

MDA 기반 모바일 컨버전스 임베디드 소프트웨어 개발 도구 및 방법론에 관한 연구

박은주, 김행곤
대구가톨릭대학교 컴퓨터공학부
e-mail:{ejpark,hangkon}@cu.ac.kr

A Study on Tool and Methodology of Embedded Software Development for Mobile Convergence using MDA

Eun-Ju Park, Haeng-Kon Kim,
Dept. of Computer Information & Communication Engineering,
Catholic University of Daegu

요 약

모바일 인터넷과 다양한 정보단말기의 보급, 소프트웨어 플랫폼의 다양화와 복잡도가 증가하면서 소프트웨어 간에 상호운용성과 이식성의 결여로 인해 플랫폼 독립적인 새로운 소프트웨어 개발 방법이 필요하다. 또한 특정 플랫폼에서 개발된 소프트웨어들은 특정 단말기를 통해 연동되므로 차세대 컨버전스 기술을 위한 미들웨어 융합에 관한 연구도 필요하다. 따라서 본 논문에서는 플랫폼 독립적인 소프트웨어 개발과 변환을 위해 MDA 기반 소프트웨어 지원 도구 개발과 다양한 플랫폼을 지원하는 미들웨어 융합 시스템 개발 및 모바일 컨버전스 임베디드 소프트웨어 품질 보증에 대한 연구를 통해 차세대 품질보장형 모바일 컨버전스 서비스 제공을 위한 기본적인 기능 구조를 제공하고자 한다.

1. 서론

21세기 세계 IT 산업은 아날로그 기술의 퇴조와 더불어 디지털 기술과 인터넷 확산으로 반도체, 디지털 가전, 컴퓨터, 방송 등 다양한 산업 분야가 융합되어 새로운 부가가치를 창출하는 모바일 컨버전스 환경으로 빠르게 진화되고 있다[1]. 소프트웨어의 개발 기술은 상호 이질적인 환경과 다양한 소프트웨어 플랫폼 기술의 컨버전스화, 글로벌 분산 환경 및 다양한 정보 단말을 지원하는 서비스 환경의 다양화, 구현 형태의 변화 및 웹서비스 기술 발전의 급격한 변화로 인해 기존 소프트웨어 개발 방법만으로는 상호 운영성의 결여 및 소프트웨어 개발의 어려움과 이식성 부족의 결과를 가져왔다[2]. 따라서 다양한 플랫폼의 등장에 따른 플랫폼 독립적인 소프트웨어 설계 방법과 서로 다른 플랫폼 간의 컴포넌트 변환이 용이한 새로운 방법이 필요하다.

또한, 유비쿼터스 컴퓨팅의 핵심적인 기술인 임베디드 소프트웨어(embedded software) 기술에 관

한 연구가 활발히 진행되면서 임베디드 소프트웨어 산업은 기술혁신과 시장에서의 치열한 경쟁에 의해 새롭고 역동적인 양상으로 전개되고 있다. 많은 임베디드 소프트웨어 업체들은 임베디드 소프트웨어 요구에 대응하기 위해 더욱더 경쟁적인 환경을 조성해 나가고 있다. 이러한 노력으로 최근 우리 주변에 있는 가전기기 및 소형 시스템의 지능화를 가능하게 하고, 앞으로는 각종 센서들이 우리 삶의 주변을 채울 것으로 보여진다[3,4]. 이들 임베디드 시스템들이 다양한 하드웨어 플랫폼과 기기 상에서 연동되기 위해서는 이를 지원하는 미들웨어 기술이 필요하다.

따라서 사용자의 다양한 서비스 요구와 변경이 빈번한 임베디드 소프트웨어의 특성을 적시에 반영할 수 있는 새로운 방식의 소프트웨어 개발 기술이 필요하며, 이들 소프트웨어는 다양한 통신 프로토콜 지원, 다양한 서비스 지원, 다양한 네트워크 지원이 필수적인 미들웨어 기능을 포함해야 한다. 그리고 임베디드 소프트웨어를 안정적이면서 신뢰성 있게

개발하기 위해 소프트웨어의 특성에 따른 다양한 요구를 반영한 기능을 효과적으로 평가하는 환경 및 도구의 개발이 절실하다.

