



소프트웨어공학
사 이 어 티

17권 제1호
Vol. 7 No. 1



KIISE · KIPS

Joint Workshop on Software Engineering Technology 2009
(KSEJW-2009)

2009

한국 소프트웨어공학기술 합동 워크샵 논문집

- 일시 : 2009년 7월 23일(목) ~ 25일(토)
- 장소 : 제주 오리엔탈 호텔

주최 : 한국정보과학회

한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학 소사이어티

한국정보처리학회 소프트웨어공학 연구회

후원 : 삼성탈레스(주), 한국과학기술정보연구원, 한국오라클
(주)다한테크, (주)모아소프트, (주)솔루션링크

목 차

초청 강연

- 국방기술기획과 국방 SW 산업의 발전 3
정재원 원장(국방기술품질원)

아키텍처 및 컴포넌트

- EHR 관점에서 수립한 CDSS 아키텍처와 적용 사례 21
김정아(관동대), 조인숙(인하대), 김윤(서울대)
- MOF 기반의 하드웨어 컴포넌트 모델링 25
손현승, 김우열, 서채연, 김동호, 안성빈, 김영철(홍익대),
김재수(한국과학기술정보연구원)
- 자가 치유 시스템의 품질 속성을 위한 Aspect 기반의 Architecture 28
윤현지, 박수용(서강대)

과제현황발표

- 과학기술 혁신정책 통합지원 지식베이스 구축에 관한 연구 43
전승수, 정경진(한국과학기술 기획평가원), 이병수(시립인천대)
- 소규모 URC 네트워크 로봇 카페 시스템 개발 57
조자연, 황선명, 김경석, 최재운, 조영민(대전대)
- 고품질의 SW 개발을 위한 SW 공학들의 적용 프로젝트 소개 68
한혁수, 도성룡, 김초이(상명대)

알고리즘 및 테스트

- LIS 를 위한 미들웨어 구축과 큐잉, 마이닝 알고리즘 81
김대경, 이귀봉, 김준환, 기재석, 이병수(인천대), 이상순(가천의과대),
- Fault Localization 기법을 활용한 테스트케이스 우선순위 기법 89
김세준, 권태용, 백종문(KAIST)
- 서버 가상화를 이용한 소프트웨어 성능 테스트 기법 96

MOF 기반의 하드웨어 컴포넌트 모델링

손현승*, 김우열*, 서채연*, 김동호*, 안성빈*, 김영철*, 김재수**

홍익대학교 일반대학원 소프트웨어공학 연구실*
 한국과학기술정보연구원**
 {son, john, jyun, ray, ahn, bob}@selab.hongik.ac.kr*
 jaesoo@kisti.re.kr**

요약: 임베디드 소프트웨어는 하드웨어에 종속적이기 때문에 하드웨어에 대한 요구사항이 변경되면 소프트웨어 역시 변경되어야 한다. 그래서 하드웨어 정보를 소프트웨어 개발자가 이해하기 쉽도록 구조화한다면 빠른 개발이 가능해질 것이다. 본 논문에서는 복잡한 하드웨어 정보를 표현하고 모델링하기 위해 MOF 기반의 하드웨어 컴포넌트 모델링 언어를 제안하였다. 제안한 하드웨어 컴포넌트 모델링 언어는 기존의 UML 다이어그램과 변환 및 확장이 용이하도록 MOF 기반으로 구성되고 하드웨어 구성정보를 모델화하여 복잡성을 줄여준다. 적용사례로서 제안한 모델링 언어를 사용해 임베디드 시스템의 하드웨어 정보를 구성하였다.

핵심어: MOF, UML, HDL, MDD

1. 서론

임베디드 시스템[1][2]에서 하드웨어 제어를 위한 기술 문서는 데이터 시트[3]라는 형태로 제공된다. 그러나 데이터 시트는 소프트웨어 개발자가 이해하기 어려운 형태로 제공된다. 또한 수많은 종류의 레지스터와 제어방법을 숙지하기란 쉽지 않은 작업이다. 또한 기존의 하드웨어를 설계[4]하기 위한 언어인 VHDL[5]과 SystemC[6] 등이 존재하지만 소프트웨어 개발을 위한 것이 아니라 칩 제조를 위한 것이어서 사용하기 불편하다.

