모바일 플랫폼의 MDA 적용 방안

김철현*, 이동수*, 이민태*, 김병기*
*전남대학교 전자컴퓨터공학과
e-mail:ch-kim@chonnam.ac.kr

MDA Application Plan of Mobile Platform

Chul-Hyun Kim*, Dong-Su Lee*, Min-Tae Lee*,
Byung-Ki Kim*
*Dept of Electronic & Computer Engineering, Chonnam University

요 약

본 논문에서는 Model Driven Architecture를 다양한 모바일 플랫폼에서 적용하는 방안에 대해 설명한다. 모바일 플랫폼은 Symbian OS, Microsoft Windows CE 등 다양한 종류가 있으며, 이들의 어플리케이션을 재사용하기 위해서는 각 플랫폼에 맞는 언어로 다시 개발해야 한다. MDA는 이러한 이기종의 플랫폼에 적용할 수 있는 가장 효율적인 아키텍처이다. PIM 모델을 작성하고 변환규칙을 적용한 자동화 도구로써 PSM 모델과 소스코드까지 자동으로 생성이 가능하기 때문에 높은 개발 생산성과 이식성, 상호운용성을 제공할 수 있다.

1. 서론

모바일 소프트웨어 개발 기술은 상호 이질적인 환경과 다양한 소프트웨어 플랫폼 기술, 분산환경 및 다양한 정보단말을 지원하는 서비스 환경의 다양화, 구현 형태의 변화로 인해 기존 소프트웨어 개발 방법만으로는 상호 운영성의 결여 및 소프트웨어 개발의 어려움과 이식성 부족의 결과를 가져왔다[1].

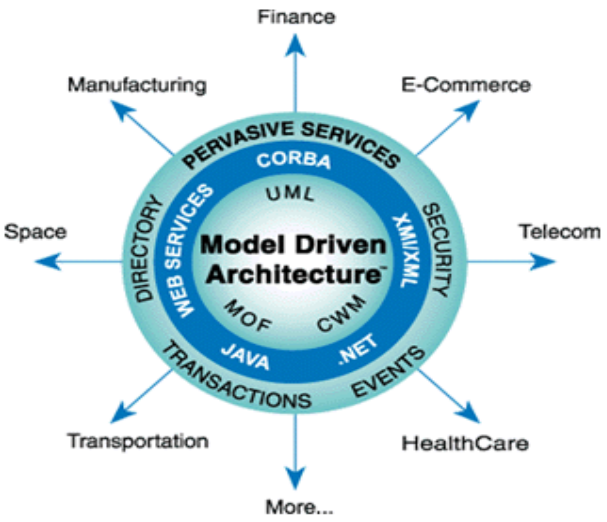
이로 인해 다양한 플랫폼에 맞는 소프트웨어를 자동으로 생산할 수 있도록 소프트웨어 모델 자동 매핑과 변환 기술을 중심으로 하는 MDA(Model Driven Architecture)[2,3,4] 기반 소프트웨어 개발 방법이 중요하게 대두되고 있다.

본 논문에서는 상이한 모바일 플랫폼에서 소프트웨어를 개발하는 문제점에 관해 이야기 하고, 이를 해결하기 위한 방안으로 Model-Driven Architecture와 VMTS (Visual Modeling and Transformation System)를 활용한 방안을 제시한다.

2. 관련연구

2.1 MDA (Model Driven Architecture)

MDA 기반 소프트웨어 개발 방법은 플랫폼 독립적인 모델(PIM: Platform Independent Model)로부터 플랫폼 종속적인 모델(PSM: Platform Specific Model)로 자동으로 변환하고 소프 코드를 자동으로 생성하는 방법으로서 원하는 플랫폼에 맞는 소프트웨어를 쉽고 빠르게 개발할 수 있는 새로운 개발 방법으로 등장하였다. OMG에서 2001년 3월에 기술 표준 방향에 대하여 합의를 한 후, 2001년 9월에 새로운 소프트웨어 아키텍처로 MDA를 전면적으로 채택하였다.

