http://www.kkits.or.kr

International Conference on Computer Convergence Technology ICCCT 2011

20~22 October, 2011
Seoul, Korea

KKITS
nipa National IT Industry Promotion Agency
KOREA TOURISM ORGANIZATION
SEOUL METROPOLITAN GOVERNMENT
HYUNDAIU&I

Emerging and Innovative Convergence Technologies and Applications

International Conference on Computer Convergence Technology (ICCCT) 2011, October 2011

Design of MOF based Hardware Component Modeling Language for Heterogeneous Embedded Systems

Woo Yeol Kim[1] Hyun Seung Son[1], Junbeom Yoo[2], and Robert Young Chul Kim[1]
[1] *Dep. Of CIC(Computer and Information Communication), Hongik University*
Jochiwon, Korea
E-mail: {john,son,bob}@hongik.ac.kr
[2] *Department of Computer Science and Engineering, Konkuk University*
Seoul, Korea
E-mail: jbyoo@konkuk.ac.kr
*Corresponding author: Robert Young Chul Kim

Abstract

As software development methodology based on MDD (Model Driven Development) facilitates the development of software that is independent from hardware modification, studies attempting to apply this method in Embedded SW development are being actively progressed. Since hardware information used in SW development must be maintained in profile form to achieve proper Embedded SW development based on MDD. This paper proposes HCML (Hardware Component Modeling Language) as a possible solution. In so doing, this language can effectively achieve relative definition. HCML uses both UML Class and Component diagram to effectively define hardware profile.

***Keywords*:**, UML, MDD, Hardware Modeling, Heterogeneous Embedded Systems

1. Introduction

As embedded software is dependent on hardware, users are confronted with the inconvenience of having to redesign it whenever the hardware is changed [1]. MDD (Model Driven Development) has applied in existing studies [2,3,4] that attempt to increase the reusability of Embedded SW. This technique allows common hardware-independent models to be developed that can be reused. It also allows defined hardware profile information to automatically form a hardware-dependent model.

UML (Unified Modeling Language)[5], is an integrated modeling language. It has been applied extensively in embedded software details and designs [6,7,8]. If the definition of hardware profile information can be achieved using UML techniques, a hardware-independent model may be transformed into UML based hardware-dependent models.

This paper proposes using HCML (Hardware Component Modeling Language) as a modeling language for detailing hardware profiles used in MDD-based model transformations. There are several reasons for this application. First, HCML uses UML based component diagram and class diagram to achieve graphic-based expressions of hardware information. The hardware component is an abstract

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0004203) and the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation.

item of the actual hardware, and the component function expresses improved class diagram. Secondly, organic connections with ATL (ATLAS Transformation Language) and inter-operations with a UML meta-model can be achieved if the MOF (Meta Object Facility) incorporates HCML.

In this paper, Chapter 2 reviews related work, including MOF; Chapter 3 describes the proposed HCML notation; Chapter 4 explains one case where HCML was used to achieve hardware component modeling in a heterogeneous embedded system; and, Chapter 5 presents concluding comments and suggestions for future research.

2. Related Work

MOF is the meta object management technique proposed by OMG [9]. MOF is expressed through UML while the meta-model of UML is defined through MOF. MOF is composed of a four layered meta-data structure. These include an information layer, a model layer, a meta-model layer, and a meta-metamodel layer. The information layer is composed of pure data that we wish to describe. The model layer is composed of metadata that explains the data in the information layer. The meta-model layer describes both the definition of the structure and the meaning of the metadata (meta－metadata). The metamodel is thus an "abstract language" for describing a different type of data. As such this language does not possess concrete grammar or characters. The meta－metamodel layer describes the structure and meaning of the meta－metamodel. This is an "abstract language" for defining a different type of metadata [10].

