caep@163.com

***Abstract*—Cyber-Physical Systems (CPS) is usually associated with large numbers of domains. The domain relevance allows system engineers' knowledge to spread many domains. The efficiency and accuracy of modeling are not able to be guaranteed. Therefore, in this paper, a set of CPS modeling guideline is proposed based on SysML, and the system's 4-layer abstract hierarchy and the kind of modeling products obtained on each layer are defined. Based on this, a modeling specification containing seven sub-processes is designed. In order to verify the correctness of CPS function and the consistency of the business logic at the primary phase of the design, we summarize the mapping rules of SysML-Modelica and define the algorithm of model conversion. Finally, a simple CPS case is used to verify our task. The results show that this method can be effective in CPS's modeling and Simulation.**

Keywords—Cyber-Physical Systems; Modeling ;SysML; Modelica;

I. INTRODUCTION

CPS is organic and in-depth integrity of Computation, Communication, and Control technologies, and a next-generation intelligent system that closely integrates and coordinates computational resources with physical resources [1-3]. CPS integrates such systems engineering as environmental perception, embedded computing, and network communication, closely coordinates computing and physical resources, covering all aspects of social life. As computing technology, network technology and control technology are constantly developing,CPS has become a new trend of research and development of modern information technology. Modeling and simulation are quite significant to the construction of CPS. It can not only verify the CPS at the primary phase of the design but also is an important section of model-based development and testing.

Since CPS is usually associated with multiple domains and the domain relevance allows the system engineers' knowledge to associate with many domains [4]. The efficiency and accuracy of modeling are not able to be guaranteed. Therefore, a set of the modeling guide of the CPS system based on SysML is proposed in this paper, and 4-layer abstract hierarchy of the system and the kind of modeling products obtained on each layer are defined. Based on this, a modeling specification containing seven sub-processes is designed. In order to verify the correctness of CPS function and the consistency of business logic at the primary phase of design, we summarize the mapping rules of SysML to Modelica model and define the algorithm of model conversion. Finally, a simple temperature control system case is used to verify our method.

The structure of this paper is organized as follows: In the second section, the background and the related work are overviewed. In the third section, the overall framework is proposed, and CPS modeling method, model mapping cabinet and conversion algorithm are described in detail. In the fourth section, the case analysis is performed. In the fifth section, summary and prospects are made.

DOI reference number: 10.18293/SEKE2019-167

II. BACKGROUND AND RELATED WORK

A. Cyber-Physical System

The typical CPS architecture mainly includes the following three kinds of parts: sensor, actuator, and controller [6]. The sensor is used to perceive all information of the physical domain. The controller can accept information from the sensor and send orders according to the control logic. The actuator receives a control order and begins to control the physical objects. The operation mechanism among the basic components is shown in Fig. 1.

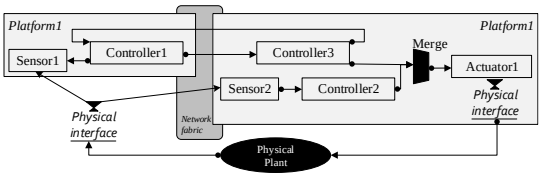

Fig. 1. C typical Cyber-physical system

In the simulation study of the CPS system, Lin Jing et al. from the Missouri University of Technology used agent-based modeling to construct a model of CPS [7]. Based on the service-oriented architecture (SOA), the University of Texas proposed Physical-entity service model for CPS modeling [8]. Frank Wawrzik et al. proposed a SysML-based CPS modeling simulation method SICYPHOS CPS [4].

SysML (System Modeling Language) is a system modeling language extended from UML (United Modeling Language) [5,9-10].Based on the nine types of SysML graphs, system engineering personnel can easily regulate term, model, design, and analysis and verification on the system [11], which is widely used in the industry field. Although SysML is widely used now, from the perspective of system security analysis, the current SysML specification cannot effectively support the dynamic simulation of physical engineering systems. Therefore, Modelica[12-13] will be selected for dynamic simulation of CPS in this paper.