2. 관련연구

2.1 모바일 컨버전스(Mobile Convergence)

최근 부피가 큰 아날로그 부품이 디지털로 대체되면서 기기의 소형경량화, 멀티미디어화가 심화되고, 이로인해 모바일 확산이 더욱 가속화되고 있다. 휴대기기에 카메라, 게임, 멀티미디어 기능이 부가되고 새로운 개념의 서비스들이 등장하면서 모바일 기기에 새로운 기능과 서비스가 결합되는 컨버전스화도 가속되고 있다[1,5].

모바일 컨버전스(Mobile Convergence)는 휴대기기에 새로운 기능, 서비스, 미디어 등이 융합되는 것을 의미한다. 휴대폰에 카메라, MP3, TV 기능 등이 결합되면서 기존의 전화기와는 차원이 다른 개인용 기기로 변모한 것이 그 예라 할 수 있다[5].

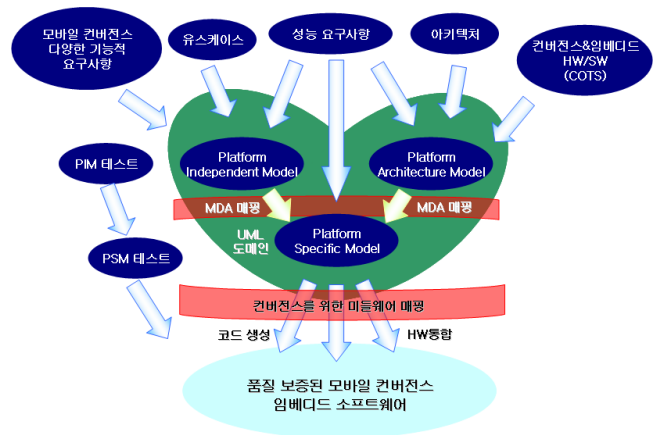
2.2 MDA(Model Driven Architecture)

MDA는 OMG가 그동안 만들어낸 플랫폼 기술과 표준 모델링 언어(UML 등)를 이용하여 구현된 여러 산업의 표준안을 결합한 모델 방식의 새로운 소프트웨어 아키텍처이다. 특히, 닷넷, 자바, J2EE 등 신기술이 등장하면서 이들을 모두 지원하는 거대한 표준 아키텍처가 요구되면서 이를 충족하기 위해 탄생되었다. MDA는 모델과 다양한 플랫폼과 프로그래밍 언어에 매핑하는 표준을 만들고, 표준에 따라 도구들이 소스를 생성하도록 함으로써 어떤 도구에서도 동일한 모델에 대해서 동일한 소스를 만들도록 하고자 하는 것이다.

MDA의 핵심은 메타모델을 기반으로 구현 환경에 독립적인 시스템을 개발하고, 이를 자동으로 구현 환경에 배치함으로써 구현 단계에서 생산성 향상, 표준 메타모델에 기반을 둔 시스템들의 일반적 호환성 확보가 가능하다는 점이다. MDA는 MDD SW 개발 방법의 표준으로 인정되고 있으며 향후 전세계 산업계 전반에서 사용될 전망이다[2,6].

3. MDA 기반 모바일 컨버전스 임베디드 소프트웨어 개발 도구 및 방법론

본 논문에서 제안하는 MDA 기반 모바일 컨버전스 임베디드 소프트웨어 개발 도구 및 방법론을 위한 전체적인 개요는 (그림1)과 같다.

