

논문 2008-4-15

## 스마트 기기의 컨버전스를 위한 모델링 도구에 관한 연구

## A Study on Modeling Tool for Convergence of Smart Appliances

손현승\*, 김우열\*\*, 김영철\*\*\*

Hyun Seung Son\*, Woo Yeol Kim\*\*, R. Young Chul Kim\*\*\*

**요약** 스마트 기기의 컨버전스를 위한 기존의 방법은 미들웨어를 사용하여 소프트웨어를 통합하는 것이다. 미들웨어를 사용하면 쉽고 빠르게 통합이 가능한 장점이 있다. 하지만 미들웨어는 기본적으로 소프트웨어가 크기 때문에 수행 속도가 느리다. 이러한 단점을 극복하기 위해 본 논문에서는 컨버전스를 위한 모델링 도구를 제안한다. 제안하는 모델링 도구는 모델 단계에서 컨버전스를 수행하고 자동코드를 생성하여 미들웨어 없이 소프트웨어의 컨버전스가 가능하게 한다. 적용사례로서 u-Home 환경의 텔레비전과 도어락 시스템의 컨버전스를 보여주었다. 미들웨어를 사용하지 않아 저렴한 비용으로 스마트 기기의 컨버전스를 수행하였고 사용자가 원하는 컨버전스 제품을 빠르게 개발할 수 있었다.

**Abstract** The traditional way for convergence of smart appliances was software integrations on middleware. It is possible advantage to integrate easily and quickly with this way. But basically the middleware approach is slowly executed as it is huge the size of software. To solve this problem, we suggest a modeling tool for convergence. As we can execute and generate the automatic source code at the design phase with our modeling tool, it is possible to do software integration without the middleware. We use one example of U-Home environment to show the convergence for TV and the door lock system. With our approach, we will be able quickly to develop the convergent products what the customers desire with inexpensiveness, that is, the convergence of smart appliances without the middleware.

**Keywords** : 컨버전스(Convergence), MDA(Model Driven Architecture), xUML(Executable UML), 모델링 도구(Modeling Tool), 스마트 기기(Smart Appliances)

## 1. 서론

소비자 니즈의 다양화 및 고도화에 따라 최근 디지털 컨버전스[1]는 기존의 컴퓨터, 통신, AV등 전자기기 간의 융합을 넘어 산업전반의 영역으로 확대되고 있다. 마이크로 소프트는 디지털 컨버전스를 위해 가전기기 업체와의 협력을 통하여 소프트웨어 플랫폼을 제공하는 전략을 수행하고 있다[2]. IBM은 ON DEMAND BUSINESS [3]라는 통합 솔루션을 바탕으로 진행하고 KT는 홈 네트

워크, 차세대 이동통신, 디지털 콘텐츠, IT서비스, 미디어를 융합하는 전략을 추구하고 있다[4].

홈네트워크 분야에서의 컨버전스를 이루기 위한 미들웨어 기술로는 UPnP(Universal Plug and Play), Jini, HAV(Home Audio Video Interoperability) 등이 있다[5]. 이것은 소프트웨어를 조립으로 구성하는 SaaS(Software as a Service)[15]를 실행하기 위해 적합하다. 하지만 미들웨어를 사용하게 되면 기본적인 하드웨어 요구사항이 크고 처리속도가 느린 단점이 있다. 미들웨어는 또한 필요 이상의 소프트웨어가 탑재되기 때문에 수행 속도에 많은 제약을 가지게 된다. 호환성과 편리성 이라는 장점 있지만 제품 경쟁력을 위해서는 앞에 제시한 문제점을 가

\*학생회원, 홍익대학교 일반대학원 석사과정

\*\*학생회원, 홍익대학교 일반대학원 박사과정

\*\*\*정회원, 홍익대학교 컴퓨터정보통신 부교수

접수일자: 2008.7.15, 최종수정일자:2008.8.2

지고 있다.