III. MODELING AND SIMULATION OF CPS

Based on the analysis of SysML and Modelica features and the current challenges for CPS simulation and verification, we designed a CPS modeling and simulation framework based on

SysML and Modelica, and realized the SysML-Modelica automatic conversion tool (Fig.3).

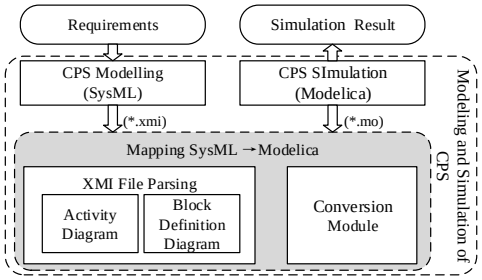

Fig. 2. Logic framework diagram of conversion

The overall logical framework is shown in Fig. 2: Firstly, construct SysML model according to the files and requirements of system design, and export of XMI [14] data model, and the automatic conversion of SysML-Modelica is realized according to the mapping rules of the construction model. Finally, input the Modelica model to verify the correctness of CPS function and the consistency of the business logic.

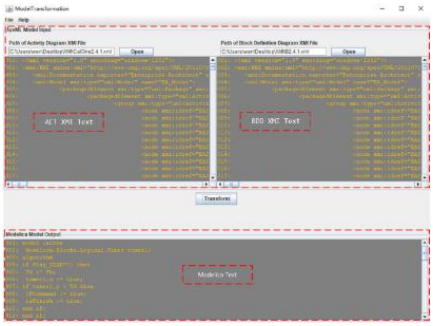

Fig. 3. Model Conversion Tool GUI

The workflow of data conversion processing and conversation in this process is shown in Fig. 4:

1) Use Enterprise Design (EA) to construct the CPS behavior model and structural model, and use export function owned by the EA tool to export the CPS SysML model as the XMI file;

2) Analyze XMI files with Java-based extension tool DOM4J;

3) Convert the extracted information into the grammatical format conforming to the Modelica modeling language, and output Modelica Text semantically equivalent to SysML information;

4) Use OpenModelica to simulate Modelica Text.

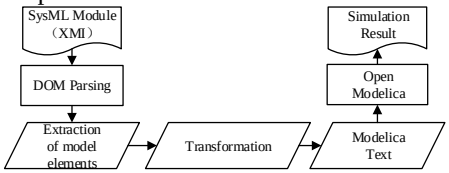

Fig. 4. Flow of data processing

There are two main technical points to achieving this work: (1) CPS modeling: Due to the increasing complexity of CPS, it makes the system modeling process more complex and difficult; (2) The conversion rules of SysML-Modelica modeling languages: Since SysML and Modelica are two different modeling languages, there are differences in grammatical descriptions in many model elements. Therefore, we need to construct a set of semantic equivalent model conversion rules based on model semantics.

A. CPS Modeling

From the above analysis on the structure of the CPS system, it can be clearly known that all physical entities in the CPS only belong to one of these three types. When modeling the static structure, a structural model can be constructed for each entity. After structural modeling, it is required to model the dynamic behavior of each entity. When the industry world constructs the corresponding SysML model, select Block Definition Diagram (BDD) and Internal Block Diagram (IBD) that can express system architecture information, to construct a structure model and use Activity Diagram (ACT), Sequence Diagram (SD), or State Machine Diagram (STM) to express dynamic activity information. Considering the semantic features contained in our Modelica model, mainly use BDD to construct the structure model of CPS and use ACT to construct the activity model of CPS in this paper.

a) Modeling Guidelines

The complex structure and activity of CPS allow the modeling process of the system to be more complicated and difficult to grasp. One of the most effective ways to solving these problems is to construct a hierarchical model for the system. Abstraction layers that are hierarchical and clearly defined can effectively reduce system complexity which helps system modeling personnel to create and manage system models at different levels of granularity. We define the system 4-layer abstract hierarchy, which is the system layer, function layer, software and hardware layer, and deployment layer from top to bottom with an increasing amount of granularity described in each layer.

In the practical modeling, from the primary requirement of CPS, the modeling design process is divided into 7 sub-processes based on the four levels, and in each modeling process, different modeling tasks need completing.