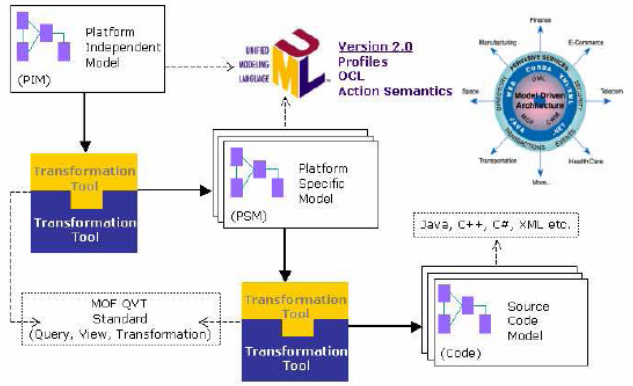


(그림 1) MDA의 기본 요소

MDA는 여러 가지 모델과 표준들에 의해서 구성된다. 모든 MDA 모델 들은 MOF(Meta Object Facility)[4]라는 추상적인 메타모델에 기초를 두고 있기 때문에 모두 연관되어 있다. 다시 말해 모든 MDA 모델은 MOF와 호환성이 있다. 이런 특징은 MDA에서 이용된 모든 모델들이 다른 MOF 호환 모델들과 호환성을 가진다는 것을 보장한다.

MDA에서 또 한 가지 중요한 부분이 UML 프로파일(profile)이다. UML 프로파일은 UML을 확장한 것이기 때문에 MOF와 호환성이 있으며, UML 2.0에서는 UML의 다양한 기능적인 사용 방법을 기술할 수 있다[5,6]. UML

프로파일은 UML의 확장인 동시에 그 자체로 MOF 메타 모델이다. 일부의 MDA 메타모델은 이미 OMG에서 과거에 정의된 것들이며 일부는 OMG 외부의 그룹 또는 벤더들에 의해 MOF 호환이 되는 형태로 만들어진 것이다. 이러한 OMG 외부의 메타모델들은 비록 공식적인 OMG 표준은 아니지만 MOF 호환성이 있기 때문에 MDA 개발 과정에는 사용되는 데 문제가 없다. 대표적인 MDA 모델들과 프로파일들로는 다음과 같은 것들이 있다.



(그림 2) MDA 개발 공정

2.1.1 CWM

CWM(Common Warehouse Metamodel)은 OMG에서 표준화한 데이터 웨어하우스를 관리하는 데 이용되는 메타 데이터 모델이다. CWM을 이용하면 개발자들이 관계형 데이터베이스 테이블, 레코드, 구조체, OLAP, XML 그리고 다차원 데이터베이스 디자인 등과 같은 수많은 데이터 모델 또는 포맷을 생성할 수 있다. 또한 CWM에는 데이터 웨어하우스 이외에도 사용될 수 있는 유용한 부분들도 있는데 예를 들면 데이터 모델이나 데이터 변환, 소프트웨어 배포 등에 관련된 내용도 포함되어 있다.

2.1.2 UML 메타모델

UML의 초기 버전은 MOF와의 완전한 호환성이 보장되지 않지만 UML 2.0은 MOF 호환성을 가진다. 이런 이유로 진정한 MDA 개발을 위해서는 UML 2.0이 사용돼야 한다. UML은 핵심 모델링 개념들과 이들을 위한 다이어그램들로 정의된다. 또한 개발자들이 UML 구성 요소에 다양한 제한(constraint)을 할 수 있도록 허용하고 있다. UML 2.0에서는 UML이 모델들의 행위를 지정할 수 있으며 이러한 행위의 표현들이 직접 코드로 변경된다.

2.1.3 XMI

XMI 표준은 MOF와 호환성이 있는 모델들을 XML 문서 형태로 표현하고 이를 MOF 호환 데이터베이스에 저장할 수 있도록 하는 표준이다. 그러므로 사실상 XMI 문서는 곧 MOF XML 문서라고 할 수 있다.

2.1.4 UML 프로파일

UML 프로파일은 특정 도메인에 대한 UML 모델을 작성하기 위한 일반 확장(generic extension) 메커니즘을 정의하는 것이다. 그러므로 UML 프로파일은 추가적인 스테레오타입(stereotype)과 태그 값(tagged value)이 적용된 엘리먼트(element), 애트리뷰트(attribute), 메쏘드(method), 링크(link) 등으로 이뤄진다. UML 프로파일은 이러한 확장 컬렉션을 이용해서 특정 도메인에 대한 모델링 문제를 기술하고 해당 도메인의 내용들을 모델링할 수 있도록 해준다.

현재 다양한 UML 프로파일이 나와 있으나 MDA와 관련해서는 플랫폼에 따라 각각의 프로파일이 정의될 수 있다. 현재 CORBA 프로파일, EAI 프로파일, EDOC 프로파일, 리얼타임 컴퓨팅을 위한 스케줄링 프로파일 등이 OMG에서 정의되었으며 EJB 프로파일은 JCP(Java Community Process)를 통해, 닷넷 프로파일은 마이크로소프트에 의해 정의되고 있다.