UML(Unified Modeling Language)[7][8][9]은 통합 모델링 언어로 표준화된 표현을 통해 소프트웨어 개발 라이프 사이클 과정에서 생길 수 있는 개발자간의 의사소통의 불일치를 해소할 수 있다. UML은 MOF(Meta Object Facility)[9][10]로 정의된다. MOF는 구현이 예상되는 어떠한 종류의 모델과 모델링 패러다임 지원이 가능하다.

본 논문에서는 복잡한 하드웨어 정보를 표현하

고 모델링하기 위해 MOF 기반의 하드웨어 컴포넌트 모델링 언어를 제안한다. 이는 하드웨어 컴포넌트 프로파일의 기초가 되며 MOF를 기반으로 구성했기 때문에 기존 UML 메타모델과 쉽게 연동이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 MOF에 대해 기술한다. 3장에서는 제안한 하드웨어 컴포넌트 모델링 언어에 대하여 설명한다. 4장에서는 적용사례로서 하드웨어 컴포넌트 모델링 언어를 사용하여 임베디드 시스템의 하드웨어 컴포넌트를 모델링한다. 마지막으로 5장에서는 결론 및 향후 연구를 언급한다.

2. 관련연구

MOF는 OMG에서 제안한 메타 객체 관리 기법이다. MOF는 UML을 통해서 표현되고 UML의 메타모델은 MOF를 통해서 정의된다. MOF는 4계층 메타데이터 구조를 가진다.

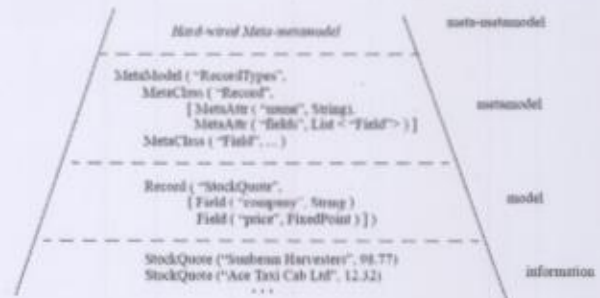


그림 1. MOF의 4계층 레이어

이 계층들은 정보계층, 모델 계층, 메타모델 계층, 메타-메타모델 계층으로 구분된다. 정보계층은 우리가 서술하기 원하는 순수 데이터들로 구성된다. 모델 계층은 정보계층에 있는 데이터를 설명

* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터육성지원사업(C1090-0903-0004)과 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.

하는 메타데이터로 구성된다. 메타모델 계층은 메타데이터의 구조와 의미를 정의하는 서술(즉 메타-메타데이터)로서 구성된다. 메타모델은 다른 종류의 데이터를 서술하기 위한 “추상 언어”이다. 즉, 이는 구체적인 문법이나 표기가 없는 언어이다. 메타-메타모델 계층은 메타-메타데이터의 구조와 의미의 서술로서 이루어진다. 이는 다른 종류의 메타데이터를 정의하기 위한 “추상 언어”이다[11].

그림 1 은 전형적인 4 계층 메타 모델링 프레임워크다. 그림에서 StockQuote 타입이 정보 단계(level)에서 많은 StockQuote 인스턴스를 서술하는 것과 똑같이 RecordTypes 메타모델은 모델 단계에서 많은 레코드 타입을 서술할 수 있다. 마찬가지로 메타-메타모델 단계는 메타 모델에서 많은 다른 메타모델을 서술할 수 있다.

전형적인 4 계층 메타데이터 구조는 단순 모델링 방법에 비해 많은 장점을 가지고 있다. 만일 프레임워크가 잘 정의되어 설계되었다면 구현이 예상되는 어떠한 종류의 모델과 모델링 패러다임을 지원, 연관된 다른 종류의 메타데이터의 허용, 증가적으로 부가되는 메타모델과 새로운 종류의 메타데이터의 허용, 임의의 메타데이터(모델)와 메타-메타데이터(메타모델)같은 메타-메타모델을 사용하는 단체간의 교환 지원이 가능하다. 결국 MOF 를 통해서 메타모델들을 표준화가 가능하기 때문에 다양한 모델간의 변환이 가능해질 수 있다.