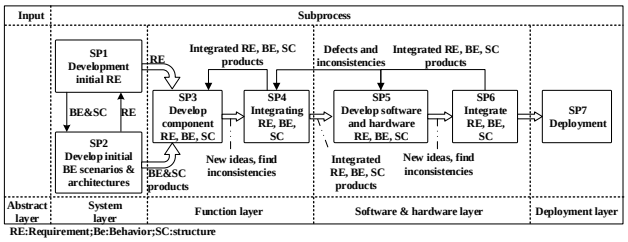

Fig. 5. Modeling process

1) Sub-process SP1 supports the development of primary requirements for system layer. The top-level requirements of the system are described from the angel of system realization according to the system prospect and the needs of relevant stakeholders.

2) Sub-process SP2 supports system layer initial activity scenarios and the development of system structure. The initial behavior scenario describes a coarse- granular workflow; The development of the initial architecture mainly includes limiting system boundaries, defining system interfaces, and linkage between systems and external environment entities.

3) Sub-process SP3: Based on the output of SP1 and SP2, refine and expand the initial requirements, activity scenarios, and architecture, and focus point shifts to the internal system.

This process is a constant iterative course until the system engineering gets a satisfactory result.

4) Sub-process SP4 is responsible for coordinating and integrating the cross-system layer and functional layer as well as the inconsistencies of requirements, activities, and structural products.

5) Sub-process SP5 develops the requirements, activities, and architecture of the hardware and software layer based on the results of SP4 integration. The focus shifts from logical functional components to hardware or software components.

6) Sub-process SP6 is similar to sub-process SP4, in charge of coordinating and integrating across functional layer and hardware layer as well as inconsistencies in requirements, activities, and structural products within the software layer.

7) Sub-process SP7 is responsible for the designing scheme of deploying software and hardware to physical units.

B. XMI file Parsing

The information in the diagram can be read by parsing the XMI file of the active graph and the module definition graph.

The block definition diagram displays the static structure information of the system. Its main element is block. We can obtain the information contained in the entity through parsing block and its value attribute tags, and use Variable and Parameter respectively to save them. The Variable data type is used to store variables information in a block. As shown in Table 1, it includes two attributes: name and type, the name represents the variable name, and type represents the variable type. The Parameter data type stores information of parameters in a block, containing attributes' type, name, and value. The attributes' name and type have the same meaning as the Variable data type, and the value attribute represents the default value of the parameter.

TABLE I. THE MAPPING RELATIONSHIP BETWEEN XMI TAGS AND GRAPH ELEMENTS

Model	Graph element	XMI Label	Define data types	Contains properties
BDD	block	<packagedElement xmi:type="uml:Class">	Variable	type, name
	Value attribute	<ownedAttribute>	Parameter	type, name, value
ACT	Activity zoning	<group>	ActivityPartition	name,classname, id
	node	<node>\<ownedParameter>	Node	id,name,classname,owner,type
	Transfer edge	<edge>\<guard>	Transition	id,source,target,guard

The main elements in the activity diagram include nodes, edges, detection values of the edge, activity parameters, and active partitions. The basic element information corresponds to node, edge, guard, ownedParameter, and group of the label in the XMI file. As shown in Table 1, we store the parsed active graph elements into the following three data types: Node, Transition, and ActivityPartition. The Node data stores information of the control nodes, action nodes, object nodes, and active parameter elements in the SysML activity diagram, which contains 5 attributes in total: ID name, classname, owner, and type, respectively. Among them, type represents the node type and contains owned Parameter. The Transition data type stores information of the edge elements in the SysML activity diagram. Its specific type is shown as follow and contains 4 attributes in total like ID, source, target, and guard. id uniquely identifies one edge, source and target are the id values of the edge source node and the target node, and guard is the monitor value, that is, the transfer condition of the edge. The ActivityPartition data type stores information of active partition elements in the SysML activity diagram and contains three attributes in total like id, name, and classname. Id uniquely identifies an active partition, name is the name of the active partition, and classname is the type of active partition.