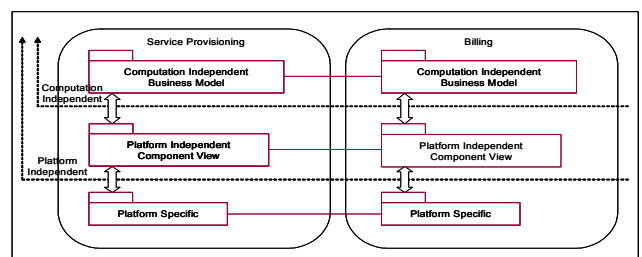


(그림 1) MDA 기반 임베디드 소프트웨어 개발

- MDA 기반 모바일 컨버전스 임베디드 소프트웨어 개발 방법과 도구 개발 : MDA 기반 모바일 컨버전스 임베디드 소프트웨어 개발 프로세스, 아키텍처, 자산 개발 및 관리 기법, 이를 기반으로 자동화된 지원도구
- 모바일 컨버전스 지원을 위한 미들웨어 시스템 개발 : 모바일 컨버전스를 위한 미들웨어 요소 기술 분석, 미들웨어를 설계, 구현
- 모바일 컨버전스 임베디드 소프트웨어 품질 보증 : 모바일 컨버전스 임베디드 소프트웨어 품질 보증 특성 및 품질 매트릭스 정의, 품질 프로세스 및 품질 표준 템플릿 작성을 통해 소프트웨어 품질 관리 도구

3.1 MDA 기반 임베디드 소프트웨어 설계 및 도구

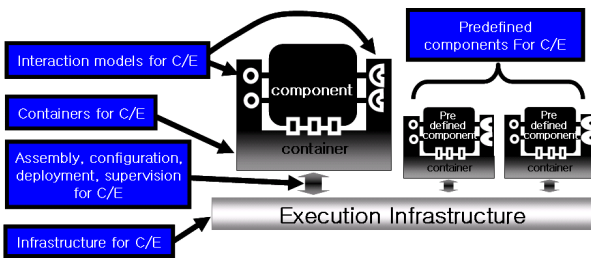
MDA는 핵심이 되는 시스템 모델들을 분리하고 그들 사이에 일관된 구조를 정의한다. 한 시스템의 PIM(Platform Independent Model)과 PSM(Platform Specific Model)의 구조와 관계를 정의할 뿐만 아니라 연관 관계를 가지는 시스템간의 PIM과 PIM 또는 PSM과 PSM의 관계에 대한 정의도 포함한다.



(그림 2) MDA에서의 모델 분리와 모델간 관계

<표 1> 모델 변환의 종류

PIM to PIM	PIM에서 PIM으로의 변환은 PIM이 개발 단계에서 좀더 상세화되는 경우다. 상세화가 이루어지더라도 변환된 PIM에 특정 기술관련 사항이 기술되지 않는다.
PIM to PSM	PIM이 실제 시스템의 기능을 충분히 기술하고 있다면 기술 종속적인 정보를 추가하여 PSM으로 변환한다. 변환된 PSM도 UML로 기술한다.
PSM to PSM	PSM과 PSM의 변환은 PIM to PIM 관계와 같이 상세화 관계이다. 하지만 이 변환에서는 실제 구현과 관련된 정보가 포함된다. 예를 들면 컴포넌트가 실제로 실행되기 위한 컨테이너정보, 설정정보, 배포정보를 추가하는 것이 이러한 변환에 해당한다.
PSM to PIM	기존의 시스템 구현 상황을 추상화하여 PIM 모델을 얻어내는 과정이다.



(그림 3) MDA 기반의 컴포넌트 조립 도구 구조

(그림 2)는 3개의 추상화 수준으로 모델링된 두 개의 시스템과 두 시스템의 같은 추상화 수준에 있는 모델간의 연관관계를 보여주는 MDA의 모델간 관계의 정형적인 구조이다.

<표 1>은 MDA에서 가능한 모델 변환의 종류를 나타내고, (그림 3)은 MDA 기반의 컴포넌트 조립 도구의 구조를 나타낸 것이다.

3.2 모바일 컨버전스 지원을 위한 미들웨어 시스템

컨버전스는 모바일 환경 하에서 제품간, 기능간 융합을 촉진시켜 사용자들이 하나의 기기로 다양한 기능 및 서비스를 제공하기 위해 사용하는 기반 기술이다. 최근에는 음성 서비스 보다 데이터 서비스 위주의 다양한 응용 서비스를 가능하게 해주는 미들웨어 컴포넌트의 중요성이 더욱 커지면서, 컨버전스의 다양한 서비스를 경쟁력 있게 수용할 수 있는 표준 플랫폼이 요구된다.

