Printed in Japan

ISSN 1343-4500 (print)
ISSN 1344-8994 (electronic)

INFORMATION

An International Interdisciplinary Journal

Volume 15 Number 5, May 2012

Published by International Information Institute
www.information-iii.org

INFORMATION
Volume 15, Number 5, pp.2067-2078
ISSN 1343-4500

HCML: An MOF-based Hardware Component Modeling Language for Profiling Heterogeneous Embedded System

Woo Yeol Kim*, Hyun Seung Son*, Junbeom Yoo**, and Robert Young Chul Kim*
*Dept. Of CIC(Computer and Information Communication), Hongik University
Sejong Campus, 339-701, Korea
E-mail: {john,son,bob}@hongik.ac.kr

** Dept. of Computer Science and Engineering, Konkuk University
Seoul, 143-701, Korea
E-mail: jbyoo@konkuk.ac.kr

Abstract

In the traditional hardware development ways, hardware engineers only focus on hardware oriented approaches with datasheets (hardware control spec. information) for hardware control, and difficult to use the same hardware code into heterogeneous hardware. But we software engineers try to develop hardware mechanisms in a little different way: MOF (Meta Object Facility) based on hardware component modeling language for representing and modeling the complicated and diverse hardware information. The original MDA (Model Driven Architecture) paradigm helps easily and quickly to develop "each software" for heterogeneous platform (CORBA, Java, .Net), but not for embedded systems. So, to adopt heterogeneous embedded software development with MDA mechanism, we need to construct hardware profiling when target specific model (TSM) should be transformed with target independent model (TIM) for developing embedded software. This paper suggests HCML (Hardware Component Modeling Language) easily to develop hardware profiling with hardware information, which is defined based on UML class diagram and component diagram. We can effectively represent hardware resources such as Port, Timer, and UART of MCU (Micro Controller Unit), with our proposed HCML. Our HCML is possible to work with Model transformation language, ATL (ATLAS Transformation Language).

Key Words: MOF(Meta Object Facility), UML(Unified Modeling Language), HDL(Hardware Description Language), MDD(Model Driven Development), Hardware Modeling, Hardware Profile, 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.

Information required in the development of embedded software is saved as a hardware profile. This profile must include detailed information, such as the CPU pins and memory addresses that can be connected to hardware devices. These values must be inserted into the registers to remove hardware device. In addition, the hardware profile must contain data for connecting hardware components. In this regard, it plays a very important role in the development of MDD-based embedded software. Unfortunately, modeling/detail language for effectively expressing and defining hardware profile information has yet been accurately developed. Thus, a proper hardware profile modeling/detail language must first be developed to achieve effective and reusable embedded software [5,6,7,8,9].

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 [10] based component diagram and class diagram to achieve graphic-based expressions of hardware information. The hardware component is an abstract item of the actual hardware, and the component function expresses improved class diagram. Specially, it is possible to effectively to represent API (Application Program Interface) connected with software, information of register to control hardware, relationships with hardware resources such as Port, Timers, and UART of MCU. HCML can represent Metamodel with the structured information based on UML, and consists of graphic notation and XMI. 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)[11] incorporates HCML. Thirdly, HCML is suitable for the development of firmware-standard software, which has yet to be achieved, particularly in software platforms and operation systems. Lastly, as primary goal of this study, HCML can be applied in the automatic transformation of a target-independent model or target-dependent model into a MDD-based dual embedded development process.

In this paper, Chapter 2 reviews related work, including MOF and existing HDL (HW Description Language); Chapter 3 describes the proposed HCML notation and the metamodel structure; 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

2.1 MOF (Meta Object Facility)

MOF is the meta object management technique proposed by OMG. 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 [12].

2.2 Existing HDL(Hardware Description Language)

Hardware description language (HDL), such as Verilog [13] and VHDL, is effective when designing and simulating hardware. However, it presents many weaknesses when applied to complex data processing functions or system standard designs [14]. In particular, it is difficult to achieve design by only using an existing hardware description language. For example, realizing semiconductor chips from complex functions with built-in processing core, such as SoC (System-on-a-Chip)[15]. This concern becomes problematic VHDL (VHSIC Hardware Description Language)[16], which is a hardware description language that is used in design automation of digital circuits. VHDL facilitates a very wide range of designs, from Behavioral Descriptions of a System Level to Gate Level. Thus, Behavioral description, RTL (Register Transfer Level) description, and Gate Level description can be achieved by using VHDL. However, as VHDL is a complex language, considerable time and effort is required to achieve actual understanding and use of the language. Its structure is difficult for regular software developers to understand. Furthermore, as the purpose of VHDL is to aid in chip design, it is unable to describe hardware control methods.

