

## UML 프로파일에 기반한 모바일 어플리케이션의 PIM/PSM 변환 기법

최 윤 석

### A Transformation Technique of PIM to PSM based on UML Profiles for Mobile Applications

Yun-Seok Choi \*

#### 요 약

모델 기반 개발 방법은 모바일 어플리케이션 개발의 특성을 잘 수용할 수 있으므로 모바일 어플리케이션 개발에 적용하기 위한 다양한 연구가 진행 중이다. 그러나 모델 기반 개발 수행 시 PIM과 PSM 설계에 필요한 UML 프로파일의 체계적인 개발 관련 연구는 부족한 상황이며, 프로파일을 활용한 모델 변환 기법 관련 연구도 요구되고 있다. 이에 본 논문에서는 모바일 어플리케이션의 PIM과 PSM 설계에 적용할 UML 프로파일 구성 기법과 이를 활용한 모델 변환 기법을 제안한다. 제안한 기법은 모바일 어플리케이션의 특성을 반영하여 스테레오타입을 분류한 후 프로파일 구성 지침에 따라 각 모델에 적용할 프로파일을 구성한다. 프로파일을 적용하여 구성한 PIM은 모델 변환 절차에 따라 프로파일 간 대응 규칙에 의하여 초기 PSM으로 변환되며, 모바일 플랫폼의 세부정보를 반영하여 구성된 PSM 템플릿을 적용하여 정제된 PSM으로 변환 생성된다. 제안한 기법의 유용성을 확인하기 위하여 안드로이드 기반 위치 기록 어플리케이션을 대상으로 제안 기법을 적용하여 개발하고, 기존의 기법들과 비교를 수행하였다.

▶ Keyword : 모바일어플리케이션, 모델기반개발, UML 프로파일, 플랫폼독립모델, 플랫폼종속모델

#### Abstract

MDD is suitable to satisfy constraints of development of mobile applications. So, there are various studies about applying MDD to the developments of mobile application but systematic development methods about UML profile for PIM/PSM of mobile applications and model transformation techniques are needed. This paper suggests that a development technique about UML profile for PIM/PSM of mobile applications and a model transformation techniques with the

• 제1저자 : 최윤석

• 투고일 : 2012. 04. 26, 심사일 : 2012. 05. 22, 게재확정일 : 2012. 06. 08.

\* 동덕여자대학교 컴퓨터학과(Dept. of Computer Science, Dongduk Women's University)

※ 이 논문은 2011년도 동덕여자대학교 학술연구비 지원에 의하여 수행된 것임

profiles. We classify stereotypes by the characteristics of mobile application to compose profiles and suggest development guidelines of profiles for PIM/PSM. On the suggested model transformation process, the PIM with the profiles is transformed to the initial PSM with the mapping rules and the PSM is transformed to the refined PSM with templates which reflected detailed information of a mobile platform. We developed a location based service mobile application with the suggested techniques on the Android platform and compared with other techniques to validate usefulness of the suggested techniques.

▶ Keyword : Mobile Application, MDD, UML Profile, PIM, PSM

## I. 서 론

스마트 폰의 급속한 보급과 더불어 다양한 모바일 어플리케이션과 이를 활용한 서비스가 등장하고 있다. 안드로이드, iOS, 윈도우즈 모바일, 바다, 블랙베리 OS, 그리고 심비안 등 스마트폰 멀티 플랫폼의 존재는 모바일 어플리케이션 개발자가 동일한 어플리케이션을 다수의 플랫폼으로 개발하여야 하는 상황을 만들고 있으며, 플랫폼 내에서도 지속적인 업데이트가 이루어지고 있으므로 이를 반영하는 노력이 필요하다. 또한 기존 소프트웨어의 시장 접근과는 달리 안드로이드의 구글 플레이와 애플의 앱스토어와 같이 개발자와 사용자가 직접 대면하는 환경으로 인해 기존 소프트웨어보다 빠른 시장 출시가 필수적이며, 사용자의 피드백에 따른 즉각적인 유지보수를 제공할 필요가 있다. 이처럼 다양한 목표 플랫폼을 대상으로 빠른 시장 출시 및 지속적인 유지보수를 제공해야 하는 모바일 어플리케이션을 개발하기 위해서는 분석 및 설계 단계에서부터 산출물의 생성 자동화 및 재사용을 통한 생산성 향상이 필수적이다.

OMG의 MDA[1] 기반 개발방법인 MDD(Model-Driven Development)는 PIM(Platform Independent Model)과 PSM(Platform Specific Model)을 활용한 모델 변환 및 코드 자동생성을 통해 효율적인 소프트웨어 개발을 지원하는 개발 방법이다[2]. MDD는 모바일 어플리케이션 개발의 특성을 잘 반영할 수 있으므로 이를 적용하기 위한 다양한 연구가 진행되고 있다.