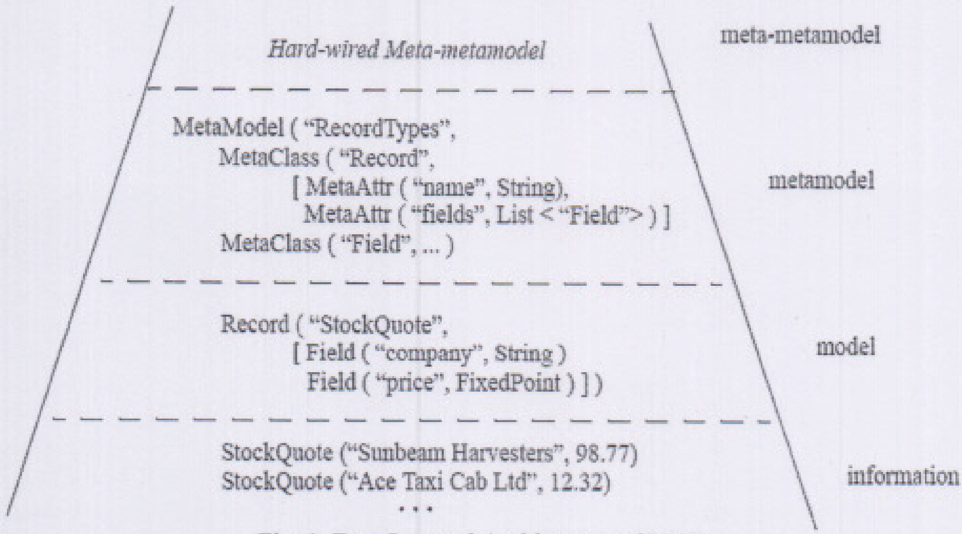


Fig. 1. Four Layered Architecture of MOF

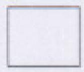
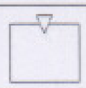
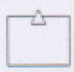
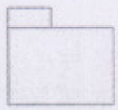
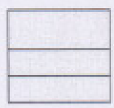
Figure 1 shows the representative four layered metamodeling framework. As presented in the figure, the StockQuote type describes many StockQuote instances in the information layer while the RecordTypes metamodel can depict many record types in the model layer. In the same manner, the meta－metamodel level can describe many other metamodels within the metamodel itself.

The typical four layered metadata structure offers many advantages when compared with the simple modeling method. If the framework is well-defined in design, the following is provided: support for the model and modeling paradigm that are expected to be realized, permission for different types of related metadata to be accessed, permission for an added metamodel and new types of metadata to be provided, and support for exchange between groups using voluntary metadata (model) and meta－metadata (metamodel). Thus, the transformation of metamodels into various models can eventually be achieved and standardized through the use of MOF.

3. Notation of HCML(Hardware Component Modeling Language)

Table 1 shows the notation of HCML. The characteristics of each notation are as follows: the Port is a device used for interfacing with an external HCP, while HCP refers to the hardware module used. As the abstract model of HCI, HCR therefore represents the hardware resources. HCI includes register information for hardware control and API for driving. The Control method used is described in code form.

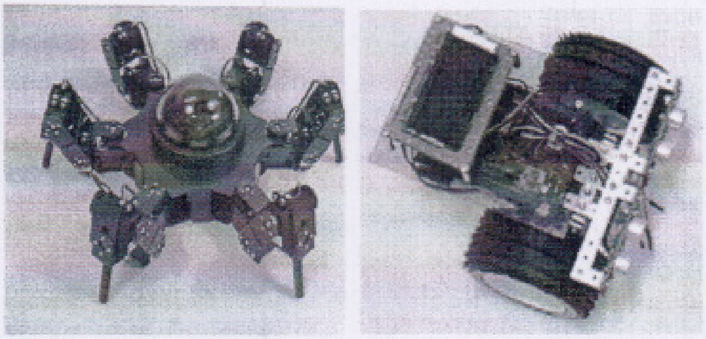
Table 11. Notation of HCML

Node Type		Notation	Comment
Port	IN/OUT		Component port for input and output
	IN		Port for receiving input signal from other components
	OUT		Port for sending output signal to other components
HW Component Package			Component module with abstract hardware
HW Component Resource			Abstract resource for controlling hardware
HW Component Information			Class describing information on hardware

When observing each component, it can be seen that the output direction of the Port can be arranged in 3 levels: in, out, in/out. This output direction setup is applied to in/out port register control to automatically establish setting information. Furthermore, the Port connects a given HCP with another HCP as well as the given HCP with the HCR. Since HCP is the component module formed from abstract hardware, it consists of the HCR and the Port. The Port manages input/output with the outside, while the HCR defines the role and function of the hardware module. In this manner, HCR has one function to perform and represent the module composed of abstract inherent hardware functions, namely the control of hardware, HCR is composed of HCI. HCI is basically identical to class in the class diagram. Hardware control register information is saved as properties, while API control information are saved as methods. Furthermore, a layer structure is used to divide the abstract level.