MDA 기반 소프트웨어 개발 기술의 핵심 요소 기술은 플랫폼 독립 모델 생성 기술, 플랫폼 종속 모델 변환 기술, 모델 매핑 자동화 기술 및 MOF(Meta Object Facility)[4] 기반 정보저장소(Repository) 기술이다. 플랫폼 독립 모델 생성 기술은 UML(Unified Modeling Langage)[5,6] 또는 EDOC(Enterprise Distributed Object Computing Specification)[7] 기반의 플랫폼 독립 모델을 생성하고 모델의 완전성을 검증하는 기술이며, 플랫폼 종속 모델 변환 기술은 OMG에서 표준으로 재정한 UML for EJB[8], UML for CORBA[9,10] 및 표준 제정중이 UML for .NET, UML for SOAP를 지원하는 매핑 명세에 따라 플랫포 독립 모델을 플랫폼 종속 모델로 자동으로 변환하는 기술이다. 모델 매핑 자동화 기술은 모델 간의 변환을 위한 매핑 규칙 및 매핑 엔진을 정의하고, 다양한 UML 프로파일을 정의하고 검증하는 기술이며, MOF 기반 정보저장소 기술은 모델 정보를 체계적으로 저장하고 관리하는 기술이다.

2.2 Mobile Platform

2.2.1 Symbian OS

심비안은 노키아 소니 에릭슨 지멘스 등 유럽의 이동통신 장비업체들이 마이크로소프트의 컴퓨터 응용시스템에 의존하지 않기 위해 1998년부터 컨소시엄을 결성해 개발한 휴대폰 PDA 스마트폰 등 모바일용 컴퓨터 운영체제 OS 이다.

OMAP(Open Multimedia Application Platform)로 Texas Instrument사의 DSP와 ARM Processor를 기반으로 하고 EPOC을 통하여 3G 이동단말의 표준 OS로 지정을 위하여 Symbian, Nokia, Texas Instrument사가 적극적인 활동을 모색하고 있다. Symbian이 관심을 받는 이유는 Microsoft가 시장 확대를 노리는 고성능 휴대폰 OS시장에 기술적으로 대응할 수 있는 거의 유일한 솔루션이며

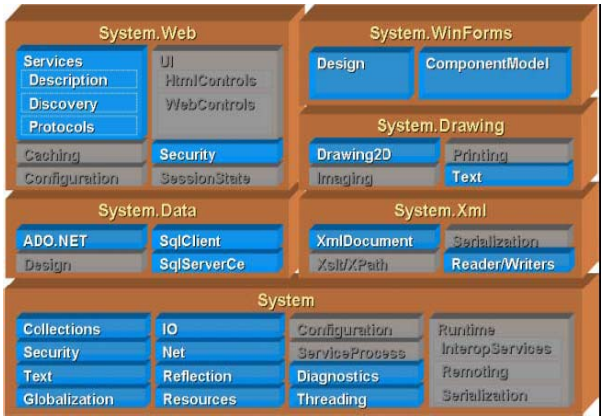
현재까지 스마트폰 OS 시장의 약50%를 점유하고 있다.

Symbian OS는 크게 Base를 기반으로 Security, Telephone, Multimedia, Communication, Application Framework, PAN, Application Engine, Messaging, Java 등으로 나뉜다. 즉, Contacts, Schedule, Messaging, Browsing 등의 다양한 application engine, 녹음, 녹화, 재생, 스트리밍의 A/V 기능, 그래픽 가속 API등이 제공죄는 Application Framework, Ipv4/v6, IrDA, Bluetooth등의 통신기능, 3G 기술까지 모두 포함하는 bearer/SIM 기능, 인증서 관리, WIM framework, secure protocol등의 보안 기능들을 모두 제공한다.

2.2.2 .NET Compact Framework

.NET Compact Framework는 스마트 장치용 응용 프로그램을 개발할 때 사용하며 .NET Framework 클래스 라이브러리 일부와 모바일 장치에 필요한 클래스 라이브러리를 가지고 있다. 그리고 WebDAV는 인터넷을 통한 다양하고 광범위한 콘텐츠의 비동기적 협업 저작을 지원하는 표준하부구조이며, HTTP1.1 프로토콜의 확장을 통하여 원거리 사용자들에게 서버의 파일을 수정하고 관리할 수 있도록 지원한다.