본 논문에서는 미들웨어의 문제점을 극복하고 스마트 기기의 컨버전스를 수행하기 위해 MDA(Model Driven Architecture) 기반의 컨버전스 모델링 도구를 제안한다. MDA[6]는 OMG에서 제안한 방법으로 PIM(Platform Independent Model)과 PSM(Platform Specific Model)을 사용하여 독립적인 모델과 종속적인 모델로 분리시켜 재사용성과 상호운용성을 높이는 방법이다. 하지만 임베디드 시스템에 적합하지 않아 확장하여 도구에 적용하였다 [7,10,12]. 제안한 도구는 타겟 독립 모델단계에서 모델간 컨버전스를 수행한다. 그리고 생성된 모델의 자동변환을 통하여 개발을 쉽고 빠르게 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 임베디드용 MDA개발 방법에 대하여 언급한다. 3장에서는 컨버전스를 위한 모델링 도구에 대해 설명한다. 4장에서는 텔레비전과 도어락을 컨버전스하는 사례를 통해 도구사용 방법에 대해 언급한다. 5장에서는 결론 및 향후 연구에 대해 기술한다.

## II. 관련연구

MDA는 플랫폼 기반의 개발 방법론으로 CORBA, .NET, EJB와 같은 엔터프라이저급에서 사용되는 방법이다. 이를 임베디드 시스템에 적용하기위해 MDA에 기본적인 사상인 모델변화과 자동코드생성 기법을 가져왔다. 제안한 MDA기반의 임베디드 소프트웨어 개발 프로세스 [11]는 요구사항분석, 타겟 독립 모델(TIM), 타겟 종속 모델(TSM), 타겟 의존 코드(TDC) 4가지 단계로 구성된다. 그리고 기존의 UML의 모든 다이어그램을 사용하는 것이 아니라 꼭 필요한 4가지(유스 케이스, 클래스, 메시지, 상태 다이어그램)만 사용한다. 첫 번째 단계는 요구사항 및 분석단계로 요구사항에 대한 정보를 문서화 하고 유스 케이스 다이어그램을 사용하여 요구사항을 설계 및 분석 한다. 두 번째 단계에서는 확장된 xUML[8,9,10]을 사용하여 타겟 독립 모델을 설계한다. 이때 사용하는 다이어그램은 클래스, 병렬 메시지, 병렬 상태 다이어그램을 사용한다. 타겟 독립 모델은 하드웨어 플랫폼이나 운영체제에 의존적이지 않는 모델 이다. 세 번째 단계는 타겟 종속 모델(TSM)으로 타겟 독립 모델로부터 자동 변환된 모델을 사용하여 설계 한다. 타겟 종속 모델로 변환

을 위해 UML 프로파일을 사용하게 된다. 프로파일에는 미들웨어, 운영체제, 프로세서의 정보가 들어 있고 모델 변환단계에서 사용된다. 타겟 종속 모델 단계는 도구를 통해 자동변환 되지만 도구로 불가능한 부분을 추가로 설계하는 단계이다. 마지막 단계는 타겟 의존 코드(TDC)로 코드 템플릿에 의해 C, C++, Java 등으로 최종 코드를 생성한다. 대부분 코드는 자동 생성되지만 생성할 수 없는 부분의 코드는 수작업을 통하여 작성한다. 생성된 소스코드를 타겟 모델에 탑재하는 작업이 바로 개발한 소프트웨어를 임베디드 소프트웨어 시스템에 적용시키는 것이다[7,11,18]. 모든 과정에서 생성된 소스코드와 모델은 재사용 저장소에 저장 후에, 기능을 변경하거나 추가하여 새로운 임베디드 시스템을 구축하고자 할때 저장된 모델을 재사용한다. 자동 코드 생성 기법은 모델을 텍스트로 변환하는 방법[13]을 이용한다.

## III. 컨버전스를 위한 모델링 도구

### 1. 모델 자동 컨버전스

컨버전스를 위해서는 2개 이상의 타겟 독립 모델이 필요하다. 레거시 시스템을 이용하여 컨버전스를 할 수도 있고 기존의 설계모델에 새로운 모델을 추가 할 수도 있다. 레거시 시스템을 이용할 경우는 모델로 변환 해주는 리버스 엔지니어링 단계가 필요하다. 모델 병합을 위해서는 타겟 독립 모델이 있어야 한다. 각 타겟 독립모델은 컨버전스 되기 전에 유사도 검사를 통해서 각 클래스들이 얼마나 유사한지를 검사하게 된다. 유사할 경우 병합되고 새로운 클래스나 메소드는 추가되어 진다.

