Tai-hoon Kim
Hyun-seob Cho
Osvaldo Gervasi
Stephen S. Yau (Eds.)

Communications in Computer and Information Science 351

Computer Applications for Graphics, Grid Computing, and Industrial Environment

International Conferences, GDC, IESH and CGAG 2012
Held as Part of the Future Generation
Information Technology Conference, FGIT 2012
Gangneug, Korea, December 2012, Proceedings

Springer

Table of Contents

Metamodel Design for Model Transformation from Simulink to ECML in Cyber Physical Systems

Hyun Seung Son[1], Woo Yeol Kim[2], Robert YoungChul Kim[1], and Hang-Gi Min[1]

[1] College of Science & Technology, Hongik University, Sejong Campus, 339-701, Korea
son@selab.hongik.ac.kr, {bob,hgmin}@hongik.ac.kr
[2] Dept. of Computer Education,
Daegu National University of Education Daegu, 705-715, Korea
john@dnue.ac.kr

Abstract. It popularly uses Simulink to design embedded system on Model oriented design platform, and to simulate dynamic system for multi-domain simulation. So, Electrics Telecommunication Research Institute (ETRI) is researching for model transformation from Simulink model to ECML model for Cyber Physical Systems Project. This paper shows how to transform Simulink model to ECML model. We propose an automatic transformation method to transform ECML model from Simulink model based on model transformation technique that is very useful mechanism for heterogeneous model. And, we also design each metamodel of Simulink and ECML. With these metamodels of them, it easily execute model transformation even with models of any vendor's tools.

Keywords: Cyber Physical Systems(CPS), Metamodel, Meta Object Facility (MOF), Electrics Telecommunication Research Institute(ETRI) CPS Modeling Language (ECML).

1 Introduction

ETRI CPS Modeling Language (ECML) is modeling language developed for Cyber Physical Systems (CPS) environment in Electronics and Telecommunications Research Institute (ETRI) [1]. ECML is modeling language for hybrid systems that contains continuous elements such as the physical, electrical, electronic analog and discrete element such as digital electronics, software [2].

Simulink for multi-domain simulation of dynamic system and model-based design provides a set of interactive graphical environment and customizable block libraries. Simulink is also very useful to support a decision of design concept with minimum of effort on some software phase such as developing and testing rapid virtual prototype. Many engineers are repeatedly design and execution using Simulink before prototype development [3]. Because of the advantages of Simulink, ETRI is researching to use the Simulink model in ECML [4].

T.-h. Kim et al. (Eds.): GDC/IESH/CGAG 2012, CCIS 351, pp. 56–60, 2012.

To be able to use the Simulink model in ECML, we suggest applying model transformation. Model transformation is transforming heterogeneous different models based on metamodels [5]. But this model transformation requires model transformation method and design of each metamodel between model of Simulink and ECML. This paper shows our proposed transformation method from Simulink to ECML and design of metamodels between them. This proposed method has three steps: first, translates file of Simulink model into file of XML, then translates file of XML into file of XMI, and finally transforms XMI of Simulink into XMI of ECML. But metamodel of Simulink is not existed for model transformation. Therefore, we design metamodel based on XML file of Simulink, and provide foundation for model transformation.

This paper consists of the following chapters. Chapter 2 explains a related study. Chapter 3 explains about method of model transformation from Simulink to ECML. Chapter 4 explains about metamodel design of Simulink model. Chapter 5 provides conclusion and future works.

2 Related Work

Metamodel is model to express model. In other words, this is mechanism for definition to express abstract model of actual worlds. Therefore, metamodel clearly describes necessary constructs and rules to organize specific models in concern domain. Metamodel is shown from three different perspectives: First, in order to build a model that is used building blocks and a set of rules, Second, model of concern domain, Third, instance of other model. Metamodel of Simulink and ECML represents to use Meta Object Facility (MOF)[6] as expressive method of metamodeling.

MOF that is establishing OMG standard consists of definition language of metamodel and framework for repository management of metadata. This MOF is used such as metamodel of UML, Common Warehouse Metamodel (CWM)[7], Model Driven Architecture (MDA)[8], and other metamodels. MOF ensures interoperability within the scope, which is defined metamodel on standard.