.NET Compact Framework[11]는 메모리, CPU와 파워에 제약이 있으면서 배터리 소모를 줄여야 하는 소규모 장치에서 효율적으로 실행될 수 있도록 공용 언어 런타임을 새롭게 구현한 것으로 Pocket PC, 휴대폰 등의 리소스가 제한된 컴퓨팅 장치에서 프로그램을 실행할 수있는 하드웨어 독립적인 환경이다.



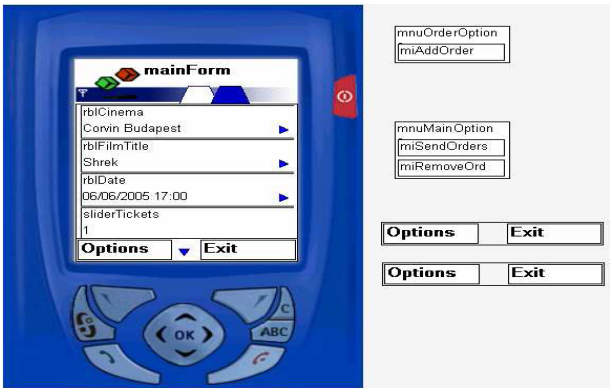
(그림 3) .NET Compact Framework 클래스 라이브러리 (음영처리 제외)

.NET Compact Framework 클래스 라이브러리는 공용 언어 런타임을 엄격하게 통합하는 재사용이가 가능한 클래스 컬렉션이다. 응용 프로그램은 이러한 라이브러리를 활용하여 기능을 파생한다. 객체 지향 클래스 라이브러리와 마찬가지로 .NET Compact Framework 형식을 사용해서 인터페이스 디자인, XML 활용, 데이터베이스 액세스 및 파일 I/O 등의 작업을 비롯한 다양한 일반 프로그래밍 작업을 수행 할 수 있다. (그림 3)은 .NET Compact Framework에서 사용할 수 있는 클래스 라이브러리를 기능별로 보여 주고 있다.

3. MDA 기반 모바일 소프트웨어 개발

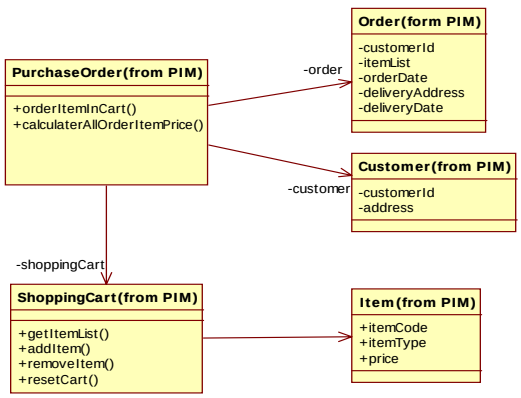
MDA 기반 모바일 소프트웨어 개발 과정은 다음과 같다.

- 1단계 : 구현 독립적 모델 작성(PIM)
- 2단계 : 구현 종속적 모델 작성(PSM)
- 3단계 : 어플리케이션 생성



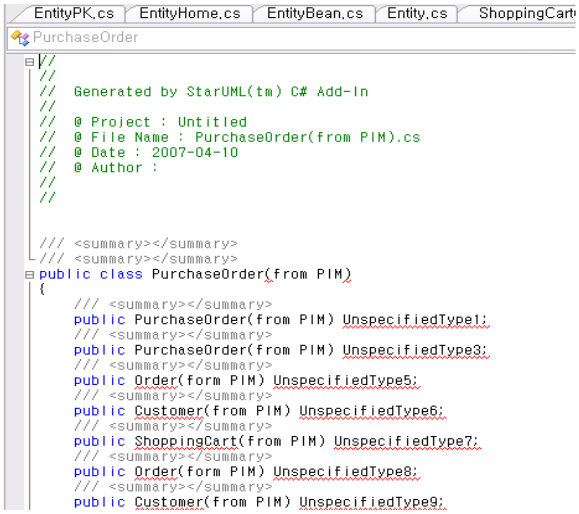
(그림 4) 자동 생성된 페이지, 버튼, 메뉴

Symbian의 리소스 모델을 통해 소스 코드를 생성하기 위해 VMTS(Vitual modeling and Transformation System)[13]을 사용하였다. VMTS는 에뮬레이터 기반의 모델-소스코드 변환 도구로서 UML Diagram을 작성하고 에뮬레이터에서 동작을 확인할 수 있고 VMTS의 VMP(Visual Model Processor)를 사용하여 각 플랫폼에 해당하는 소스코드를 생성해주는 도구이다. 개발자는 도메인 전문가의 요구사항을 분석하여 PIM 모델을 만든 후 VMTS를 통해 동작을 확인한다. (그림 5)는 쇼핑몰에서 구입을 원하는 상품을 선택하여 장바구니에 넣어놓고, 주문을 요청했을 때 주문이 되는 시스템의 PIM모델을 설명한 것이다.