그림 1은 컨버전스를 수행하기 위해 2개의 클래스 다이어그램을 선택한 모습이다. 클래스 다이어그램은 이미 생성되어 있어야 하고 "Load"버튼을 이용하여 위치한 디렉토리를 선택하면 가능한 클래스 다이어그램 목록이 나타난다. 이때 컨버전스할 클래스를 선택한 후 다음을 클릭하면 클래스 다이어그램의 병합을 수행한다.

그림 2는 클래스 다이어그램을 병합한 결과이다. 결과를 확인해보면 클래스 이름과 메소드들로 구분되고 마지막에는 추가되었는지 재사용되었는지에 대한 결과를 보여준다. 이 상태에 완료 버튼을 누르면 병합된 클래스 다이어그램이 생성되고 생성된 다이어그램은 도구에 자동으로 불러지게 된다.

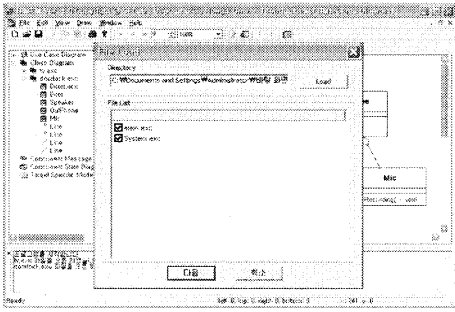


그림 1. 컨버전스를 위한 클래스 다이어그램 선택  
 Fig. 1. Selecting Class diagram for convergence

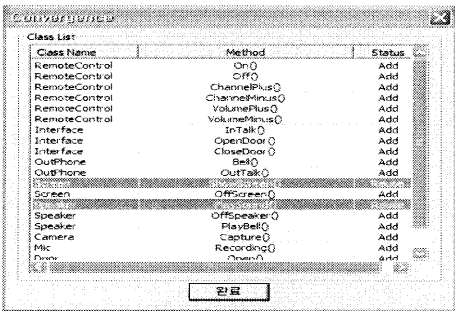


그림 2. 병합된 클래스 다이어그램 목록  
 Fig. 2. Combined Class diagram list

2. 타겟 독립 모델 리팩토링

타겟 독립 모델 상태에서 각 클래스의 연관관계와 메소드의 사용율을 이용하여 리팩토링을 수행하게 된다. 수행할 리팩토링은 Move Method, Extract Method로 너무 많은 연관 관계와 한 메소드에 많은 기능이 들어있는 것을 방지하기 위해 2가지 방법을 사용하게 된다. 리팩토링을 수행하기 위해 방법은 하나의 메소드를 수행하기 위해 연관된 클래스의 수가 3개 이상일 경우 Move Method를 수행하고 하나의 메소드를 5개 이상의 클래스에서 하나의 연관관계를 가진 상태에서 호출하게 되면 Extract Method를 수행하게 된다.

3. 병합된 모델 타겟 종속 모델로 변환

리팩토링 과정까지 마치면 UML 프로파일을 적용시켜 타겟 시스템에 종속적인 모델로 변환시켜 시스템 개발을 수행하게 된다.

IV. 적용사례

1. 요구사항

적용사례는 최근 컨버전스가 빈번하게 발생하는 홈네트워크 환경에서 텔레비전과 도어락을 대상으로 수행한다. 텔레비전 및 도어락을 통해 수행하는 사용자의 작업(Task)을 수집하여 데이터를 시간에 따라 순차적으로 표현하여 모델링 하였다. 사용자 시나리오는 국민대학교 테크노 디자인 대학원 인터랙션 디자인 연구실의 과제인 “사용자 형태 분석을 통한 홈 네트워크 사용자 인터페이스 연구”에 데이터를 사용하였다[14]. 이것을 사용자의 행위 분석을 방법을 사용하여 모델링 할 수 있다[16, 17]. 사용자들의 행위들을 분석해보았더니 텔레비전을 볼 때 도어락의 기능을 사용하고 싶은 욕구를 사용자 관측 데이터를 통해서 찾아 낼 수 있었다. 이러한 사용자 니즈(Needs)를 통한 서비스 영역을 도출 하고 이것을 개발해야 된다는 새로운 비즈니스의 요구를 추출 할 수 있다.