4. Case Study

Figure 2(a) shows the Hexapod robot used as an application, while Figure 2(b) presents the Javelin model. As a multi-joint robot, the Hexapod is designed to handle rugged topography like mountains, rather than even topographical terrain. By contrary, the Javeline can only move where even topography is present, through its wheels can achieve quick and easier movements than its counterpart, the Hexapod.



(a) Hexapod sensor robot (b) Javeline

Fig. 2. Device Examples

Figure 3 shows the modeling of the Hexapod through the use of HCML notation. The Hexapod multi-joint robot uses 18 ports from PA0~PC7 by the way of the Atmega128. It also uses Timer0, Timer1, and Timer2 to send control signals to each motor. In addition, UART1 in the Interrupt0 connects the ultrasound sensor in Bluetooth to the control hardware. In this regard, the hardware composition and device information can be easily comprehended through HCML modeling.

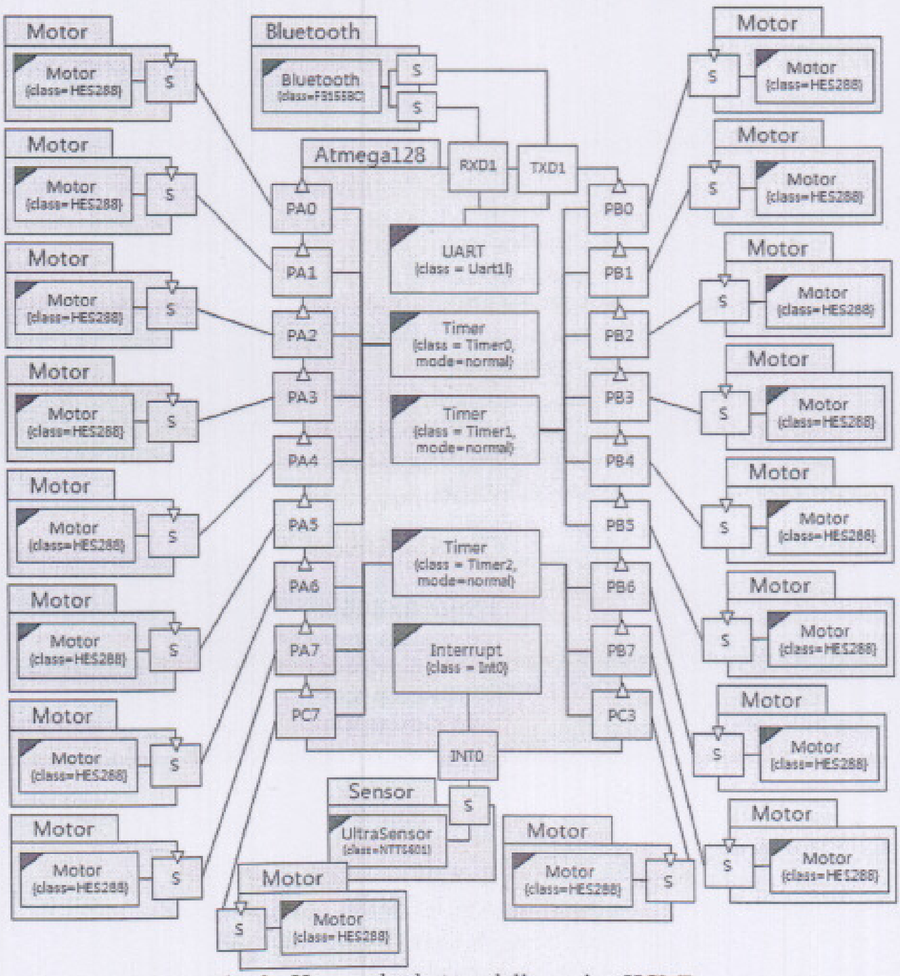


Fig. 3. Hexapod robot modeling using HCML

Figure 4 shows the results of Javeline modeling by incorporating HCML. Javeline uses wheels as its means of transportation, and is operated by 2 motors. For this reason, simple modeling is presented. The Ubicom SX48AC is composed of 16 pins of which P6~9, and P15 are primarily used. That is, P6 and P7 are used to control the ultrasound sensor; P8 and P9 are used to control the servomotor; P15 is used to control the LCD. Furthermore, Javeline uses the UART Timer to control the hardware as an MCU function.

