

플랫폼 독립적인 모바일 서비스 프레임워크 기반 안드로이드 응용 설계

탁지우*, 김행곤**
 대구가톨릭대학교 컴퓨터정보통신공학과
 e-mail:{lebbenle*, hangkon**}@cu.ac.kr

Design of Android Application Platform On Independent Mobile Service Framework

Ji-Uoo Tak, Haeng-Kon Kim
 Dept of Computer information & Communication Engineering
 Catholic Univ. of Daegu, Korea

요 약

해를 더해갈수록 스마트폰 없이는 생활이 불편할 정도까지 왔다. 이에 따라 스마트폰과 스마트폰 관련 시장은 빠르게 발전하였다. 하지만 여전히 풀리지 않는 문제 중의 하나는 스마트폰을 구동시키는 플랫폼에 따라 각 플랫폼 별로 개발을 수행하는 번거로움이 있다. 이를 해결하기 위한 방법으로 MSF(Mobile Service Framework)는 각 기능을 하나의 서비스 모듈로 구성을 하고 이 서비스 모델이 각 플랫폼에 매핑 가능한 상태로 변환이 되는데 그 변환 결과를 저장소에 두어 다른 플랫폼에 적용 시 개발시간을 단축할 수 있는 프레임워크이다. 본 논문에서는 이러한 MSF를 적용한 예로 안드로이드 기반으로 어플리케이션을 개발하였다.

1. 서론

최근, 스마트폰을 구동시키는 대표적인 플랫폼으로는 iOS, Android, Windows 등이 있으며 각 플랫폼에 맞는 언어로 여러 어플리케이션이 개발되고 있다. 같은 어플리케이션 일 지라도 아이폰에 먼저 개발되었던 어플리케이션이 안드로이드 기반으로 후에 개발 된 경우, 안드로이드에서 호환 등의 여러 문제들로 아이폰 기반에 비해 사용자의 불편함이 있는 것도 사실이다. 이러한 문제에 따라서 여러 플랫폼들 간에 공통분모를 만들어 각 기능을 서비스로 두고, 각각의 서비스들은 플랫폼에 적용 가능한 의사코드로 변환을 하며 이러한 각각의 산출물들은 저장소에 저장하는 모바일 서비스 프레임워크(Mobile Service Framework)를 제시하고 이는 각 플랫폼 간의 호환성 확보를 위해 필요하다.

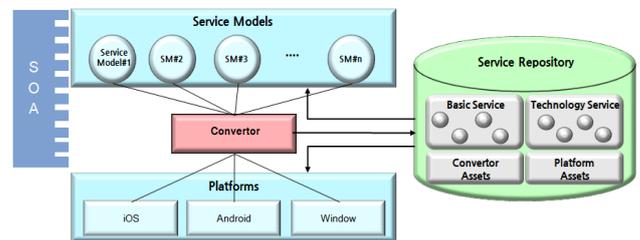
본 논문에서는 각 모바일 디바이스 및 OS에 독립된 어플리케이션 개발을 위해 제시되었던 MSF(Mobile Service Framework)를 적용하여 안드로이드 기반으로 어플리케이션 개발을 프로세스를 수행하여 실행 예로써 유저스토리 와 서비스 모델 추출, 플랫폼 독립 의사코드, UML 생성, 플랫폼 종속 구현 예 그리고 PSM원시코드(플랫폼 종속된 코드)를 결과로 제시한다.

2. 관련연구

2.1 Mobile Service Framework(MSF)구조

모바일 서비스 프레임워크(이하 MSF)의 전체적인 시

스템 구조는 (그림 1)과 같다.