2. 타겟 독립 모델

시스템을 개발하기 전에 텔레비전과 도어락 시스템이 컨버전스 될 것을 알았다면 하나의 모델로 설계를 하였을 것이다. 하지만 보통의 그러한 경우는 드물다. 이미 개발된 시스템을 새롭게 설계한다는 것은 개발시간을 증가 시키기 때문이다. 또한 하나의 시스템은 알고 모르는 시스템이 있을 경우에는 개발하기 위한 많은 애로사항을 가지게 될 것이다. 이러한 문제를 해결하기 위해서 제한한 도구를 사용하게 된다. 본 논문의 적용사례는 기존의 두 개의 모델이 서로 다르게 설계 되었다는 가정으로 진행한다.

그림 3, 4는 2개의 서로 다른 시스템의 클래스 다이어그램이다. 먼저 텔레비전 시스템을 만들고 도어락 시스템을 추가하는 상황으로 생각한다면 클래스 다이어그램이 2개 생길 것이다. 텔레비전의 구성은 “RemoteControl”는 리모컨 처리를 위한 클래스이고 “TV”는 텔레비전의 컨트롤 클래스이다. “Screen”은 화면 표시장치이고 “Speaker”는 스피커 이다.

도어락 시스템은 그림 4처럼 구성된다. “DoorLock”은 컨트롤 클래스이고 “Door”는 문의 잠금장치 “OutPhone”은 인터폰 이다. “OutPhone”은 “Speaker”와 “Mic”로 구성되고 각각 스피커와 마이크를 나타낸다. 서로 상이한 기능을 가지고 있는 2개의 클래스를 컨버전스를 위해 그림 5와 같이 컨버전스 파일을 로드하고 선택하여 클래스 다이어그램의 병합하게 된다.