모바일 어플리케이션 개발에 MDD를 적용하기 위해서는 다음 사항에 대한 고려가 필요하다. 첫째, UML기반의 PIM과 PSM 설계를 수행하기 위해 UML 확장 메커니즘인 UML 프로파일은 모바일 어플리케이션의 특성에 적합하게 개발하여 적용하여야 한다. PIM과 PSM에 적용할 UML 프로파일을 명확히 정의한 후 적용해야 대응규칙을 결정할 수 있으며 모델 변환의 자동화가 가능해진다[3][4]. 둘째, 모델의 순방

향과 역방향 추적을 지원하여 설계 변경의 영향 분석과 시스템 변경의 설계 반영을 용이하게 수행할 수 있어야 한다[5]. 이를 위해서는 모델 변환 시의 대응관계에 대한 명확한 정의가 필요하며, 자동화 도구를 사용할 수 있는 충분한 정보를 제공할 수 있어야 한다. 셋째, PIM과 PSM의 모델 변환 규칙 작성이 용이하여야 한다. 어플리케이션의 변경 또는 어플리케이션과 관련한 기술에 변경이 있을 경우 이를 모델 변환 규칙에 반영하여야 한다[6]. 모바일 플랫폼은 지속적인 업데이트가 이루어지고 있으며, 개발 시 기존 플랫폼이 아닌 새로운 모바일 플랫폼을 고려하여야 할 수도 있다. 따라서 모델과 모델 변환에 대한 노력을 최소화 할 수 있도록 모델 및 모델 변환 규칙 작성이 용이하여야 하고, 더불어 기존 개발 방법 및 도구를 활용할 수 있어야 한다.

본 논문에서는 이와 같은 고려사항을 반영하여 모바일 어플리케이션의 PIM/PSM 설계에 적용할 UML 프로파일의 구성 기법과 이에 기반한 모델 변환 기법을 제안한다. UML 프로파일의 구성을 위해 모바일 어플리케이션의 특징을 반영한 스테레오타입의 유형 분류 및 작성 지침을 제안하며, 각 모델에 적용한 스테레오타입 사이의 대응관계를 기반으로 모델 변환 절차를 제시하고 이에 필요한 프로파일의 대응규칙, 그리고 변환을 위한 템플릿 구성 방법을 제안한다. 대응규칙 및 템플릿은 접근의 용이성을 위해 XML 기반으로 구성하며 모델 정보는 표준 모델 교환 형식인 XMI를 활용한다.

논문의 구성은 다음과 같다. 2장에서는 연구 배경 및 관련 연구를 살펴보고, 3장에서는 PIM과 PSM을 위한 UML 프로파일 구성 기법을 기술한다. 4장에서는 프로파일을 활용한 모델 변환 기법을 기술하며, 5장에서는 사례 연구 및 분석을 통해 제안한 기법의 유용성을 확인한다. 6장에서는 결론 및 향후 연구 계획을 설명한다.

## II. 관련 연구

## 1. 연구 배경

OMG의 MDA를 기반으로 한 개발방법인 MDD는 플랫폼에 의존적이지 않은 모델인 PIM을 명세하고, PIM을 바탕으로 시스템에 의존적인 모델인 PSM을 명세한 후 특정 플랫폼에서 운영할 수 있는 시스템으로 개발한다. MDD는 반복적으로 사용할 수 있는 설계 모델을 재사용하고 모델 변환 및 코드의 생성을 자동화하여 소프트웨어 생산성을 높일 수 있다[5]. 또한 모델의 순방향과 역방향으로의 추적을 지원하므로 개발 시 설계의 변경에 따른 영향 분석과 시스템 개발 후 유지보수에 따른 시스템 변경의 설계 반영이 용이한 장점이 있다. 따라서 복수의 모바일 플랫폼 대상 어플리케이션 개발에 적용하여 생산성을 높일 수 있다[7].

MDD 적용 시 작성한 모델을 변환하기 위한 방법으로는 모델에 기호 표시, 메타모델 활용, 모델 타입 및 타입 변환 규칙 정의, 변환 패턴 정의, 추가 정보 활용 등이 있다[1]. 기호 표시 방법으로 분류할 수 있는 UML 프로파일 활용 방법은 개발 어플리케이션의 플랫폼을 반영한 UML 프로파일을 정의하여 모델에 적용한다. 이 방법은 모델 구성요소의 의미를 명확하게 정의할 수 있으며, 여러 가지 다른 유형의 대응 정보를 표현할 수 있는 장점이 있다.