3 Model Transformation from Simulink to ECML

In order to transform model of Simulink, we propose one method as Figure 1. This method consists of two phase. First, to use a command name that is "save_system" in Matlab translates file of MDL of Simulink into file of XML. Translated file of XML includes all information of Simulink. In other words, this includes a Simulink model in addition to information such as solver, structure, editor, GUI. Therefore, we remove unnecessary information. In addition, XML-based Metadata Interchange (XMI)[9] that is input file used model transformation is translated by XMI translator because Simulink XML is not match to XMI.

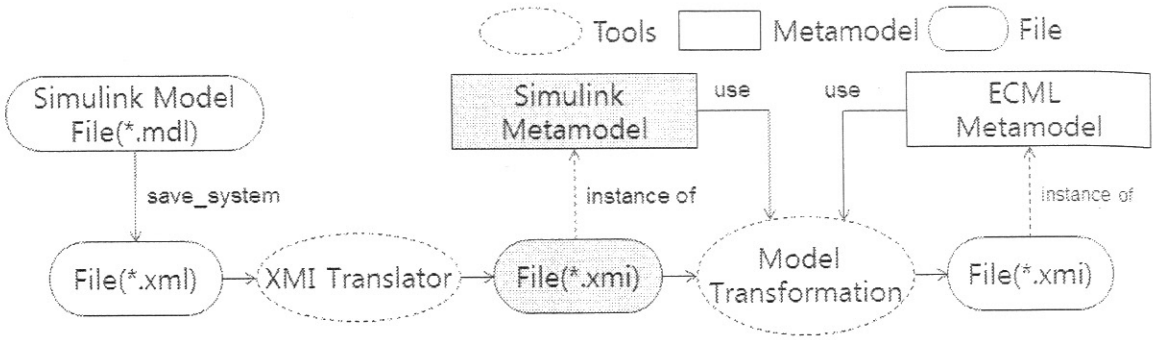

Fig. 1. Method of Model Transformation from Simulink to ECML

Second, to use model transformation transforms translated file of XMI of Simulink into file of XMI of ECML. But, in order to model transformation technique, metamodel is designed by designer. Metamodel of Simulink is not existed for model transformation. So, we design metamodel based on XML file of Simulink. Metamodel of ECML just is used with the contents of our previous research [1]. Model Transformation is writing transformation rules with analyzing similarities and differences of Simulink and ECML.

4 Metamodel Design for Simulink Model

In order to design metamodel of Simulink, we can choose two ways as follows: The way is newly to define with ignoring the structure of existed file of XML of Simulink