3. HCML: Hardware Component Modeling Language

This chapter analyzes how a HW Component Profile is applied in a development process. It also presents MOF-based HCML, notation and relative cases. By using the proposed HCML method, simple modeling of complex hardware profile information can be achieved.

3.1 Architecture of HW Component Profile

As the metamodel of HCML is designed by MOF, the existing UML diagram can easily be transformed and expanded. Figure 1 schematizes how the HW Component Profile designed by HCML is applied to the model transformation process of the existing MDD development method [2,3,4,17].

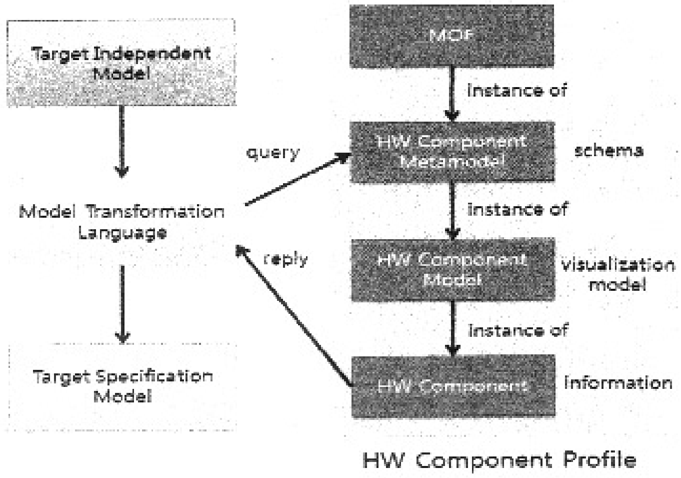


Fig. 1. Four Layered Architecture of HW Component Profile

Model transformation language is required to transform TIM, which is independently formed in hardware, into TSM, a subordinate model, in hardware. Also, a hardware profile is needed since model transformation language can only assist in the transformation between models, but cannot provide hardware information. If only data is saved in the HW Component Profile, it becomes very difficult to use in model transformation language. Thus, MOF-based HCML is more effective instrumental in structuralizing hardware information.

3.2 Graphical notation of HCML

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.

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 is saved as methods. Furthermore, a layer structure is used to divide the abstract level.

Table 1. 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

HCML provides measures and methods for constructing a hardware profile. The four Layer Architecture of the MOF is used to map the relationship between the metamodel, the model, and the information received. The Metamodel uses information and notation to define the model. Although the metamodel cannot be seen by users, it expresses structural information of the model. The Model in turn is the level where users can access graphical notation and achieve 1:1 mapping with information. Furthermore, although information from the metamodel can be used by XML, it is also the advantageous in locating information through XML technologies such as XPath. The relationship between the HCP and the metamodel is presented in Figure 2. The HCP is expressed as a model layer, and the figure shows mapping achieved using metamodels, such as HCP, HCR, Port, ElementProperties.

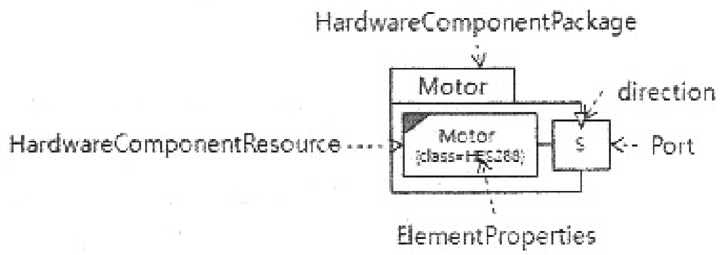


Fig. 2. Relationship between the HCP and the metamodel

Furthermore, a given HCP is connected with another HCP of identical type through the Port. The connected line is called the Connector while the Association and AssociationEnd are set up as related ports. It is described shown in Figure 3.