UML 프로파일은 UML 모델을 특정 도메인에 적용할 수 있도록 UML의 확장을 지원하는 메커니즘의 패키지이다. UML 프로파일은 UML 요소를 표현하는 스테레오타입, 요소에 추가할 새로운 속성을 정의하는 태그값, 그리고 모델이 갖는 값이 유지하여야 할 상태를 명시하는 제약조건으로 구성한다[8]. UML 2.0부터는 스테레오타입 태그라는 명칭 대신에 속성이라는 명칭을 사용하므로 본 논문에서는 스테레오타입 속성을 사용한다.

## 2. MDD 기반 모바일 어플리케이션 개발 방법

연구 [9]에서는 MDD를 적용하여 시각적인 모델로 어플리케이션을 설계한 후 자동화 도구를 사용하여 모바일 어플리케이션을 개발하는 개념을 제시하고 있다. 이 연구는 모바일 개발 경험 없이도 효과적으로 어플리케이션을 설계하기 위해 UML을 적용하지 않고 시각적인 모델을 사용하여 어플리케이션을 개발하는 것에 초점을 맞추고 있다. 따라서 PIM 또는 PSM을 설계하기 위한 구체적인 방법 및 지침은 제시하고 있지 않으며, 모델 설계 및 변환 절차 등을 고려하고 있지 않다.

연구 [10]은 모델 변환 언어인 ATL을 활용하여 고수준의 모델을 윈도우즈 모바일 어플리케이션을 위한 상세 설계 모델로 변환하는 방법을 제안하였다. 플랫폼 독립적인 특성들과

종속적인 특성을 분류한 후 스테레오타입을 적용하여 목표 독립 모델인 TIM을 생성하고 ATL로 작성한 변환규칙을 활용하여 윈도우 모바일 플랫폼 상의 목표 종속 모델인 TSM을 자동으로 생성하였다. 이 연구는 목표 독립 모델이 이종의 다른 플랫폼으로의 변환될 수 있는 가능성을 제시하였으나 TIM 설계 시 설계에 활용하는 스테레오타입을 정의하고 적용하기 위한 구체적인 방법을 제시하고 있지 않다. 또한 변환 규칙으로 활용하는 ATL의 경우 모델 변환의 규칙과 모델 변환의 정보를 분리하여 작성할 수 없으므로 플랫폼의 변경에 따른 모델 변환 정보의 반영과 모델 사이의 순/역방향 추적이 필요한 정보 제공이 용이하지 않다.

연구 [11]에서는 모바일 플랫폼 종류에 관계없이 일관성 있는 UX를 제공하고 복수의 개발 프로세스 진행에 따른 어려움을 해결하기 위해 MDD를 적용한 멀티 플랫폼 상의 모바일 어플리케이션 사용자 인터페이스 자동 생성 기법을 제안하였다. 사용자 인터페이스 개발을 위한 MDD 기반의 모델 변환 과정을 제시하였고, 도구를 활용하여 모델로부터 사용자 인터페이스의 소스코드를 자동 생성하는 방법을 제시하였다. 이 연구는 단일 모델로부터 복수의 모바일 어플리케이션 사용자 인터페이스를 자동 생성하는 방법을 제시하고 있으나 모바일 어플리케이션 아키텍처 중 사용자 인터페이스 생성에 한정되어 있으므로 아키텍처 전체에 MDD를 적용하는 방법이 별도로 요구된다. 또한 PIM과 PSM 모델링을 수행하기 위한 구체적인 방법을 제시하고 있지 않으며, 변환 시에는 표준 모델 교환 형식인 XMI를 XOZ라는 별도의 형식으로 변환하는 추가적인 노력이 필요하다.

이와 같이 모바일 어플리케이션 개발에 MDD를 적용한 기존의 연구들에서는 모델 변환에 필요한 규칙을 구성하는 방법과 이에 기반한 자동화 도구를 제시하고 있으나 프로파일 등을 사용하여 변환 대상을 정의하는 방법이 명확하지 않으며, 모델 변환의 순/역방향 추적을 위한 정보 제공을 고려하고 있지 않고 있다. 또, 모델 변환 규칙과 변환에 사용하는 정보가 결합되어 있는 형태이므로 플랫폼 변경에 따른 모델 정보의 변경 적용이 용이하지 않은 문제가 있다.

## III. 모바일 어플리케이션을 위한 UML 프로파일 구성 기법

본 장에서는 모바일 어플리케이션의 PIM과 PSM에 적용할 UML 프로파일의 구성 기법을 제안한다. 모바일 어플리케이션을 분석하여 설계에 필요한 스테레오타입의 유형을 분류

하며, 분류에 따라 스테레오타입을 지정하고 프로파일을 구성하기 위한 지침을 기술한다. 제안 기법을 적용하여 개발한 프로파일은 모델링 도구를 사용하여 구성하므로 MOF[13]를 준수하며, XMI 형식을 사용하여 기존 UML 모델링 도구에서 활용이 가능하다.