3. MOF 기반 하드웨어 컴포넌트 모델링

하드웨어 컴포넌트 모델링 언어의 메타모델은 MOF 로 설계된다. 때문에 기존의 UML 다이어그램과 변환 및 확장이 손쉽게 이루어진다.

그림 2 는 하드웨어 컴포넌트 모델링 언어로 설계된 하드웨어 컴포넌트 프로파일이 기존의 임베디드 개발방법[12][13]의 모델변환 과정에서 어떻게 적용되는지를 도식화 한 것이다.

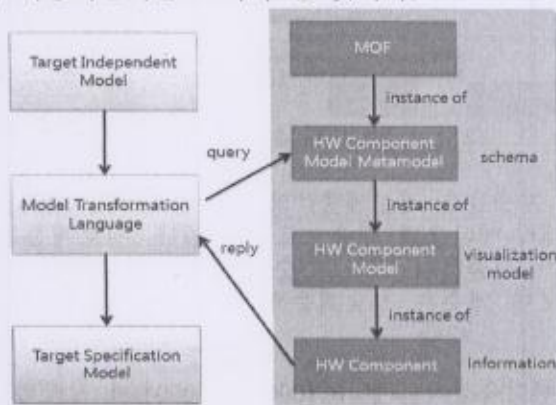


그림 2. 모델 변환에서 하드웨어 프로파일 적용

하드웨어에 독립적으로 구성된 타겟 독립 모델(TIM)을 하드웨어에 종속적인 모델(TSM)로 변환하기 위해서는 모델 변환 언어가 필요하다. 모델 변환언어는 하나의 모델이 다른 모델로 변환이 쉽게 하도록 도와줄 뿐 실제 하드웨어 정보를 제공해 주지는 않는다. 그래서 하드웨어 정보를 제공하는 프로파일 개념이 필요하다. 하드웨어 컴포넌트 프로파일은 MOF 기반의 하드웨어 컴포넌트 모델링 언어를 사용하여 하드웨어 정보를 구조화한다. 그리고 미리 정의한 하드웨어 컴포넌트 모델링 언어의 메타모델을 통해 하드웨어 정보가 프로파일에 저장된다.

4. 적용사례

적용사례로 사용된 Javeline 은 그림 3 과 같다. javeline 은 평탄한 지형에만 움직일 수 있지만 바퀴를 사용하여 빠르고 쉽게 이동할 수 있는 특징이 있다.

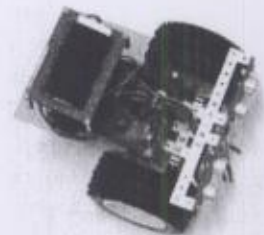


그림 3. Javeline 적용사례

Javeline 은 표 1 과 같이 Ubicom SX48AC, 모터 2 개, 센서 2 개로 구성된다. Javeline 은 바퀴형 로봇으로 적은수의 모터를 사용한다. 또한 MCU 내에 자바가상머신이 하드웨어로 내장되어 자바언어를 수행할 수 있도록 되어있다.

표 1. 하드웨어 구성

구분	Javeline
Microcontroller	Ubicom SX48AC 20MHz
RAM	32 KByte
EEPROM	32 KByte
Sensor	2
Communication	None
Motors	2
Languages	Java

Javelin 에서 사용하는 Ubicom SX48AC 는 총 16 개의 핀으로 구성되고 이중에서 P6~9, P15 를 사용한다. P6, 7 은 초음파센서, P8, 9 는 서보모터, P15 는

LCD 를 제어하는 용도로 사용한다. Javelin 은 제어 위해서 MCU 의 기능 중 UART, Timer 를 사용해서 하드웨어를 제어한다.

이러한 하드웨어 구조를 하드웨어 컴포넌트 모델링 언어로 표현하면 그림 4 와 같다.