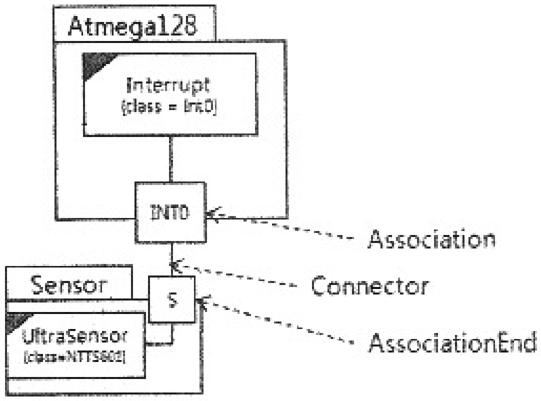


Fig. 3. Relationship between two HCP (Sources/Components)

The HCR is defined by the internal part of the HCP. One HCR possesses one HCI and is expressed as the form presented in Figure 4. The Timer HCR is defined as Timer class, Timer0 class, and Timer1 class. Register and interface information can be placed in each class. The HCI is composed of Classes while each Class is composed of Attribute, Operation, and ElementProperties. The important point of HCI is that the hardware control register related with hardware resource is managed in the UML class diagram. Thus, various hardware control registers can be expressed simultanously.

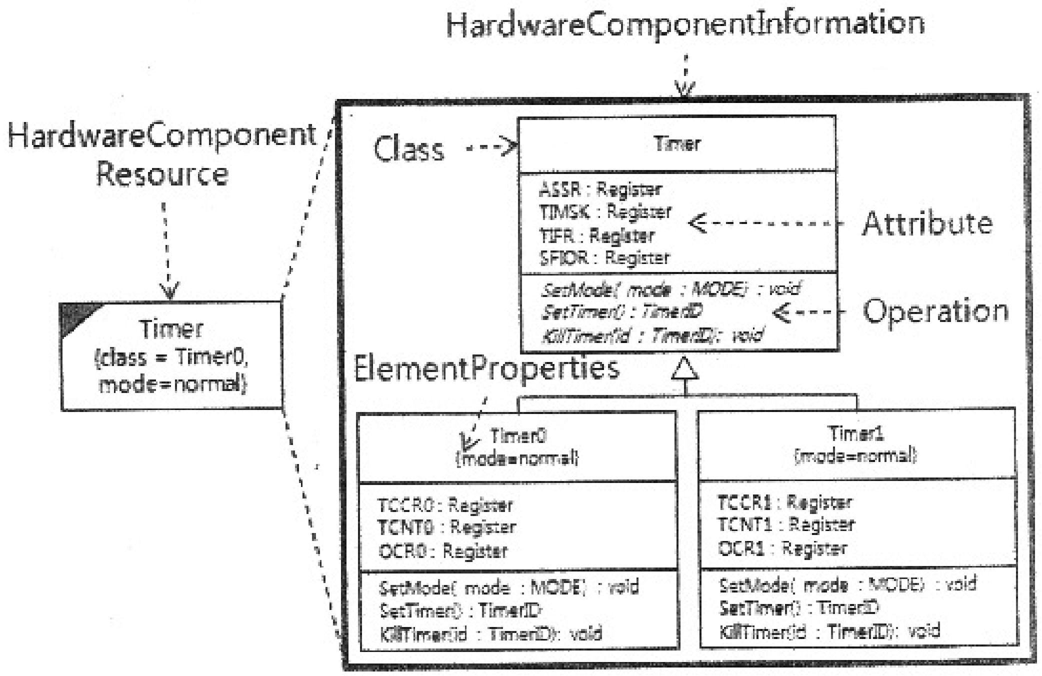
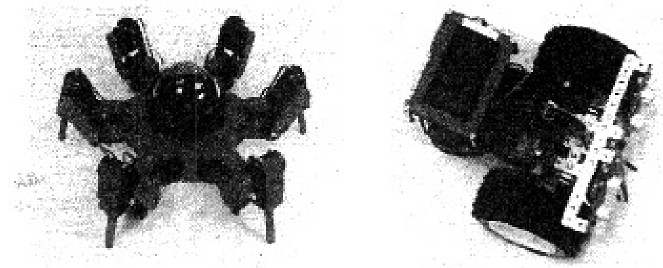


Fig. 4. Relationship between the HCR and the HCI

4. Case Study

4.1 Hardware Composition Information

Figure 5(a) shows the Hexapod robot used as an application, while Figure 5(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. 5. Examples of Devices

Hexapod sensor robot is composed of an Atmega128 MCU with 3 joints per leg and a total of 18 servomotors. Bluetooth wireless technology is used to achieve real-time control. In addition, one ultrasound sensor is used to detect surrounding objects. The MCU used in hexapod multi-joint robot is very small and contains an 8bit processor. For this reason, it was necessary to use C language during development, while object-oriented languages, such as C++, Java, cannot be used. By contrast, the Javeline is composed of Ubicom SX48AC, 2 motors, and 2 sensors. As a wheel based robot, it uses a smaller number of motors than the Hexapod. Furthermore, the Javeline contains a Java virtual machine built into the MCU capable of executing functions in the Java language.