### 1. 모바일 어플리케이션 PIM의 UML 프로파일 구성 기법

본 절에서는 PIM 설계에 적용할 UML 프로파일 구성 기법을 제안한다. PIM은 특정 플랫폼에 독립적인 상위 수준의 추상화 설계이며 클래스 다이어그램을 중심으로 소프트웨어 아키텍처를 표현한다. PIM은 설계 시 플랫폼을 고려하지 않고 설계를 수행하므로 추상화 수준에 있어 설계자에 따라 모델링 상세화 정도가 달라질 수 있다. 추상화 수준이 일정하지 않을 경우 PIM에서 PSM으로 변환하는 작업을 자동화하기가 어려워진다. 따라서 PIM을 PSM의 구조 및 구성요소에 대응시킬 수 있도록 일정한 추상화 수준의 구조 및 구성요소로 표현할 수 있어야 한다.

본 논문에서는 이러한 조건을 만족하기 위하여 모바일 어플리케이션 PIM의 UML 프로파일 구성 시 스테레오타입을 아키텍처, 이벤트, 그리고 사용자 정의의 세 가지 유형으로 분류하고, 이를 기준으로 PIM에 적용할 스테레오타입을 지정한다. 아키텍처 유형의 스테레오타입은 모바일 어플리케이션의 구조를 표현하며, 이벤트 유형의 스테레오타입은 사용자 인터페이스 이벤트 처리를 표현한다. 이밖에 도메인 객체 등은 사용자 유형의 스테레오타입으로 표현한다.

#### 1.1. PIM 프로파일의 스테레오타입 유형 분류

아키텍처 유형의 스테레오타입은 모바일 어플리케이션 아키텍처의 설계 구성요소에 적용할 스테레오타입으로서 모델의 클래스를 표시 대상으로 지정한다. 아키텍처 유형의 스테레오타입을 정의하기 위해서는 모바일 플랫폼 및 플랫폼 상에 운용하는 어플리케이션의 아키텍처 분석이 필요하다. 대표적인 스마트폰 플랫폼인 iOS[14]의 경우 MVC, Delegation, 그리고 Target-action 등의 패턴을 아키텍처에 적용하도록 강제하고 있으므로 패턴들의 구성요소를 아키텍처 스테레오타입으로 지정할 수 있으며, 이 외에 아키텍처 관련 설계 패턴의 구성요소를 아키텍처 스테레오타입으로 지정할 수 있다.

이벤트 유형 스테레오타입은 PIM에서 설계할 모바일 어플리케이션의 사용자 입출력 관련 이벤트를 대상으로 하며, PIM의 클래스 오퍼레이션을 표시 대상으로 지정한다. PIM 상에 표현할 수 있는 사용자 입출력 관련 이벤트 중 PSM으

로 변환 시 PSM의 구조에 영향을 줄 수 있는 이벤트들이 존재할 수 있다. 스마트폰에서 기본적으로 지원하는 사용자 입력 이벤트 중 하나인 탭 이벤트나 롱 탭 이벤트의 경우 이벤트를 발생시키는 클래스와 이벤트를 처리하는 클래스로 재구성할 수 있다. 예를 들어 안드로이드 기반 모바일 어플리케이션에서 지도상의 특정 지점을 롱 탭하는 이벤트를 처리하기 위해서는 탭을 표현하는 View 클래스에 Overlay 클래스를 추가해야한다[15]. 이와 같이 PSM의 구조에 영향을 줄 수 있는 모바일 어플리케이션의 이벤트 요소를 식별한 후 이를 스테레오타입으로 지정한다. 이벤트 유형 스테레오타입으로 지정할 수 있는 모바일 어플리케이션 이벤트는 표 1과 같다.

표 1. 모바일 어플리케이션의 이벤트  
Table 1. Events of Mobile Application

이벤트 종류	설명
키보드 입력	하드웨어/가상 키보드를 통한 입력
하드웨어 입출력 요소 사용	스마트폰을 구성하는 필수 입출력 요소 (e.g. 전원버튼, 볼륨버튼, 스피커, 마이크 등)
탭(터치)	화면을 일정시간 미만으로 터치
롱 탭(롱터치)	화면을 일정시간 이상으로 터치
멀티 터치	두 군데 이상의 화면을 동시에 터치
스크롤	화면을 터치한 상태로 움직임

키보드 입력과 하드웨어 키 입력 등과 같이 구현 환경에 의존적인 이벤트를 PSM에서 정의하는 것이 아닌 PIM에서부터 지정하는 이유는 스마트폰의 경우 하드웨어 입출력 요소와 키보드가 필수적이며, PIM 모델링 시 전화걸기와 같이 반드시 키보드를 활용하는 이벤트가 존재할 수 있으므로 이를 PIM에서부터 반영한다. 이 외에 스마트폰 디바이스의 필수적인 이벤트 처리를 위해 필요한 요소가 있을 경우 PIM 스테레오타입으로 표현한다.