(그림 1) MSF(Mobile Service Framework)구조

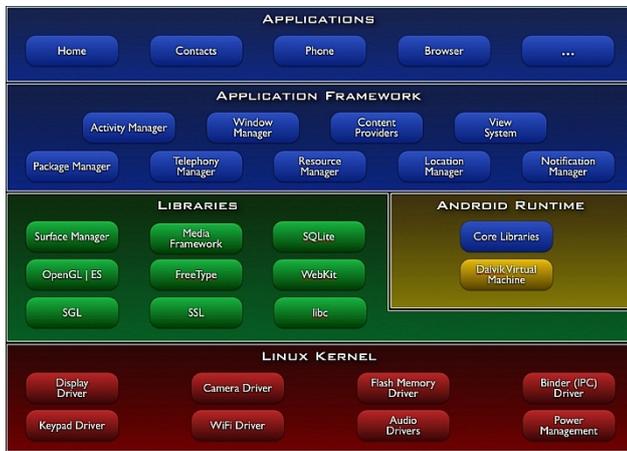
(그림 1)에서 서비스 저장소, 서비스 모델들의 집합, 컨버터와 플랫폼으로 구성되어 있으며 서비스 저장소에는 기본 서비스, 기술 서비스, 컨버터 자산, 플랫폼 자산으로 이루어져 있다. 이 서비스 저장소는 서비스 모델과 플랫폼에게 자료를 제공하고 컨버터로 부터의 결과물을 보관한다. 서비스 모델들은 MDA에서의 PIM단계라고 할 수 있으며 각각의 기능을 모델로 두어 그 모델들의 집합을 말한다. 이러한 서비스 모델들은 컨버터로 전송이 되고 컨버터에서는 전송받은 서비스 모델을 토대로 매핑(mapping)된 결과물을 서비스 저장소의 컨버터 자산에 저장한다. 여기서 서비스모델과 컨버터는 SOA를 기반으로 서비스 제공형태를 가지며 개발자가 어플리케이션을 개발함에 있어서 개발자는 기본서비스(Basic Service)와 기술 서비스(Technology Service)에 저장된 서비스들을 제공받아(served) 새로운 서비스 즉, 어플리케이션을 개발할 수 있다. 이러한 것은 SOA방식에 기반 한 개발방법으로 플랫폼

폼에 독립적인 어플리케이션 개발을 가능케 한다.

Service Repository는 서비스 저장소로써, 스마트폰에서 제공하는 여러 서비스들과 자산(assets)을 저장해 둔 저장소로 두 종류의 서비스(Basic service, Technology service)와 Convertor(컨버터)의 결과물을 저장하는 Convertor Assets(컨버터 자산), 각 Platform에 대한 정보를 담고 있는 Platform Assets(플랫폼 자산)가 있다.[3]

2.2 안드로이드 플랫폼

안드로이드 플랫폼은 운영체제, 미들웨어, 키 응용 프로그램들을 포함한 모바일 디바이스를 위한 소프트웨어 집합이다. (그림 5)는 안드로이드의 구조로 여기에서는 4개의 계층으로 분류되며 각각의 계층은 Application, Application Framework, Libraries, Android Runtime, Linux Kernel로 구성한다[4].



(그림 2) 안드로이드 아키텍처

(1) 어플리케이션

안드로이드 어플리케이션은 이메일을 확인할 수 있는 클라이언트, SMS프로그램, 캘린더, 지도, 브라우저, 주소록 등을 키 어플리케이션으로 제공하며 모든 어플리케이션들은 자바 언어로 작성한다.

(2) 어플리케이션 프레임워크

응용 개발자는 키 응용프로그램에서 사용한 것처럼 어플리케이션 프레임워크 API를 모두 접근하여 자신만의 어플리케이션을 개발한다. 어플리케이션 프레임워크는 각종 응용을 개발하기 위한 클래스와 메소드들을 제공한다. 안드로이드 아키텍처는 컴포넌트를 재사용하기 쉽도록 디자인 되어 있다. 하지만 프레임워크의 보안계약은 따라야 하는데 이와 같은 메커니즘은 컴포넌트를 사용자에게 의해 교체할 수 있도록 한다.

(3) 라이브러리

안드로이드는 안드로이드 시스템에서 다양하게 사용되는 C/C++ 라이브러리들을 포함하며, 안드로이드 Application Framework를 통해 개발자들은 이런 사항을 알 수 있다.

3. MSF(Mobile Service Framework)적용 예

3.1 유저스토리, 서비스모델 추출

MSF의 프로세스를 적용하여 어플리케이션 개발을 한 예로 먼저 시나리오 작성 전 서비스 모델 설계서를 제시한다.