Based on the above analysis of the block definition graph and activity graph XMI file and customized data types, the element information under the tags is stored according to the corresponding data type created by traversing each label in the XMI file as required.

C. Mapping Rule of SysML2Modelica

According to the modeling rules for CPS in the previous section, we will obtain a series of block definition diagrams and activity diagrams. After establishing the mapping rule, SysML model can be converted to Modelica model. The element in the block definition diagram is mainly converted to an element in the Modelica model declaration area, and the elements in the activity diagram are mainly converted to elements in the Algorithm area of Modelica model.

The block of the block definition diagram is the basic unit in SysML and corresponds to an instance of the Modelica model, thus establishing the mapping rules for SysML block and Modelica model. The value property of the block maps the variables of Modelica model; The port and components of the block (Instances of other modules) map to the member type in Modelica model. They are directly mapped to Modelica model equation if modular constraints are not combined with other SysML activity diagrams. At this, the structure of the module has been basically built successfully in Modelica model.

When modeling CPS, we use activity charts to represent the activity of CPS. The action in the activity diagram represents processing or transformation, so it is mapped to the equation of Modelica. Determines that the node is mapped to if-else judgment, and the merge node indicates that the above input continues to execute when arriving, and is mapped to continue executing downwards in the code of Modelica model.

The following table summarizes the rules for mapping elements in the above two SysML diagrams to Modelica elements.

TABLE II. MAPPING RULES BETWEEN SYSML AND MODELICA

	SysML	Modelica
BDD	block	model
	interfaceblock	connector
	port node	instance of connector
	value	variable

	constraint	equation
	decision node	if-else
ACT	merge node	execute sequentially
	action	equation
	State	Step
	Transition(no trigger)	Transiton

According to the above mapping rules, the model conversion algorithm is defined according to section 3.4, and the SysML diagram can be directly converted to Modelica code.

D. Model Conversion Algorithm

The conversion algorithm converts the block definition diagram and activity diagram into the Modelica model according to the conversion rules. The input of the algorithm is the XMI file for the block definition diagram and the active graph, and the output is Modelica model. The basic idea as follows: Firstly, use the resolution algorithm of the active graph XMI file to obtain all active partition collections, node collections, and edge collections. Construct a Modelica model for each active partition, and use the XMI file resolution algorithm in block definition graph to obtain the corresponding block variable collection and parameter collection, and then declare these variables and parameters in the model declaration area. If the activity diagram contains latency actions for relative time events, Declare an instance of Timer in the model declaration area. Then, go down from the starting node of the active partition and output different contents in the model algorithm region according to the node type, including selection, circulation, and concurrent structure processing. The specific model conversion algorithm is shown in Table 3 of the arithmetic.

TABLE III. SYSML-MODELICA CONVERSION ALGORITHM

Algorithm: SysML2Modelica

Input：Activity zoning set AP, node set N, edge set T, variable set V, parameter set P
Output：Modelica model
Procedure SysMLtransformModelica(M) Begin
1. for each ap in AP do
2. Create Modelica Model;
3. Export variables and parameter declarations based on V and P;
5. Get the set of all waiting time action nodes: TimerNodes;
6. for each TimerNode in TimerNodes
7. Declare an instance of ModelicaTimer;
8. end for
10. node= Initial node of activity zoning;
11. while(node!=null) do
12. if(node =="Action") then output"node.name";
14. else if(node =="Decision") then output "if + node.name + then";
16. else if(node=="AcceptEvent") then output "if + node.name + then";
18. else if(node=="AcceptEventTimer") then
19. if Relative time events then output "if time > Relative time then";
21. else output "when time > Absolute time then";
23. end if
24. else if(node=="Merge") then
25. if Merge nodes and decision nodes form a cycle then
26. output "while +Decision node name Attribute value+ loop";
27. else output "end if;"
29. end if
30. else if(node=="Fork") then
31. Each concurrent branch continues to call the transformation algorithm;
32. else if(node =="ActivityFinal" or "FlowFinal") then
33. output End statement that match statements if、when
34. break;
35. end if
36. Find the next node of node;
37. node=nextNode;
38. end while
39.end for
End Procedure SysML2Modelica