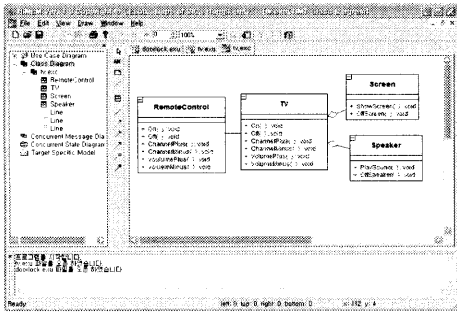


그림 3. 텔레비전의 클래스 다이어그램  
Fig. 3. lass diagram of TV

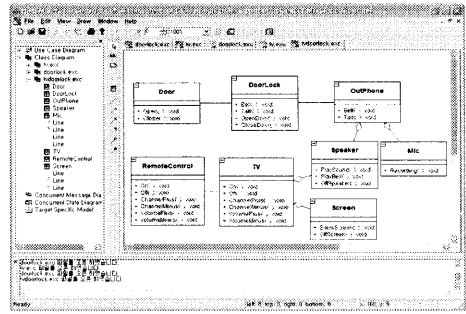


그림 6. 컨버전스된 클래스 다이어그램  
Fig. 6. Converged Class diagram

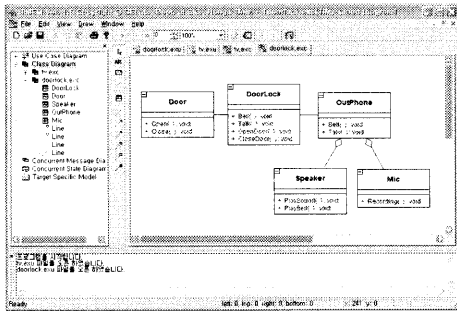


그림 4. 도어락의 클래스 다이어그램  
Fig. 4. Class diagram of door lock

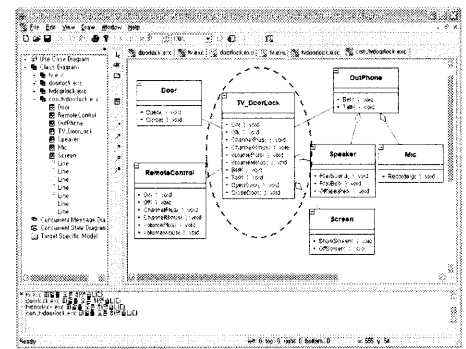


그림 7. 클래스 다이어그램 리팩토링  
Fig. 7. Refactoring for class diagram

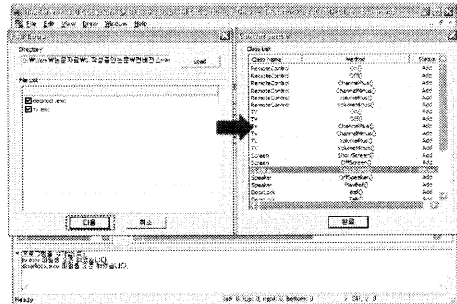


그림 5. 도구를 이용한 컨버전스 단계  
Fig. 5. Convergence phase using tool

### 3. 타겟 종속 모델

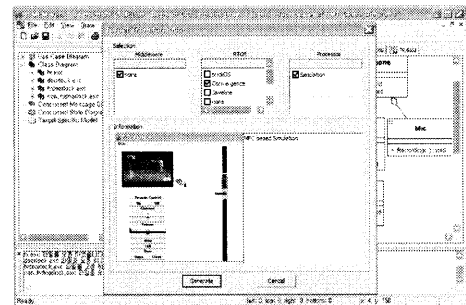


그림 8. 시뮬레이션을 가능하도록 모델 변환  
Fig. 8. Model transformation for simulation

도구를 통해 병합하게 되면 그림 6과 같은 클래스 다이어그램을 얻을 수 있다. 2개의 클래스 다이어그램을 같은 부분은 합치고 다른 부분은 추가하여 다이어그램을 자동 생성한다.

병합시킨 클래스 다이어그램으로 시스템을 그대로 개발한다면 많은 문제가 생길 것이다. 그래서 리팩토링 과정을 거치게 된다. 그림 7의 점선으로 표시된 원과 같이 컨트롤 클래스를 합치고 연관관계를 리팩토링하여 클래스 다이어그램을 완성한다.

타겟 독립 모델인 완성되면 타겟 종속 모델로 변환하게 된다. 그림 8은 타겟 독립 모델에서 타겟 종속 모델로 변환하는 모습이다. 기존의 도구는 Middleware, RTOS, Processor의 3계층을 이용하여 변환 하지만 실제 임베디드 시스템에 적용하지 못하므로 시뮬레이션 할 수 있는 환경으로 변환 한다. Middleware에는 적용하지 않는 none을 RTOS는 Convergence, Processor에는 Simulation

을 매핑 시켜 시뮬레이션 수행을 위한 모델로 변환한다.

#### 4. 타겟 의존 코드

타겟 독립 모델로 생성되면 타겟 의존 코드로 생성시킬 수 있다. 그림 9는 타겟 의존 코드 생성을 위한 언어를 선택하는 과정이다. 시뮬레이션 수행은 마이크로 소프트웨어의 MFC 환경에서 수행되므로 C++언어로 생성하도록 하였다.

그림 10은 도구를 통해서 생성된 코드의 모습이다.

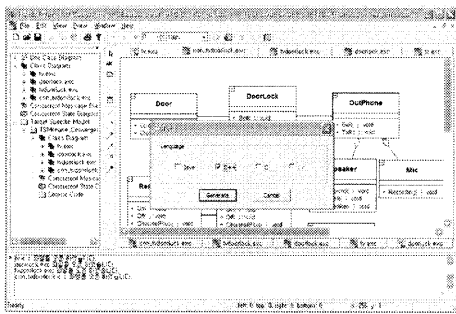


그림 9. 코드 자동 생성을 위한 언어 선택  
Fig. 9. Selecting Program language for automatic code generation