<표 1> 서비스모델 설계서

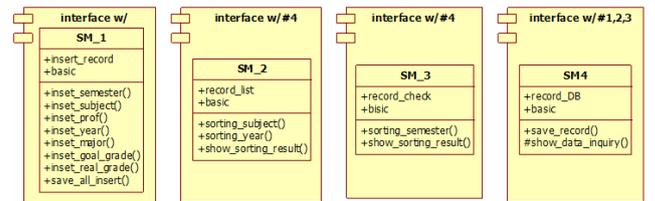
```
<service model #>
service.name(user define)
service.type(get_service repository)
service.attribute
service.function
```

위의 <표 1>은 시나리오를 토대로 서비스모델을 구성할 시에 필요한 서비스 모델 설계서이다. 서비스 모델의 순번을 기점으로 서비스의 이름은 개발자가 임의로 정하도록 하고 서비스 타입은 서비스 저장소에서 제공받은 서비스로 기본서비스 혹은 기술서비스를 말한다. 다음으로 각 서비스가 가지는 속성과 기능을 기재하여 하나의 모델이 완성되도록 한다.

(그림 3)은 MSF 프로세스를 적용한 어플리케이션 개발 예로써, 위에 있는 유저스토리를 토대로 서비스모델 설계서를 거쳐서 다음과 같은 서비스 모델을 생성한다.

****유저 스토리****

사용자는 성적 어플리케이션을 실행한다.
 어플리케이션 내 메뉴(성적입력, 성적조회, 성적리스트) 선택한다
 자신의 수강과목에 대한 정보를 입력하기 위해 성적 입력을 선택한다
 성적입력에 있는 학기, 과목명, 교수명, 학년, 목표학점, 학점을 입력하도록 한다.
 (이때, 목표학점까지만 입력하고 학점은 결과가 나온 뒤 본인이 입력 한다)
 성적조회는 학기 별로 sort하여 과목명과 목표학점, 획득학점을 결과값으로 보여준다.
 성적리스트는 학기 별, 학년 별로 선택 받아 결과값을 보여준다.



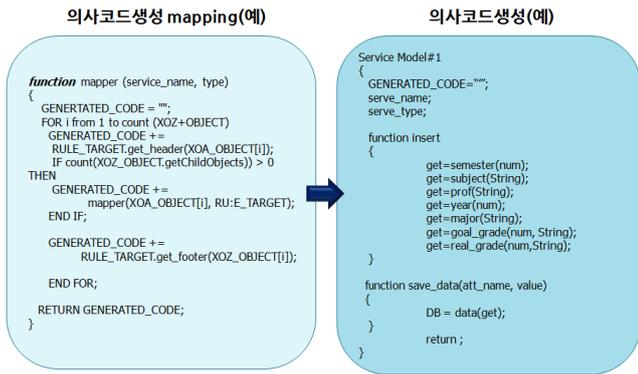
(그림 3) 유저스토리와 서비스 모델의 추출 예

서비스 모델은 총 네 가지로 구성을 했으며, SM_1은 성적에 관한 정보를 입력하는 모델이고 SM_2는 연별 과목에 대한 성적을 확인할 수 있는 성적 리스트, SM_3은 학기 별 과목에 대한 성적확인을 하는 성적확인 기능, 마지막으로 SM_4는 입력받은 정보를 저장하여 원하는 정보를 찾을 수 있도록 도와주는 데이터베이스 기능으로 각각의 서비스 모듈을 구성하였다.

이와 같은 모델추출은 SOA를 기반으로 모델을 추출한다.

3.2 플랫폼 독립 의사코드

추출한 서비스 모델을 토대로 컨버터에서 첫 번째 기능인 의사코드 생성을 하는 예제로 (그림 4)에 나타내었다.



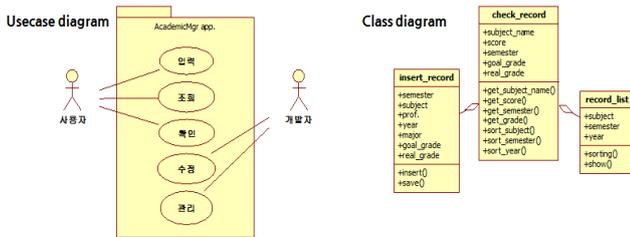
(그림 4) 의사코드 생성 예제

서비스 모델로부터 받은 정보를 토대로 (그림 4)의 왼쪽과 같이 의사코드를 생성하는 코드를 거쳐 오른쪽 그림과 같은 의사코드가 생성되는 과정을 볼 수 있다.