IV. CASE

In this section, a simple temperature control system [15] is used to illustrate how to model CPS, to convert into the Modelica model by mapping rules and finally conduct stimulation. The temperature control system is a temperature adjustment system involving temperature sensors, air conditioners and switch controllers. The system requires temperature control between 16 ℃ and 28 ℃ and limits the season to be summer. The temperature sensor senses room temperature and transmits the temperature to the switch controller. The controller learns the temperature and sends the coolOn or coolOff order after judging. The air conditioner implements cooling operations according to the corresponding order. If the air conditioning does not work, the temperature will gradually increase as time. Due to the space limitation, in this paper mainly take the switch controller as an example to illustrate.

A. Structure Modeling

The structure analysis on the temperature control system shows that temperature sensor corresponds to sensor entity in CPS, air conditioner corresponds to the actuator entity, and controller corresponds to the controller entity. For each such entity, block is used to define the basic information in the block definition diagram. The temperature sensor has 2 ports with one port perceiving temperature and one port transmitting temperature information to the switch controller. In this case, we require the temperature sensor to learn the temperature directly from the temperature port of the air conditioner. The controller also has two ports, one of which learns the temperature from the temperature sensor as described above, and one port sends order to the air-conditioning. Similarly, the air conditioner has a port for receiving order and a port for transmitting temperature. Besides, air conditioners accept different order and follow different temperature variable equation constraint. Temperature sensors follow only one working equation constraint. At this point, the block definition diagram for each entity and that for the port can be given. The BDD of the system is shown in Fig. 6.

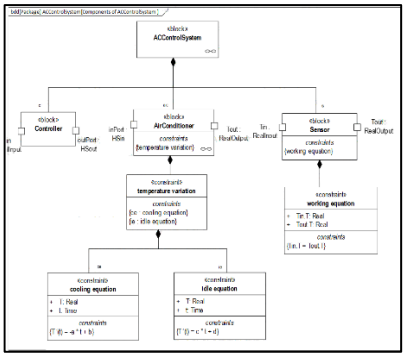

Fig. 6. System Entity Block Definition Chart

B. Behavior modeling

As mentioned above, for controllers SysML activity graph is used to represent activity information, controllers usually have complex control logic. The biggest advantage of activity graph is that they can represent the logical information well.

Fig. 7 is the activity diagram of the switch controller in the temperature control system.

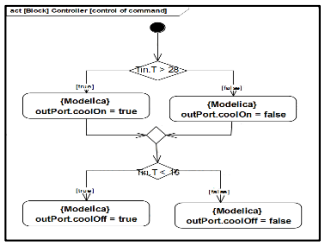

Fig. 7. Activity Chart of Switching Controller

C. SysML-Modelica Model Conversion

Firstly, according to the entity block definition diagram, we know that there are three different entities in the entire system, and each entity corresponds to Modelica model. In terms of the switch controller, combined with the block definition diagram and the activity diagram, according to the SysML-Modelica model mapping rule in Section 3.3, Modelica code shown in Fig. 8 (a) can be obtained.

```
model Controller
RealInput Tin;
HSout outPort;
equation
if Tin.T >= 28 then
outPort.coolOn = true;
else
outPort.coolOn = false;
end if;

if Tin.T <= 16 then
outPort.coolOff = true;
else
outPort.coolOff = false;
end if;
end Controller;
```

```
model ACControllSystem
  Controller c;
  AirConditioner ac;
  Sensor s;

  equation
  connect(ac.Tout,s.Tin);
  connect(s.Tout,c.Tin);
  connect(c.outPort,ac.inPort);
end ACControllSystem;
```

Fig. 8. Modelica code Code for Switch Controllers and Systems

Though temperature sensors and air conditioners belong to different types of CPS entities, alike switch controllers, they can be converted to Modelica code combined with module diagrams and model mapping rules. For all models required for StateGraphics library, automatically a line of Modelica code is required to "addinner Modelica.StateGraph.StateGraphRootstateGraphRoot; ". It is pointed out particularly that the two temperature variance equations are quadratic equations of time t. After the time T is derived, two Timer types are required to add to the Modelica code to calculate the temperature variable. Finally, the important step is to connect all Modelica models according to BDD to form the entire temperature control system. From Fig. 6, we can see the system model Modelica code as shown in Fig. 8.