4.2 Modeling a Heterogeneous HW Component Profile using HCML

Figure 6 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.

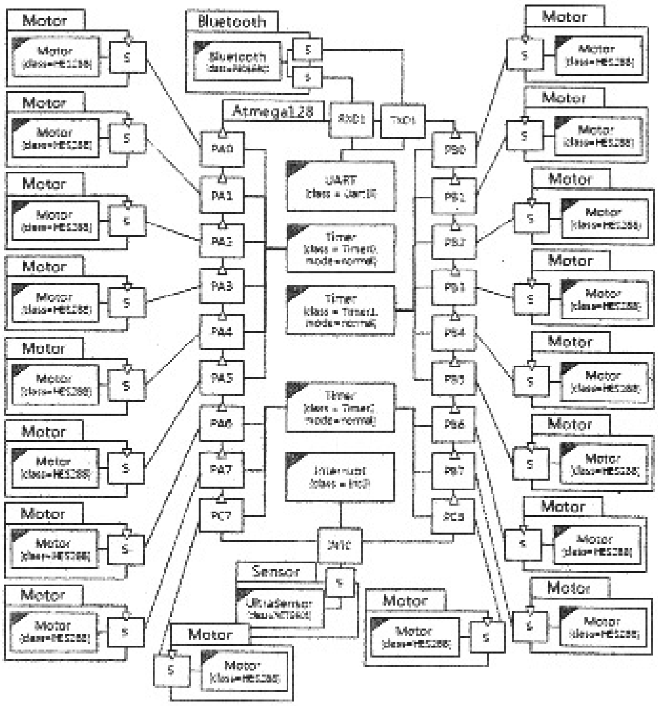


Fig. 6. Hexapod robot modeling using HCML

Figure 7 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.