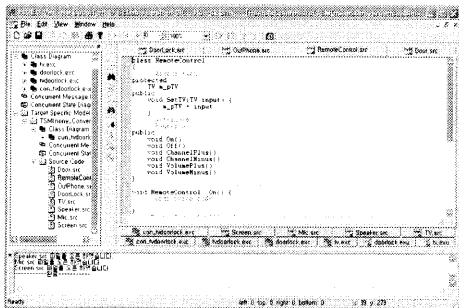


그림 10. 타겟 종속 모델에서 생성된 코드  
Fig. 10. Generated code from target specific model

#### 5. 시뮬레이션

위에서 모델링한 것들을 윈도우 환경에서 MFC(Microsoft Foundation Class Library)를 이용하여 시뮬레이션 하였다. 그림 11은 시뮬레이션 환경 모습이고 적용하여 모델을 가지고 만들어낸 프로그램을 보여주고 있다. 그림을 보면 사용자가 외부폰(OutPhone)에게 다가가 Bell을 작동시키면 도어락의 컨트롤 클래스에 의해서 Bell()메서드가 호출되고 텔레비전 컨트롤은 Bell()메서드를 수행하여 화면에 도어락으로 부터 받은 메시지를

텔레비전 화면 한 구석에 출력시켜주게 된다. 사용자는 이 화면을 보고 밖에 사용자의 문을 열어주게 된다. 여기서는 리모컨(RemoteControl)를 통해서 수신 받은OpenDoor() 메서드에 따라서 TV는 Door Service에게 Open()이라는 메서드를 호출함으로써 실제로 Door가 문을 열어준다.

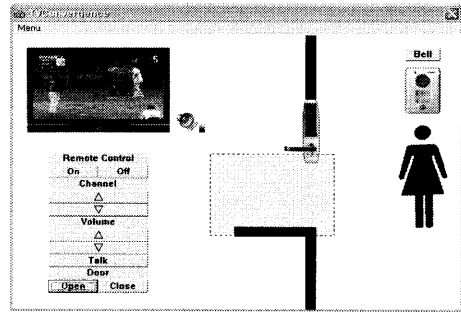


그림 11. TV와 도어락의 시뮬레이션  
Fig. 11. imulation of TV and door lock

### V. 결 론

본 논문에서는 기존의 미들웨어의 문제점을 극복하고 스마트 기기의 컨버전스를 수행하기 위해 확장된 MDA 기반의 모델링 도구를 제안하였다. 제안하는 모델링 도구는 모델 단계에서 컨버전스를 수행하고 자동코드를 생성하여 미들웨어 없이 소프트웨어의 컨버전스가 가능하게 하였다. 또한 병합된 모델을 정제하기 위해 도구가 지원하는 리팩토링 과정을 보여주었다. 이러한 과정을 통해서 2개의 서로 다른 클래스를 도구로 쉽게 병합하여 빠르게 컨버전스를 수행할 수 있었고 리팩토링 과정으로 보다 완성된 모델로 개발할 수 있었다. 미들웨어를 사용하지 않아 저렴한 비용의 프로세서를 사용하여 시스템을 개발할 수 있게 되었다.

향후에는 시뮬레이션을 위한 것이 아니라 실제로 수행할 수 있는 임베디드 시스템에 적용할 수 있도록 할 것이다. 그 경험을 바탕으로 제안한 개발 프로세스를 정제하고 완벽한 코드생성을 할 수 있는 개발도구와 환경을 만들 것이다. 그렇게 되면 빠르게 변화하고 있는 스마트 기기(Smart appliance)를 사용자가 원하는 컨버전스 제품을 적시에 개발할 수 있게 될 것이다.