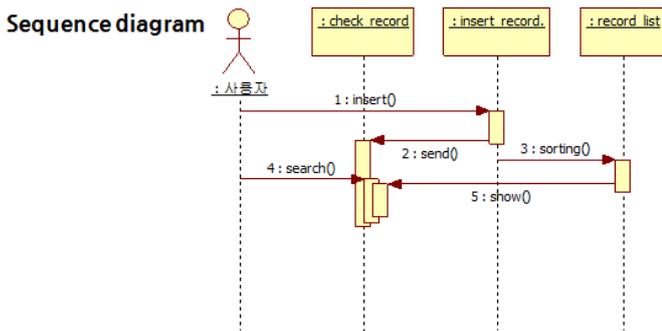
3.3. UML 생성

추출한 서비스 모델을 토대로 컨버터에서 두 번째 기능인 UML 생성 예이다.

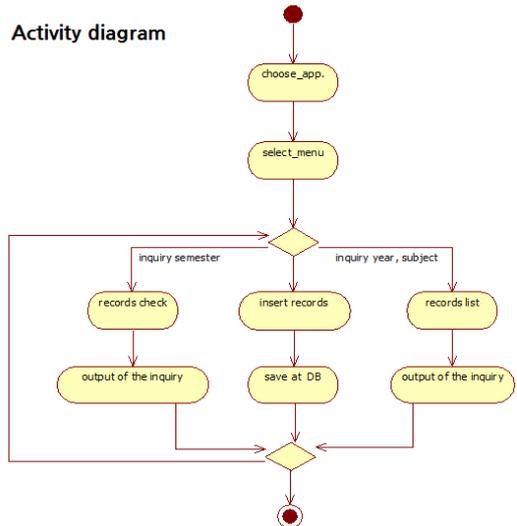
(그림 5)는 성적어플리케이션의 전체적인 UML로써 유스케이스 다이어그램에서는 사용자와 개발자가 성적어플리케이션을 사용하고 수정·관리하는 것을 볼 수 있고 클래스 다이어그램에서는 저장한 성적을 원하는 정보를 제공하기 위한 각 기능별 속성과 동작을 나타내며 (그림 6)의 시퀀스 다이어그램에서는 사용자가 어플리케이션을 사용하게 되는 순서, (그림 7)의 액티비티 다이어그램에서는 어플리케이션의 실행부터 종료까지의 수행과정을 볼 수 있다.



(그림 5) 성적관리 어플리케이션의 유스케이스와 클래스 다이어그램



(그림 6) 성적관리 어플리케이션의 시퀀스 다이어그램



(그림 7) 성적관리 어플리케이션의 액티비티 다이어그램

3.4 플랫폼 종속 구현 예

안드로이드 에뮬레이터 화면으로 성적어플리케이션을 실행한 결과는 (그림 8)의 왼쪽에서 확인할 수 있으며, (그림 8)의 오른쪽은 성적관리 어플리케이션을 실행한 초기 화면이다.



(그림 8) 에뮬레이터 실행화면과 어플실행화면

(그림 9)는 성적어플리케이션을 실행하여 성적입력 작업 수행 시 볼 수 있는 화면으로 학기, 과목, 교수님(과목담당 교수님), 학년, 목표성적과 획득학점 란이 있어 처음 기입 시 목표성적을 두고 점수 획득 후 다시 입력을 하여 목표와 획득학점 간의 차이를 알 수 있도록 한다. 이렇게 입력하여 저장(확인) 하면 데이터베이스에 저장이 되며 성적 조회 시 데이터베이스에서 검색한 결과를 바로 보여줄 수 있도록 한다. 저장 한 성적을 확인 시 (그림10)에서와 같이 과목별 혹은 학년 별 선택을 하여 확인하도록 한다. 특정과목에 대한 정보를 얻고자 할 시에는 과목별 선택으로 정보를 얻을 수 있고 자신의 학년 별 수강 과목들과 그 과목의 성적을 보고자 할 경우에는 학년 별 선택을 하여 원하는 정보를 데이터베이스 내에 저장된 정보를 가져온다.