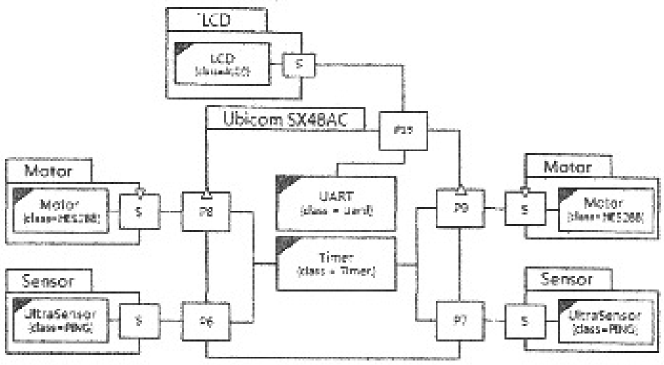


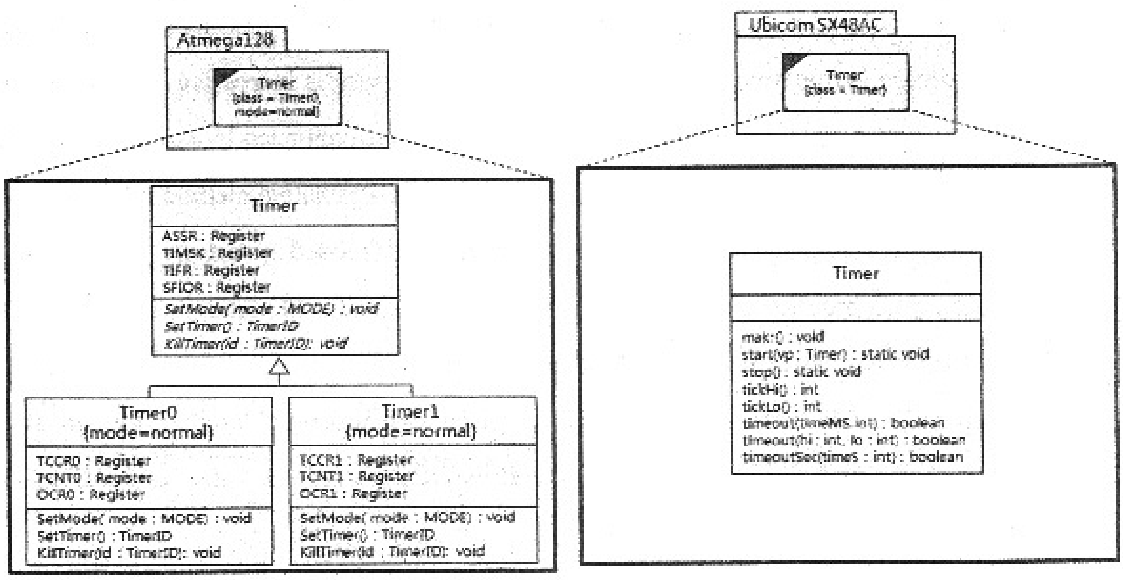
Fig. 7. Modeling Javelin using HCML

As presented in Table 2, Hexapod and Javeline robot modeling information are saved as XML. Table 2 only describes the port-related content among all codes. The Port is contains direction, name, and ID details. These 3 types of information are saved in each connected Port. A total of 21 ports are used in the Hexapod robot while 5 ports are used in the Javeline.

Table 2. Port Comparison in Hardware Profiling

(a) Hexapod robot	(b) Javeline
<Port direction="out" name="PA0" id="PORT1" />	<Port direction="in out" name="P6" id="PORT1" />
<Port direction="out" name="PA1" id="PORT2" />	<Port direction="in out" name="P7" id="PORT2" />
<Port direction="out" name="PA2" id="PORT3" />	<Port direction="out" name="P8" id="PORT3" />
<Port direction="out" name="PA3" id="PORT4" />	<Port direction="out" name="P9" id="PORT4" />
<Port direction="out" name="PA4" id="PORT5" />	<Port direction="in out" name="P15" id="PORT5" />
<Port direction="out" name="PA5" id="PORT6" />	
<Port direction="out" name="PA6" id="PORT7" />	
<Port direction="out" name="PA7" id="PORT8" />	
<Port direction="out" name="PB0" id="PORT9" />	
<Port direction="out" name="PB1" id="PORT10" />	
<Port direction="out" name="PB2" id="PORT11" />	
<Port direction="out" name="PB3" id="PORT12" />	
<Port direction="out" name="PB4" id="PORT13" />	
<Port direction="out" name="PB5" id="PORT14" />	
<Port direction="out" name="PB6" id="PORT15" />	
<Port direction="out" name="PB7" id="PORT16" />	
<Port direction="out" name="PC3" id="PORT17"/>	
<Port direction="out" name="PC7" id="PORT18" />	
<Port direction="in out" name="INT0" id="PORT19" />	
<Port direction="in out" name="RXD1" id="PORT20" />	
<Port direction="in out" name="TXD1" id="PORT21" />	

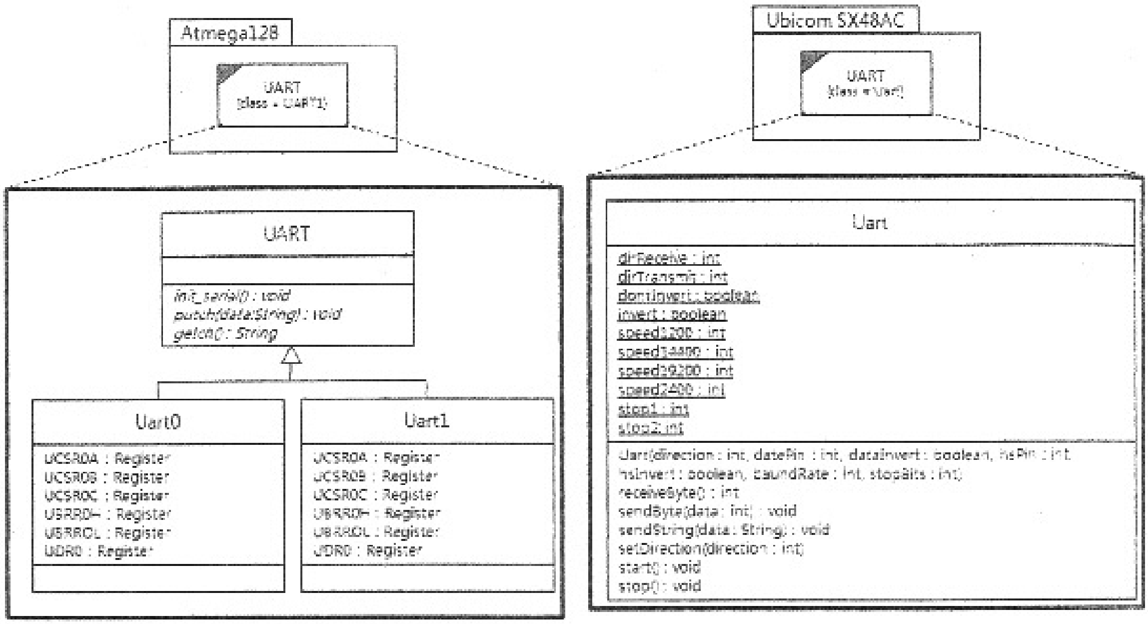
HCR is a main function of HCP in HCML. This is composed of hardware provided functions. By using HCML, heterogeneous SUGVs(e.g., the Hexapod robot, Javeline) can be expressed in the HCR. Figure 8 presents a modeling of the Timer functions among heterogeneous MCUs. Atmega128 is defined as 2 Timers, while Ubicom SX48AC is defined as 1 Timer. Whereas Atmega128 can directly access the hardware register, Ubicom SX48AC cannot access register since it passes through a java interpreter. Thus, Control can only be achieved through the use of API.