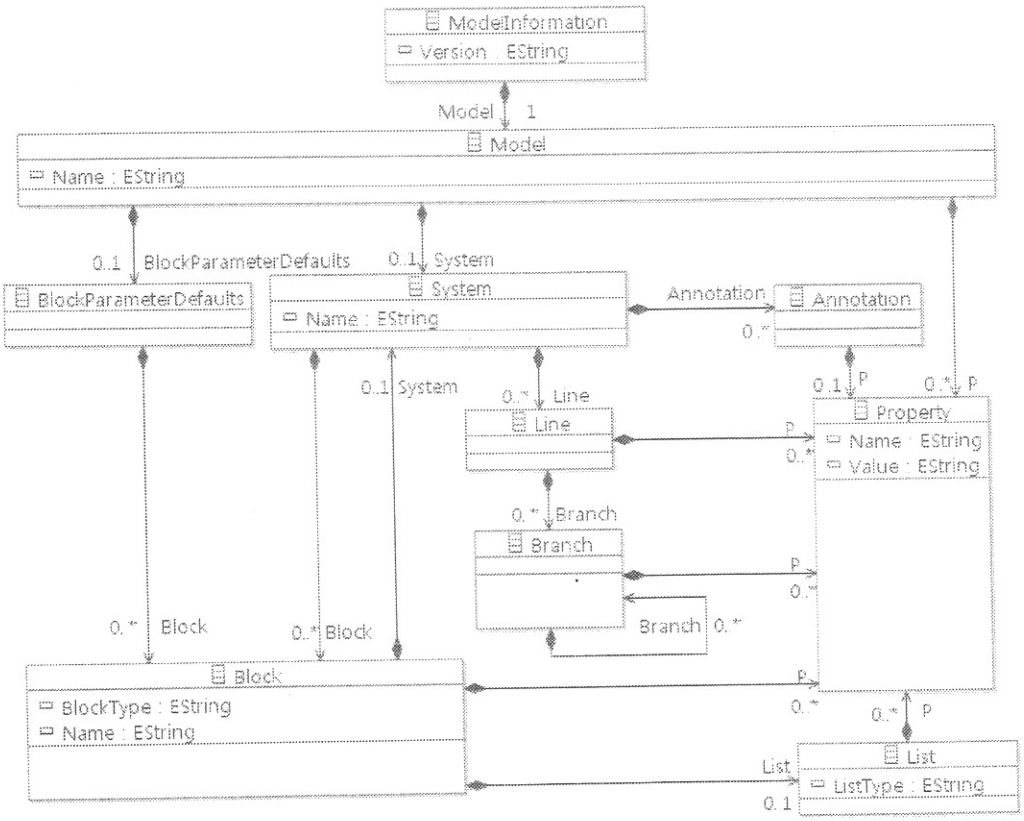

Fig. 2. Metamodel of Simulink Model

or to define the metamdoel based on file of XML of Simulink. The first method is useful to define structure of metamodel more semantically. But there is a problem that file of XML must be translated once more. The second method is useful to reuse existed file of XML of Simulink without almost converting. But metamodel based on file structure is very difficult to understand the meaning of the metamodel at model transformation. Because the existing file structure of XML of Simulink is originally not designed as a metamodel perspective, and just purpose for saving the file.

But this paper designs metamodel with the second method. The reason is why it easily understands and is organized simple structure for file of XML of Simulink. So we can simply design structure of metamodel for Simulink model. Also it should do unnecessary works to translate the existed file of XML of Simulink into form of metamodel, and has not advantage even through redesigning metamodel of simulink. As a result, we did reverse design based on the existed file of XML of Simulink.

Figure 2 is metamodel designed with reverse engineering based on file of XML of Simulink. *ModelInformation* as a root node has a *Model*. *Model* consists of *BlockParameterDefaults*, *System*, and *Property*. *BlockParameterDefaults* and *System are* shared with *Block*. *Block* includes *Subsystem*. *System* consists of *Block*, *Line*, and *Annotation*. *Model*, *Line*, *Annotation*, *List*, and *Block* share with *Property*.

5 Conclusion

This paper shows our proposed model transformation method, and designs metamodel of Simulink. This proposed model transformation is translating designed model of Simulink into file of XML, and then translating file of XML into file of XMI. So, model transformation executes to use metamodel of Simulink and ECML. But in order to execute this model transformation, metamodel is required. Metamodel of Simulink is not existed for model transformation. So, we design metamodel based on XML file of Simulink and provide a foundation for model transformation.

Further research on modeling transformation involving language and engine should be conducted in the future, which is not dealt in this study.

Acknowledgments. This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS(Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"] and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2012-(H0301-12-3004)).

References

1. Jeon, J., Chun, I., Kim, W.: Metamodel-Based CPS Modeling Tool. In: Park, J.J(J.H.), Jeong, Y.-S., Park, S.O., Chen, H.-C. (eds.) EMC Technology and Service. LNEE, vol. 181, pp. 285–291. Springer, Heidelberg (2012)

2. Lee, H.Y., Chun, I., Kim, W.-T.: DVML: DEVS-Based Visual Modeling Language for Hybrid Systems. In: Kim, T.-h., Adeli, H., Stoica, A., Kang, B.-H. (eds.) CA/CES^3 2011. CCIS, vol. 256, pp. 122–127. Springer, Heidelberg (2011)
3. MATLAB/Simulink, http://www.mathworks.com
4. ETRI, Cyber-Physical Systems research, http://www.etri.re.kr/etri/res/res_05020102.etri
5. Czarnecki, K., Helsen, S.: Feature-based survey of model transformation approaches. IBM Systems Journal 45(3), 621–645 (2006)
6. OMG, Mata Object Facility, version 2.4, formal/2010-12-08
7. OMG, Common Warehouse Metamodel, version 1.1, formal/2003-03-02
8. OMG, MDA Guide Version 1.0.1 (2003), http://www.omg.org/cgi-bin/doc?omg/03-06-01
9. OMG, MOF 2.0/XMI Mapping, version 2.1.1, formal/2007-12-01

Communications in Computer and Information Science

The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers and abstracts reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are considered for publication as well.

The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, as printed books and/or electronically; furthermore CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library.

The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/files, authors have to strictly adhere to the Springer CCIS Authors' Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

Detailed information on CCIS can be found at www.springer.com

ISSN 1865-0929

ISBN 978-3-642-35599-8

springer.com