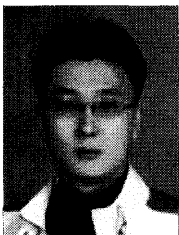
## 참 고 문 헌

- [1] 한국정보사회진흥원, “컨버전스 시대의 디지털 콘텐츠 시장”, IT Issue Report, 2006. 6
- [2] 한국정보사회진흥원, “MS 사례로 살펴본 디지털 컨버전스 방향 및 시사점”, IT Issue Report, 2006.
- [3] 강이훈, “Convergence, Growth Paths for Telecom Service Providers”, IBM, 2005
- [4] 맹수호, “디지털컨버전스시대의 KT 기업전략”, KT, 2005. 3.3
- [5] IITA, “홈네트워크 미들웨어 특허 동향“, 주간기술동향 통권 1326호, 2007. 12. 12
- [6] OMG, <http://www.omg.org/mda/>
- [7] 김우열, 김영철, “Adapting Model Driven Architecture for Modeling Heterogeneous Embedded S/W Components”, ICHIT2006, Vol. 2, 2006. 11.
- [8] Steve Mellor, Marc Balcer, ‘Executable UML: A Foundation for Model-Driven Architecture, Addison-Wesley’,(2002)
- [9] Leon Starr, ‘Executable UML: How to build class models’, Prentice Hall PTR,(2002)
- [10] 김우열, 김영철, “확장된 xUML을 사용한 MDA 기반 임베디드 SW 컴포넌트 모델링에 관한 연구”, KIPS Trans., Vol. 14-D, No. 1, 2007. 2.
- [11] 김동호, 김우열, 김영철, “MDA 기반 임베디드 SW 설계에 관한 연구”, IWIT, Vol.6, no.1, 06.03.
- [12] 손현승, 김우열, 서윤숙, 김동호, 김동우, 김재수, 김영철 “이종 임베디드 소프트웨어를 위한 코드 생성 메커니즘 및 지원도구”, 한국소프트웨어공학회, Vol. 9, No. 1, 170-177, 07.02.22
- [13] OMG, [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm](http://www.omg.org/technology/documents/modeling_spec_catalog.htm)
- [14] 정지홍, “사용자 행위 분석을 통한 홈 네트워크 사용자 인터페이스 연구”, KIDP 2005.8
- [15] KIPA, “SaaS 모델의 국내외 도입 현황“, SW 산업동향, 2007.8.27
- [16] 김우열, 손현승, 김영철, 정지홍, “A Study on the Convergence of Smart Appliances through User Behavior Analysis”, KIPS, Vol. 13, No. 2, 519-522, 06.11.10
- [17] 손현승, 김우열, 김영철, “A Study on User Behavior Analysis Tool for Convergence of Smart Appliances”, KIPS, Vol. 13, No. 2, 559-562, 06.11.10
- [18] 손현승, 김우열, 김영철, “MDA(Model Driven Architecture) based Design for Multitasking of Heterogeneous Embedded System”, KIPS, Vol. 15-D, No. 3, 355-360, 08.06

Acknowledgement: 이 논문(저서)은 2007년도 한국학술진흥재단의 지원을 받아 연구되었음 (KRF-2007-521-D00426).

## 저자 소개

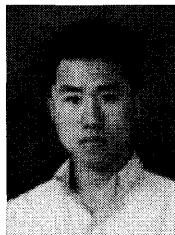
### 손 현 승(학생회원)



- e-mail : son@selab.hongik.ac.kr
- 2007년 홍익대학교 컴퓨터정보통신(학사)
- 2007년 ~ 현재 홍익대학교 일반대학원 소프트웨어공학전공(석사)

<주관심분야 : 임베디드 소프트웨어 자동화 도구 개발, 임베디드 RTOS 개발, 임베디드 MDA (Model Driven Architecture) 연구, 모델 검증 기법 연구>

### 김 우 열(학생회원)



- e-mail : john@selab.hongik.ac.kr
- 2004년 홍익대학교 컴퓨터정보통신(학사)
- 2006년 홍익대학교 일반대학원 소프트웨어공학전공(석사)
- 2006년 ~ 현재 홍익대학교 일반대학원 박사과정

<주관심분야 : 상호운용성, 임베디드 소프트웨어 개발 방법론 및 도구 개발, 컴포넌트 시험 및 평가, 리팩토링>

김 영 철(정회원)



- e-mail : bob@selab.hongik.ac.kr
- 2000년 : Illinois Institute of Technology(공학박사)
- 2000년 ~ 2001 : LG 산전 중앙연구소 Embedded system 부장
- 2001년 ~ 현재 : 홍익대학교 컴퓨터정보통신 부교수

<주관심분야 : 테스트 성숙도 모델, 임베디드 S/W 개발 방법론 및 도구 개발, 모델 기반 테스트, CBD, BPM, 사용자 행위 분석 방법론>