아키텍처 유형과 이벤트 유형의 스테레오타입을 제외한 기타 PIM 구성요소를 사용자 정의 유형 스테레오타입으로 지정한다. 사용자 정의 유형은 모바일 어플리케이션의 도메인 객체 등이 될 수 있으며, 이 밖에 모바일 어플리케이션의 외부 구성요소를 PIM 스테레오타입으로 지정할 수 있다. 모바일 기기에서 제공하는 GPS, 가속도 센서, 자이로 센서, 광센서, 그리고 중력 센서 등과 같은 다양한 센서 및 장치들을 스테레오타입을 지정한다. 센서와 같은 요소는 PIM 프로파일에서 추상화 한 서비스로 표현하며 스마트폰 디바이스에 따라 해당 기능의 탑재 유무가 달라질 수 있으므로 사용자 정의 유형으로 분류한다.

#### 1.2. PIM 프로파일의 스테레오타입 구성

스테레오타입의 유형을 분석하여 분류를 수행한 후 선정된 스테레오타입을 대상으로 PIM의 UML 프로파일을 구성한다. 프로파일 구성 시 UML 모델링 도구를 사용하여 모델링한 후 XMI 형식으로 변환한다. XMI 형식으로 저장한 프로파일은 도구를 사용한 PIM/PSM 변환 시 입력정보로 활용할 수 있다. 표 2는 PIM의 UML 프로파일의 예이며, 그림 1은 모델링 도구를 사용하여 설계한 PIM 프로파일 모델의 예이다.

표 2. PIM 프로파일의 예  
Table 2. Example of PIM Profile

Type	Stereotype	Target	Description
architecture	<<model>>	class	MVC pattern
architecture	<<view>>	class	MVC pattern
architecture	<<controller>>	class	MVC pattern
event	<<longTab>>	operation	user event
user	<<device>>	class	attribute (deviceType)

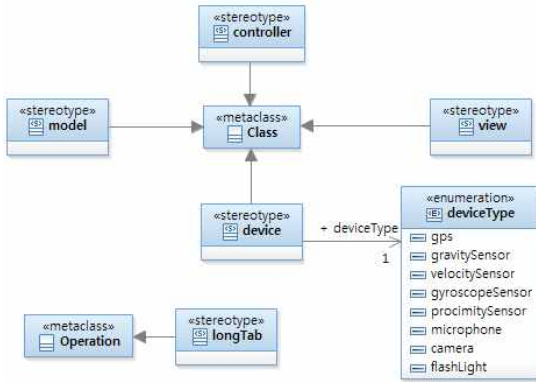


그림 1. PIM 프로파일 모델의 예  
Fig. 1. Example of PIM Profile Model

PIM용 UML 프로파일의 스테레오타입 명은 PSM의 플랫폼 요소와 구분하기 위하여 소문자로 명명한다. UML 프로파일 구성의 Type 항목은 PIM에 적용할 스테레오타입의 유형으로서, architecture, event 그리고 user의 세 값 중 하나를 기록한다. Stereotype 항목은 유형 별로 정의한 스테레오타입의 명칭을 기록한다. Target은 스테레오타입을 적용하는 PIM의 구성요소 유형으로서 모델에 나타나는 클래스, 속성, 그리고 오퍼레이션을 표현한다.

다음은 PIM 프로파일의 스테레오타입 구성 지침이다.

- 지침 1. architecture 유형의 스테레오타입은 아키텍처에서 구성요소의 역할을 표현하는 명칭을 사용한다.
- 지침 2. architecture 유형의 스테레오타입 중 설계 패

턴의 구성요소를 지정하는 스테레오타입은 패턴에서의 명칭과 동일하게 명명한다.

- 지침 3. event 유형의 스테레오타입은 스마트폰 디바이스들의 이벤트 중 공통요소만을 대상으로 지정한다.
- 지침 4. architecture 유형과 event 유형을 제외한 나머지 대상을 user 유형의 스테레오타입으로 지정한다.
- 지침 5. 동일한 유형의 스테레오타입일 경우 각각 스테레오타입을 지정하지 않고 스테레오타입의 속성으로 표시한다.

지침 3의 경우 모바일 플랫폼 별 스마트폰 디바이스를 분석하여 디바이스의 하드웨어 입출력 장치 및 사용자 인터페이스 이벤트 중 공통적인 요소만을 추출하여 스테레오타입으로 지정하는 것을 의미한다. 지침 5를 적용하여 그림 1의 <<device>>와 같이 하드웨어 구성요소를 각각의 스테레오타입으로 지정하지 않고 속성을 사용하여 표현한다.