(a) Atmega128 (b) Ubicom SX48AC

Fig. 8. HCR Timer

Among heterogeneous MCU functions, Figure 9 specifically expresses UART as HCR. 2 UARTs exists in Atmega128, while 1 UART exists in Ubicom SX48AC. As in the Timer, hardware register access can be achieved in Atmega128. However, it is impossible to achieve access in Ubicom SX48AC. Thus, hardware control can only be achieved through the use of API in Ubicom SX48AC.



(a) Atmega128 (b) Ubicom SX48AC

Fig. 9. UART of HCR

5. Conclusion

Although the existing hardware design method was expressed using abstract hardware, most are used to design hardware chips. The importance of software is increasing in comparison with hardware as the application field for embedded systems continues to expand, which in turn requires more functions to be made available. Thus, an alternative method is required to help many software developers to understand and develop embedded systems.

To solve this, we propose using 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. As a result, it is possible to specify the use of the API for software control, and simply construct hardware information with our proposed method. Automation devices are currently being developed and represented to achieve faster and easier production of more diverse hardware component information.

6. Acknowledgments

This research was supported by 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-(C1090-1131-0008)) and the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation.

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(2009), 775-779.

[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(2008), 409-414.

[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, 2008, 93-98

[5] Axel Jantsch, Modeling Embedded System and SOCs. *Mogan Kaufmann*, 2004.

[6] IEEE Std 1451.2 1997, Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. *Institute of Electrical and Electronics Engineers*, 1997.

[7] A. Ferrari and A. Sangiovanni-Vincentelli, System design: Traditional concepts and newparadigms. *presented at the 1999 Int. Conf. Computer Design*, Austin, TX, 1999.

[8] D. Perry 1991, VHDL. *McGraw-Hill*, 1991.

[9] D.C. Black and J. Donovan, SystemC: from the ground up. *Kluwer Academic Publishers*, 2004.

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

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

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

[13] D.E. Thomas and P.R. Moorby, The Verilog hardware description language. Kluwer *Academic Publishers*, 1991.

[14] Morison,J.D., Peeling N. E. and Thorp T. L., ELLA: A Hardware Description Language. *Proc. IEEE International Conference on Circuits and Computers*, Sep.-Oct. 1982, 604-607.

[15] F. Fallah, S. Devadas and K. Keutzer 1998, Functional vector generation for HDL models using linear programming and 3-satisfiability. *Proceedings Design Automation Conference*, 1998, 528-533.

[16] R. Waxman, J.H. Aylor and E. Marschner, The VHSIC hardware description language (IEEE standard 1076): language features revisited. *Compcon Spring '88.*, San Francisco, CA, USA, 1988, 310-315.

[17] 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(2007), 83-88.

*Corresponding author: Robert Young Chul Kim, Prof.

Department of Computer and Information Communication,

Hongik University,

300 Jochiwon-eup, Yeonki-gun, Choongchungnam-do 339-701, Korea

E-mail: bob@hongik.ac.kr