모바일 컨버전스를 위한 미들웨어 개발 기술은 셋탑 박스, 스마트폰, 모바일 장치와 같은 "Web-connected" 장비 및 산업용 제어 및 과학용 계측을 포함한 매우 제한된 메모리 장치에 수행될 수 있도록 작은 공간을 갖는 임베디드 어플리케이션

의 작성을 지원해야 한다. 이러한 특수 목적의 미들웨어를 설계할 때 고려할 사항은 다음과 같다.

- 미들웨어는 공통 추상화가 같은 도메인의 다른 어플리케이션 사이에 재사용될 수 있도록 충분히 일반적이어야 한다.
- 각 특정 어플리케이션의 요구 사항에 미들웨어를 맞추기 위해 세밀한 변경을 할 수 있어야 한다.

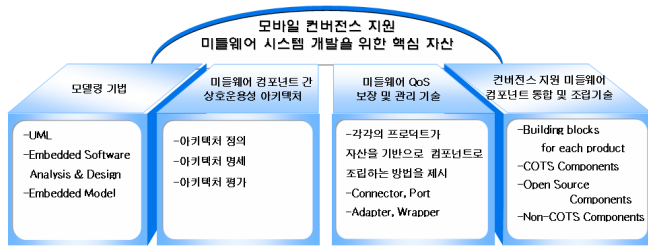
미들웨어를 개발할 때 두 가지 접근 방법은 기존의 범용 미들웨어 프레임워크(예, TAO)를 분할하는 하향식방법과 하위 수준의 하부 구조(예, ACE)로부터 특수 목적의 미들웨어를 작성하는 상향식 방법이 있다. 두 방법 모두 기능의 재사용과 어플리케이션에 특정한 요구 사항의 커스터마이징과의 균형을 구한다.

임베디드 미들웨어 하부 구조를 작성하는데 가장 중요한 설계 문제 중의 하나는 미들웨어 하부구조가 은닉하려는 특성이 무엇인가이다. 예를 들면, 분산과 자동화된 재구성을 프로그래머로부터 숨길 것인지, 인터랙션을 위한 장치의 자동적인 선택을 숨길 것인지, 장치의 발견과 개인화를 숨길 것인가이다. 따라서 하나의 특정 도메인을 지원하는 특정한 상위 수준의 추상화를 개발하는 것이 중요한 반면 그 추상화는 다양한 어플리케이션을 다룰 수 있도록 일반적이어야 한다. 아울러 미들웨어 하부 구조는 각 어플리케이션의 요구에 따라 동적 재구성과 정보 표현을 구현할 수 있도록 융통성이 있어야 한다.

기존 미들웨어의 기능에 추가하여 모바일 컨버전스를 위한 임베디드 미들웨어는 기존 미들웨어가 제공하는 기능 이외에 어플리케이션이 개발된 이후에 기능상의 요구 또는 운영 조건의 변화에 대응한 어플리케이션 적응(adaptation) 특성이 제공되어야 한다. 이러한 적응 특성은 크게 정적 및 동적 적응 타입으로 분류될 수 있으며 정적 적응 타입에는 두 가지 유형이 있다.

- customizable: 어플리케이션 컴파일(또는 링크)동안 어플리케이션을 변경해서 개발자가 어플리케이션의 커스터마이징된(적응된) 버전을 생성할 수 있도록 한다.
- configurable: 어플리케이션 시작 때에 변경을 가능케 한다. 관리자가 컴파일 이후에 실현된 기능 및 운영상의 변화에 응답하여 미들웨어를 구성할 수 있도록 한다.

본 미들웨어 개발에서는 어댑티브 미들웨어(adaptive middleware)를 개발하고자 하며 임베디드 미들웨어의 개발을 위한 어댑티브 특성을 포함하여



(그림 4) 미들웨어 기술 지원을 위한 핵심 기술

다음과 같은 세 개의 핵심 소프트웨어 기술을 식별하고 이를 통하여 미들웨어를 개발하고자 한다(그림 4).