#### IV. 모바일 어플리케이션 PSM의 UML 프로파일 구성 기법

본 절에서는 PSM 설계에 적용할 UML 프로파일 구성 기법을 제안한다. PSM은 어플리케이션을 운용할 플랫폼의 특성을 반영하여 설계한 모델로서 어플리케이션 아키텍처에 플랫폼 종속적인 구성요소를 포함한다. 따라서 PSM의 UML 프로파일 구성 시 모바일 플랫폼 특성과 해당 플랫폼의 모바일 어플리케이션 특성을 반영하여야 한다.

모바일 어플리케이션은 소프트웨어 구성요소 외에 플랫폼의 서비스 지원을 통하여 스마트폰 디바이스의 하드웨어 구성요소를 활용한 서비스를 사용할 수 있다. 그러므로 모바일 어플리케이션의 PSM 설계 시 어플리케이션의 소프트웨어 구성요소와 더불어 하드웨어 구성요소를 설계에 반영한다.

##### 1. PSM 프로파일의 스테레오타입 유형 분류

본 논문에서는 PSM에 적용할 프로파일의 스테레오타입을 하드웨어, 소프트웨어 그리고 사용자가 정의한 유형으로 분류하여 구성하였다. 표 3은 PSM 프로파일의 스테레오타입 유형이다.

하드웨어 유형의 스테레오타입은 모바일 플랫폼에서 지원하는 디바이스 구성요소를 대상으로 적용한다. 디바이스 구성요소는 각종 센서류와 디바이스 입출력 장치 등이며 모바일 플랫폼은 디바이스 구성요소를 사용하기 위해 이를 추상화한

서비스 구성요소를 제공한다. 그러므로 하드웨어 유형의 스테레오타입은 디바이스 구성요소가 제공하는 플랫폼의 서비스를 대상으로 지정한다. 디바이스 구성요소는 스마트폰의 디바이스 사양과 모바일 플랫폼의 변경에 영향을 받을 수 있으므로 PSM 프로파일 구성 시에는 스마트폰 디바이스 및 모바일 플랫폼에 대한 분석을 선행하여야 한다. 소프트웨어 유형의 스테레오타입은 플랫폼 상에서 운영하는 어플리케이션의 아키텍처 구성요소로서 아키텍처 기본 요소, 사용자 인터페이스, 이미지, 사운드 및 비디오와 같은 리소스, 네트워크, 그리고 데이터베이스 등이 될 수 있으며 각 구성요소 별로 지정한다. 이외에 모바일 플랫폼과 관련된 요소 이외에 사용자가 직접 정의한 모델 구성요소를 사용자 스테레오타입으로 지정한다.

표 3. PSM 스테레오타입의 유형  
Table 3. Category of PSM Stereotype

유형	세부 유형	지정 대상
HW	I/O	- 사용자 입출력 관련 디바이스 요소 - 스피커, 마이크, 하드웨어 버튼, 하드웨어 키보드, 카메라, 플래시, 진동 모터 등
	Sensor	- 각종 센서 요소 - 중력센서, 가속도센서, 자이로센서, 광센서, GPS 등
	Network	- 통신관련 구성요소 - 블루투스, Wi-Fi, 3G/4G, 적외선 통신, NFC 등
SW	Architecture	- 모바일 플랫폼에서 지정한 어플리케이션 구조의 구성요소
	UI	- 사용자 인터페이스를 구성하기 위한 모바일 플랫폼의 구성요소 - 사용자 인터페이스 컴포넌트, 컨테이너 클래스 등
	Resource	- 모바일 어플리케이션에서 활용하는 각종 자원 관련 구성요소 - XML, 이미지, 사운드, 비디오 등
	Storage	- 모바일 어플리케이션의 데이터 저장을 지원하는 구성요소 - 파일, 데이터베이스 등
user	user	사용자가 직접 정의한 모바일 플랫폼에 독립적인 소프트웨어 구성요소

## 2. PSM 프로파일의 스테레오타입 구성

PIM 프로파일에서 분류한 하드웨어, 소프트웨어, 그리고 사용자 정의 유형 정보와 모바일 플랫폼의 정보를 분석하여 PSM 프로파일을 구성한다. 표 4는 안드로이드 플랫폼 대상의 PSM 프로파일을 구성한 예이다.

표 4. 안드로이드 플랫폼의 PSM 프로파일의 예  
Table 4. Example of PSM Profile for Android Platform

Type	Stereotype	Target	Description
HW	<<LocationService>>	class	enumeration (ServiceType)
SW	<<Activity>>	class	enumeration (ActivityType, MandatoryOperation)
SW	<<SQLiteOpenHelper>>	class	data store
SW	<<View>>	class	enumeration (ViewType)
SW	<<LongTab>>	operation	user event
SW	<<Click>>	operation	user event
user	<<Dtc>>	class	user defined

다음은 PSM 프로파일 스테레오타입 구성 시의 지침을 나타낸다.