D. Analysis of Simulation Results

Introduce all model converted code into the OpenModel tool for simulation. We can know the variation of room temperature with the time in the temperature control system, as shown in Fig. 9. The entire room temperature fluctuates directly from 16 ℃ to 28 ℃ according to the switch controller.

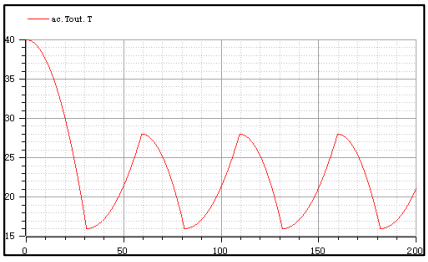

Fig. 9. Temperature variable curve

V. SUMMARY AND DISCUSS

In this paper, firstly the architecture of CPS is analyzed and divide it into three types: sensors, actuators, and controllers. For the two aspects of structure and behavior, CPS is hierarchical modeling with SysML. In order to be able to simulate modeling, the mapping rules between the SysML-Modelica models are summarized. On this basis, the model conversion algorithm is designed, the SysML-Modelica model automatic conversion tool is realized, and an case analysis is conducted by the temperature control system. Although in this article the CPS modeling and simulation process based on SysML and Modelica has been realized, there are still shortcomings. We will further study the following two aspects: 1) Extend SysML-Modelica mapping rules to support automatic conversion of more SysML-Modelica models; 2) Provide more modeling language interfaces, such as SystemC. Establish a unified modeling and simulation framework for CPS that can be applied to more domains.

REFERENCES

[1] Lv, C., et al. "Simultaneous Observation of Hybrid States for Cyber-Physical Systems: A Case Study of Electric Vehicle Powertrain. " IEEE Transactions on Cybernetics 48.8(2018):1-11.

[2] Zanero, Stefano. "Cyber-Physical Systems."IEEE Computer50.4 (2017): 14-16.

[3] Mo, Haining, Neeti Sharad Wagle, and Michael Zuba. "Cyber-physical systems."ACM Crossroads Student Magazine20.3 (2014): 8-9.

[4] Wawrzik F, Chipman W, Molina J M, et al. Modeling and simulation of Cyber-Physical Systems with SICYPHOS[C] International Conference on Design & Technology of Integrated Systems in Nanoscale Era. IEEE, 2015.

[5] OMG, "Systems Modeling Language (SysML) specification," OMG standards, formal/2013-06-01

[6] Modelica by Example. http://book.xogeny.com/

[7]]Lin J, Sedigh S, Miller A. A General Framework for Quantitative Modeling of Dependability in Cyber-Physical Systems: A Proposal for Doctoral Research[J]. 2009, 1:668-671.

[8] Huang J, Bastani F, Yen I L, et al. Extending service model to build an effective service composition framework for cyber-physical systems[C]// IEEE International Conference on Service-Oriented Computing and Applications. 2009:1-8.

[9] Friedenthal S, Moore A, Steiner R. A Practical Guide to SysML[J]. San Francisco Jung Institute Library Journal,2013, 17(1):41-46.

[10] Delligatti, Lenny. SysML Distilled: A Brief Guide to the Systems Modeling Language. 2013.

[11] Weilkiens T. Systems engineering with SysML/UML: modeling, analysis, design[M]. Morgan Kaufmann, 2011

[12] Fritzson P. Principles of object-oriented modeling and simulation with Modelica 2.1[M]. John Wiley & Sons, 2010.

[13] Introduction to physical modeling with Modelica[M]. Springer Science & Business Media, 2012.

[14] Kovse J, Härder T. Generic XMI-based UML model transformations[C]//International Conference on Object-Oriented Information Systems. Springer, Berlin, Heidelberg, 2002: 192-198.

[15] Chen X, Ye R, Sun H, et al. Deriving Requirements Specification with Time: A Software Environment Ontology Based Approach[J]. 2013:431-43