(그림 5) 쇼핑몰 예제의 PIM

PIM 모델이 작성되면 이를 PSM으로 변환 시켜야 한다. 여기서 모델 변환을 위한 변환정보(transformation definition)을 기술해 주어야 한다. (그림 6)는 C# 변환정보를 입력하여 PIM에서 코드를 생성한 장면이다. 간단한 예제를 통해 모바일 플랫폼에서 사용가능한 소스코드를 생성할 수 있는지 확인해 볼 수 있다.



(그림 6) PIM을 통해 PSM 코드 생성

4. 결론

기존의 모바일 어플리케이션 개발 방법은 어플리케이션 설계 시 대략 전체적인 개념만 잡고 필요에 따라 세부 설계를 하여 개발 했고 한번 작성한 모델이 구현 단계에서 변화하는 경우가 빈번했다. 그러나 MDA 개발 프로세스에서는 PIM 구축시 모든 비즈니스에 대한 정적 구조 및 동적인 부분을 완벽하게 기술해야 한다. 그래야말 원하는 어플리케이션을 구축 할 수가 있다. 결국 MDA 개발 프로세스로 인해 구축돼야 할 핵심이 변화하고 이에 따라 개발자의 역할도 각 핵심을 실현하는 형태로 변화해야 한다. 기존 개발 방식은 대략 비스니스의 개념을 잡는 정도로만 모델링 해도 이후에 개발자가 직접 소스코드를 작성할 것이기 때문에 큰 문제가 없었으나 MDA 개발 방식에서는 기술한 비즈니스 규칙이 그대로 마지막 소스코드로 구현되기 때문에 상당히 주의해야 한다.

모바일 플랫폼의 대표적인 예인 Windows CE 와 Symbian은 각각의 어플리케이션 개발시에 종속적인 모델링을 할 수 밖에 없어 서로간에 상호운영성과 효율성, 이식성이 결여된 상태이다.

본 논문에서는 이러한 개발의 효율성과 이식성을 향상시킬 수 있는 방안을 제안한다. 제안된 방법은 모바일 플랫폼과 상관없는 플랫폼 독립 모델을 생성하고 변환규칙을 정의한 후 플랫폼 종속 모델을 작성한다. CASE 도구를 사용하여 모델에서 소스코드를 만들어내고 최종적으로 완전한 시스템을 구축한다.

참고문헌

[1] 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.

[2] OMG, "Model Driven Architecture", OMG document ormsc/01-07-01.

[3] Anneke Kleppe, Jos Warmer, and Wim Bast, "MDA Explained", Addison-Wesley, 2003.

[4] Stephen. J. Mellor, Kendall Scott, Axel Uhl, and Dirk Weise, "MDA Sistilled", Addison-Wesley, 2004.

[5] OMG, "MOF 2.0 Query/Views/Transformations RFP", OMG document ad/02-04-10, April 2002.

[6] OMG, "UML 2.0 Infrastructure Specification, OMG Final Adopted Specification", pctc/03-09-15, Sept. 2003.

[7] OMG, "UML 2.0 Superstructure Specification, OMG Final Adopted Specification", pctc/03-09-15, Sept. 2003.

[8] OMG, "UML Profile for Enterprise Distributed Objet Computing Specification", 2002.

[9] JSR-000026 UML/EJB™ Mapping Specification 1.0 Public Review Drift , http://jcp.org/aboutJava/communityprocess/review/jsr026/index.html

[10] OMG, "CORBA Components, version 3.0", formal/02-06-65, June 2002.

[11] OMG, "UML Profile for CORBA", OMG document formal/02-04-01, April 2002.

[12] .NET Compact Framework, "http://msdn.microsoft.com/smartclient/understanding/netcf/"

[13] VMTS, http://avalon.aut.bme.hu/~tihamer/researchr/vmts