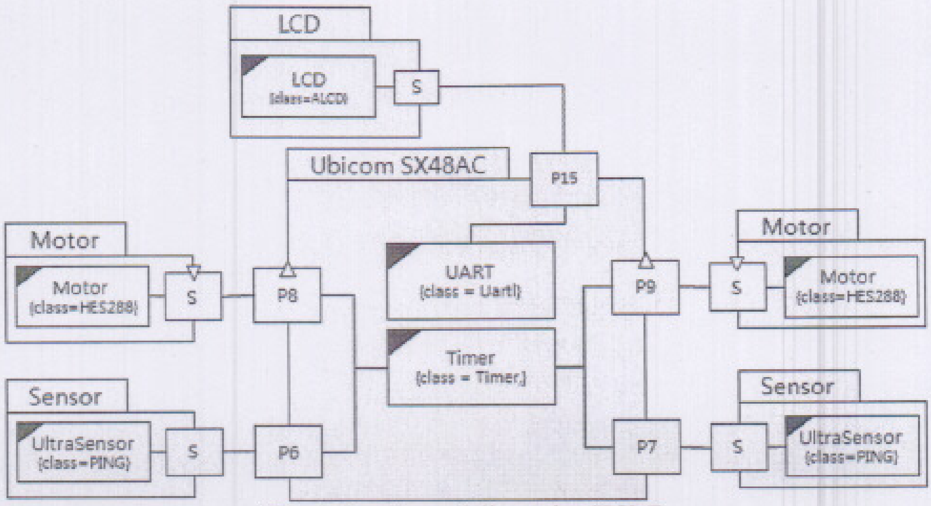


Fig. 4. Javelin modeling using HCML

5. Conclusions

In this paper, we propose HCML (Hardware Component Modeling Language) as a modeling language for detailing hardware profiles used in MDD-based model transformations. HCML with UML notation based on standardizing Software Design uses UML based component diagram and class diagram to achieve graphic-based expressions of hardware information.

It will be possible to work the interoperability and UML meta-model through defining the MOF (Meta Object Facility) based HCML. The hardware component is an abstract item of the actual hardware, and the component function expresses improved class diagram. Through developing heterogeneous HW component profiles with Hexapod robot and Javeline, we have experienced to develop easier than the existing hardware design.

References

[1] David E. Simon, An embedded software primer. *Addison-Wesley*, 1999.

[2] Woo Yeol Kim, Hyun Seung Son, R. Young Chul Kim, C. R. Carlson, "MDD based CASE Tool for Modeling Heterogeneous Multi-Jointed Robots," *Proceedings of the 2009 World Congress on Computer Science and Information Engineering, IEEE Computer Society*, Los Angeles/Anaheim, USA, Vol. 7, 775-779, 2009.

[3] Woo Yeol Kim, Hyun Seung Son, Young B. Park, Byung H. Park, C. R. Carlson, R. Young Chul Kim, "The Automatic MDA (Model Driven Architecture) Transformations for Heterogeneous Embedded Systems," *Proceedings of The 2008 International Conference on Software Engineering Research and Practice*, Las Vegas Nevada, USA, Vol. 2, 409-414, 2008.

[4] Hyun Seung Son, Woo Yeol Kim, Robert Young Chul Kim, "Semi-Automatic Software Development based on MDD for Heterogeneous Multi-Joint Robots," *Proceedings International Symposium on Control and Automation 2008*, Hainan Island, China, 93-98, 2008.

[5] OMG, Unified Modeling Language Specification Version 1.4.2. ISO/IEC 19501, formal/05-04-01, http://www.omg.org/, 2005.

[6] Woo Yeol Kim and R. Young Chul Kim, "Adapting Model Driven Architecture for Modeling Heterogeneous Embedded S/W Components," *International conference on Hybrid Information Technology(ICHIT2006), IEEE Computer Society*, Vol. 2, 705-711, 2006.

[7] Woo Yeol Kim, Hyun Seung Son and R. Young Chul Kim, "Design Automation for Heterogeneous SUGVs with UML Profile Mechanism," *Journal of KIISE : Software and Applications*, Vol. 35, No. 12, 705-715, 2008.

[8] Woo Yeol Kim, R. Young Chul Kim, "A Study on Modeling Heterogeneous Embedded S/W Components based on Model Driven Architecture with Extended xUML," *The KIPS Transactions : Part D*,Vol. 14-D No. 1, 83-88, 2007.

[9] OMG, Meta Object Facility (MOF) Specification Version 1.4.1. ISO/IEC 19502, formal/05-05-05, http://www.omg.org/, 2005.

[10] Atkinson, C. and Kuhne, T., "Model-driven development: a metamodeling foundation," *Software, IEEE*, Volume: 20 Issue: 5, 36-41, 2003.