- 컴포넌트 기반 설계 : 미들웨어 기능을 컴포넌트로 분할하여 정적 및 동적으로 미들웨어의 구조를 변경하거나 관리하기 쉽도록 한다.
- 관점지향(aspect-oriented) 프로그래밍 : 미들웨어 핵심사항 (QoS, 전력 소비, 보안 및 결합허용)을 개발시에 분리하여 컴파일이나 런타임 때 이들 고려 사항이 코드로 선택적으로 결합되도록 한다.
- 소프트웨어 설계 패턴 : 어댑티브 미들웨어에 가상 컴포넌트 패턴과 같은 설계의 재사용을 이용한다.

3.3 임베디드 소프트웨어 품질 보증 기술

모바일 임베디드 소프트웨어의 품질을 보증하기 위한 품질보증 구조는 (그림 5)와 같이 구성된다. 먼저 임베디드 소프트웨어의 특징분석 즉, 목적의 한정성, 실시간 처리성, 강한 내구성 등은 핵심특성으로 이들 특성을 고려한 품질 보증 기술이 개발되어야 한다. 또한 MDA 기반의 PIM과 PSM에 대한 테스트를 통하여 최적의 고품질 어플리케이션을 설계해야 한다. 임베디드 소프트웨어 테스트 베드 구축에서는 응용 소프트웨어에 대한 테스트 도메인 분석과 테스트 데이터 생성 및 테스트 드라이버를 구축한다. 이때 테스트 에이전트를 통하여 실제 타킷에 탑재되어 호스트와 통신하면서 타킷의 이슈들을 호스트에 전달한다. 임베디드 소프트웨어의 품질 프로세스 프레임워크를 구축하여 best practices를 설정하고 프로세스 능력을 측정한다. 품질관리 도구를 개발하는 것으로 RO예측 등 Value engineering을 통한 품질관리 도구를 개발하여 비용 개발기간 예측, 결합 등을 관리 하도록 한다.

4. 결론

현재 임베디드 시스템 개발에 관해 연구는 소프트웨어보다는 하드웨어 중점적으로 연구·개발되고 있다. 임베디드 소프트웨어 개발에 관한 연구는 실



(그림 5) 임베디드 소프트웨어 품질보증 구조

제 적용면에서 특정 시스템에 매우 제한적이다. 즉, 기존 소프트웨어 개발 방법만으로는 설계 시 플랫폼에 종속적인 모델링을 할 수 밖에 없어 소프트웨어 간에 상호운영성과 이식성이 결여된 상태이다. 또한, 모바일 인터넷과 다양한 정보단말기의 보급에 따라 소프트웨어의 빠른 개발을 위해서는 다양한 소프트웨어 개발 환경과 플랫폼을 지원해 주면서, 플랫폼이 다른 소프트웨어 간에 변환을 쉽게 해주는 도구의 조기 개발이 필요하다.

본 논문에서는 이러한 문제를 해결하기 위하여 다중 플랫폼 지향 개발 방법으로 MDA를 기반으로 한 임베디드 소프트웨어 개발 방법론, 그리고 모바일 컨버전스 즉, 개별적인 서비스들을 하나의 상품이나 서비스로 통합하고 단일 접속 장치를 통해 다양한 서비스에 접속할 수 있도록 지원하기 위한 미들웨어 기술 및 임베디드 소프트웨어 품질 보증 기술에 대해 제시하였다.

참고문헌

[1] 이지평, 강선구, “디지털 컨버전스에 따른 뉴트렌드”, LG 주간경제, LG경제연구원, 2002.
 [2] Ragnhild Kobro Runde and Keril Stolen, “What is Model Driven Architecture”, University of Oslo Department of Informatics, Research Report 304, ISBN 82-7368-256-0, 2003.
 [3] 2002 Embedded Software Tools Worldwide Forecast, Gartner Dataquest Market Statistics 110850, 2002.
 [4] Worldwide Embedded Software Tools Outlook, Gartner Dataquest Alert, 2002.
 [5] 민병석, 임태윤, 권기덕, “모바일 컨버전스의 확산과 대응”, CEO Information 497호, 삼성경제연구소, 2005
 [6] “SW 생산에 있어서 MDA와 MDD”, 전자상거래연구조합, 2004
 [7] Tatsuo Nakajima, etc., “Middleware design issues for ubiquitous computing”, Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia, pp.55-62, 2004.