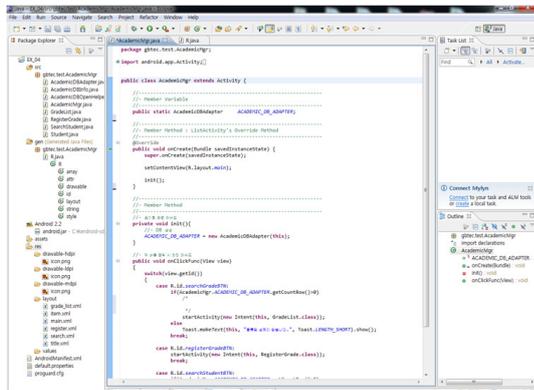
(그림 9) 성적어플리케이션 내 입력 화면



(그림 10) 성적확인 및 조회 화면

3.5. PSM 원시코드

성적관리 어플리케이션을 MSF를 적용하여 최종적으로 얻은 PSM원시코드이다. (그림 12)는 안드로이드에 종속된 코드로 소스 내용은 성적을 입력받아 저장을 하는 데이터베이스를 생성하고 어플리케이션 실행 시 수행할 수 있는 기능을 구성한 각 버튼 클릭 시 처리하는 메소드에 관한 내용이다.



(그림 11) 안드로이드 종속 PSM 코드

4. 평가

MSF와 같은 공통 프레임워크의 부재로 평가를 기존의 스마트폰 어플리케이션 개발 방법과 본 논문에서 제안한 MSF의 차이점을 평가로 하자면, 기존 방법론은 플랫폼에 종속적인 개발을 하였지만 MSF는 플랫폼에 독립적으로 개발을 한다. 그러므로 기존 플랫폼은 타 플랫폼에 적용 시에 다시 개발을 해야 하는 번거로움을 가지며 개발 및 유지보수에 많은 시간이 요구되는 반면 MSF는 독립적 개발로 타 플랫폼에 적용과 구현이 용이하며 모바일 개발 자산 저장소를 제공하여 서비스 재사용이 가능하며 개발 및 유지보수에 용이성을 제공한다.

<표 2> MSF와 기존 방법론 평가표

MSF	기존 방법론
- 플랫폼 독립적 개발	- 플랫폼 종속적 개발
- 타 플랫폼 적용, 구현 용이	- 타 플랫폼 적용 시 재개발
- 모바일 개발 자산 저장소 제공	- 개발 및 유지보수에 많은 시간 요구
- 서비스 재사용 가능	
- 개발 및 유지보수 용이성 제공	

5. 결론 및 향후연구

기존의 모바일 어플리케이션은 각 모바일의 플랫폼 별로 플랫폼에 종속된 OS기반에 플랫폼에 종속된 사용가능한 언어로 개발이 되어야 한다. 각각의 언어로 개발해야 하는 번거로운 작업을 줄이기 위해서 MDA와 SOA를 응용한 플랫폼에 독립적인 모바일 서비스 프레임워크인 MSF를 적용하여 플랫폼에 종속되지 않는 어플리케이션 개발방법을 제시하여 이기종 플랫폼 간 적용과 구현이 용이하고 서비스의 재사용으로 개발생산성을 증가시켜 최종적으로는 유지보수에도 용이성을 제공하고자 한다.

본 논문에서는 안드로이드의 예만 보였는데, 향후엔 타 플랫폼에도 성적관리 어플리케이션을 적용하고 다양한 어플리케이션 개발에 적용하여 평가하며, 또한 서비스 구성의 폭을 넓히고 주요 기능인 컨버터의 기능을 향상하고자 한다.

참고문헌

- [1] 김윤규, 이동훈, “국내의 스마트폰 어플리케이션 마켓 동향 분석”, 한국정보보호학회, 정보보호학회지, 제 21권, 제1호, pp26-37, 2011
- [2] 전중홍, 이승윤, “차세대 모바일 웹 어플리케이션 표준화 동향”, 전자통신동향분석, 제25권, 제1호, pp.100-113, 2010
- [3] 탁지우, 김행곤, “MDA+SOA 응용 플랫폼 독립적인 스마트폰 어플리케이션 개발에 대한 연구”, 제36회 한국정보처리학회 춘계학술발표대회, 제18권, 제2호, 2011
- [4] 김평중, “안드로이드 플랫폼과 어플리케이션 프레임워크 기술”, 정보처리학회지, 제17권, 제3호, pp.51-60, 2010