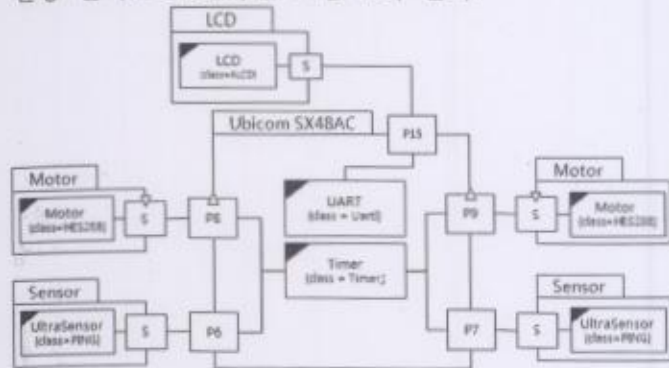


그림 4. 하드웨어 컴포넌트 모델링 언어를 이용한 Javelin 모델링

Javelin 의 모델링 정보가 저장될 때는 표 2 와 같이 XML 로 표현된다. 표 2 에는 전체 코드 중 Port 에 관련된 내용만 기술하였다. Port 는 방향, 이름, 아이디로 구성된다. 각 연결된 Port 에 3 가지 정보를 저장하도록 한다. Javelin 은 5 개의 포트가 사용되었다.

표 2. Javelin 의 하드웨어 프로파일(Port)

```
<Port direction="in out" name="P6" id="PORT1" />
<Port direction="in out" name="P7" id="PORT2" />
<Port direction="out" name="P8" id="PORT3" />
<Port direction="out" name="P9" id="PORT4" />
<Port direction="in out" name="P15" id="PORT5" />
```

5. 결론

기존의 하드웨어 설계 방법은 하드웨어를 추상화 하여 표현하지만 대부분이 하드웨어 칩 설계를 위한 용도로 사용된다. 임베디드 시스템의 활용분야가 넓어지고 많은 기능을 요구함에 따라 하드웨어 보다는 소프트웨어의 중요성이 커져가고 있다.

그래서 본 논문에서는 소프트웨어 설계의 표준인 UML 의 기능을 충실히 살려 MOF 기반의 하드웨어 컴포넌트 모델링 언어를 제안하고 이를 이용하여 임베디드 시스템의 하드웨어 정보를 구성하였다. 그 결과 하드웨어에 대한 정보를 간편하게 구축할 수 있는 장점이 있었다.

향후 연구로 다양한 하드웨어 컴포넌트 정보를 쉽고 빠르게 구성할 수 있도록 자동화 도구를 개

발 중이다.

참고문헌

- [1] David E. Simon, *An embedded software primer*, Addison-Wesley, 1999.
- [2] Axel Jantsch, *Modeling Embedded System and SOCs'*, Mogan Kaufmann, 2004.
- [3] IEEE Std 1451.2, "Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats", IEEE, 1997. 09. 26.
- [4] A. Ferrari, A. Sangiovanni-Vincentelli, "System design: Traditional concepts and new paradigms", *Computer Design*, 1999. 10.
- [5] D. Perry, *VHDL*, McGraw-Hill, New York, 1991.
- [6] D.C. Black, J. Donovan, *SystemC: from the ground up*, Kluwer Academic Publishers, 2004.
- [7] OMG, "Unified Modeling Language Specification Version 1.4.2", ISO/IEC 19501, 2005. 04. 01.
- [8] C. Snook, M. Butler. "UML-B: Formal Modeling and Design Aided by UML", *ACM Trans. Softw. Eng. Methodol.*, 15(1), 92-122, 2006.
- [9] Francisco A. M. do N. et al. "MDA-based approach for embedded software generation from a UML/MOF repository", *Proceedings of the 19th annual symposium on Integrated circuits and systems design*, 143-148, 2006. 08. 29.
- [10] OMG, "Meta Object Facility (MOF) Specification Version 1.4.1", ISO/IEC 19502, 2005. 05.05.
- [11] Atkinson, C., Kuhne, T., "Model-driven development: a metamodeling foundation, *Software*", IEEE, Volume: 20, Issue: 5, 36-41, 2003.
- [12] 김우열, 손현승, 김영철, "UML 프로파일 메커니즘을 이용한 이종 소형 무인지상차량 설계 자동화", *정보과학회*, 제 35 권, 12 호, 705-715, 2008. 12.
- [13] 손현승, 김우열, 김영철, "이종 임베디드 시스템의 멀티태스킹을 위한 MDA(Model Driven Architecture) 기반의 설계", *정보처리학회*, 제 15 권, 3 호, 335-360, 2008. 06.