- 지침 1. 동일한 서비스를 제공하는 하드웨어 유형의 구성요소는 대표 서비스명을 결정하여 스테레오타입 명으로 사용하며, 세부적인 분류는 스테레오타입 속성으로 표시한다.
- 지침 2. 소프트웨어 유형 중 상속관계에 있는 구성요소들이 나타날 경우 필요에 따라 하위클래스를 스테레오타입 속성으로 표시한다.
- 지침 3. 구성요소의 구현 방법이 하나 이상인 경우 구현방법을 스테레오타입 속성으로 표시한다.
- 지침 4. 구성요소 추가 시 필수적인 속성 및 오퍼레이션이 존재할 경우 스테레오타입 속성으로 표시한다.

지침 1은 하드웨어 구성요소의 서비스를 스테레오타입으로 지정할 경우에 적용한다. 예를 들어 위치 정보 서비스의 경우 스마트폰 디바이스의 GPS와 네트워크 정보를 사용하여 현위치를 파악할 수 있다. 이 경우 위치 정보서비스를 대표할 수 있는 명칭인 <<LocationService>>를 스테레오타입 명으로 설정하며 GPS 또는 Network를 속성 값으로 지정한다.

지침 2는 소프트웨어 구성요소 중 상속관계에 있는 클래스를 스테레오타입으로 지정하기 위한 지침이다. 플랫폼에서 활용한 모든 클래스를 스테레오타입으로 지정할 경우 모델의 가독성이 저하될 수 있다. 이 경우 상속관계에 있는 클래스 중 슈퍼클래스를 스테레오타입 명으로 지정한다. 예를 들어 사용자 인터페이스 컴포넌트중 하나인 버튼 클래스와 같이 일반/라디오/체크/이미지 버튼 등 다양한 유형으로 나타날 경우 모바일 플랫폼의 클래스 구조에서 최상위 클래스인 버튼을 지정하는 스테레오타입 <<Button>>을 지정하고 실제 적용할 하

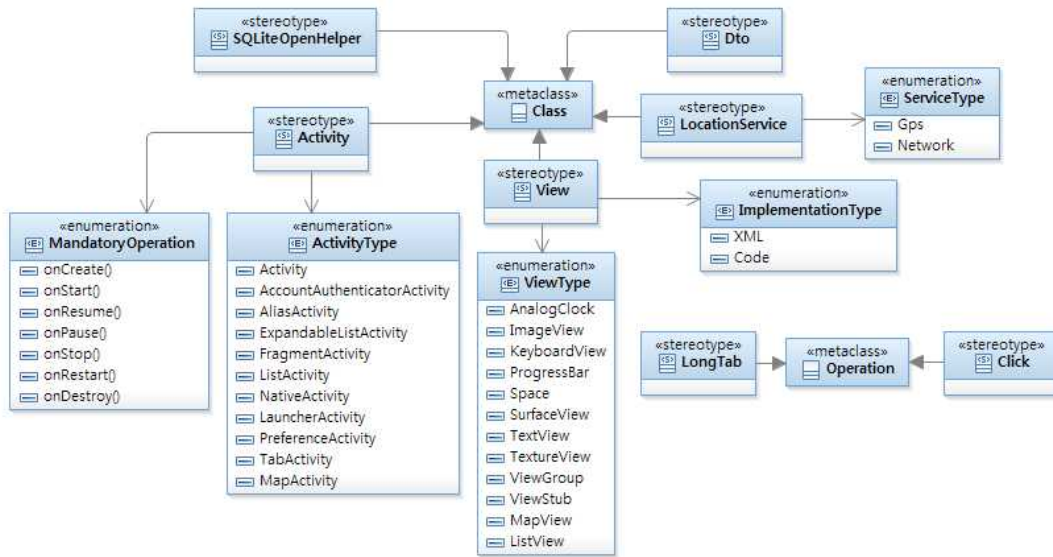


그림 2. 안드로이드 플랫폼 PSM 프로파일 모델의 예  
Fig. 2. Example of PSM Profile Model for Android Platform

위클래스는 스테레오타입의 속성 값으로 설정할 수 있다.

지침 3과 4는 PSM을 사용하여 코드 자동 생성 시 필요한 부가정보 기술을 위한 지침이다. 안드로이드 플랫폼의 경우 사용자 인터페이스는 XML 파일로 구성하는 방법과 코드 상에서 직접 생성하는 방법을 사용할 수 있다. 코드를 자동 생성하기 위해서는 생성 유형을 PSM에서 결정할 필요가 있다. 이 경우 지침 3에 따라 스테레오타입에 구현 유형 속성을 추가하여 결정 유형을 기록한다. 특정 스테레오타입으로 표현한 모바일 플랫폼의 구성요소가 구현 시 반드시 포함하여야 할 속성이나 오퍼레이션이 있을 경우 이를 표현하는 스테레오타입 속성을 추가하여 기록한다. iOS의 경우 UIViewController 클래스를 상속하는 클래스는 메모리 관리를 위해 라이프 사이클 관련 메소드인 viewDidLoad()를 포함하여야 한다. <<UIViewController>> 스테레오타입을 지정하는 경우 속성을 추가하여 viewDidLoad()를 기록한다.

PSM 프로파일은 PIM 프로파일과 동일하게 UML 모델링 도구를 활용하여 모델링 한 후 XML 형식으로 저장한다. 그림 2는 안드로이드 플랫폼 대상의 PSM 프로파일 일부를 모델링 도구로 설계한 모델의 예이다.

본 장에서는 PIM 프로파일을 적용하여 설계한 모델을 PSM으로 변환하는 모델 변환 기법을 제시한다. 모델 변환을 수행하기 위한 절차와 PIM 프로파일과 PSM 프로파일의 대응규칙 구성 방법을 제시한다. 또, 초기 PSM에서 정제한 PSM으로 변환하기 위한 PSM 템플릿을 제안하며, 모델의 자동 변환을 수행하는 도구 사용을 제시한다.

### 1. 모바일 어플리케이션의 모델 변환 절차

본 논문에서는 기존의 MDD 적용 절차를 보완하여 초기 PIM에서 스테레오타입을 기재하여 PIM을 구성한 후 PSM을 생성하며, 생성한 PSM을 템플릿을 적용하여 구체화한다. 그림 3은 프로파일을 적용한 PIM의 PSM 변환 절차이다.

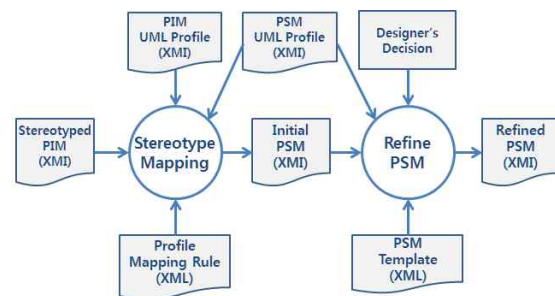


그림 3. 프로파일을 적용한 PIM/PSM 변환 절차  
Fig. 3. Transformation Process of PIM/PSM with Profiles

## V. 모바일 어플리케이션의 PIM/PSM 변환 기법

UML 프로파일을 구성 기법을 사용하여 구성된 프로파일을 토대로 초기 PIM에 스테레오타입을 지정한다. PIM은 UML 모델링 도구를 활용하여 작성하며, 작성한 PIM은 변환 도구의 입력으로 활용하기 위하여 XMI 형식으로 변환한다. XMI 형식의 PIM을 입력으로 PIM/PSM 프로파일 대응 규칙을 적용하여 PIM에 나타난 스테레오타입을 PSM 프로파일의 스테레오타입으로 변환하여 초기 PSM을 생성한다. 초기 PSM은 PIM과 동일한 모델 구조를 가지며, PIM 스테레오타입을 이에 대응하는 PSM 스테레오타입으로 변환한 상태이다. 초기 PSM의 스테레오타입에 속성이 존재할 경우 모바일 어플리케이션의 설계 목적에 따라 속성 값을 설정한다. 스테레오타입의 속성 값을 설정한 후 PSM 변환 템플릿을 적용하여 PSM 정제를 수행한다. 템플릿 적용 시 초기 PSM 구조의 변경이 발생할 수 있다. 템플릿을 적용하여 정제를 수행한 PSM을 토대로 설계자는 추가 모델링을 수행할 수 있으며, 모델링 수행 후 XMI 형식을 입력으로 하는 코드 자동 생성 도구에 적용하여 코드를 자동 생성할 수 있다.

2. 프로파일 대응규칙 구성

스테레오타입을 지정한 PIM을 PSM으로 변환하기 위해 프로파일 사이의 대응규칙을 구성한다. 대응규칙은 정보 변경 및 활용성을 고려하여 XML로 구성한다.

PIM 프로파일의 스테레오타입은 PSM 프로파일의 스테레오타입과 1:N의 대응관계를 구성할 수 있다. 대응 관계의 스테레오타입은 스테레오타입 자체의 변경과 스테레오타입 대상 정보의 변경을 수행한다. 스테레오타입 자체의 변경을 통해 PIM 스테레오타입을 PSM 스테레오타입으로 대체한다. 복수의 PSM 스테레오타입과 대응할 경우 순차적으로 스테레오타입 정보를 추가한다. 이후 스테레오타입 대상 정보의 변경을 수행하여 대상 클래스, 속성, 오퍼레이션 명을 모바일 플랫폼의 이름 명명 규약에 적합하게 수정한다. 그림 4는 PIM/PSM 프로파일 대응규칙의 XML 구조이다.

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping_rules>
  <rule name="rule_name" target_type="target_type">
    <source name="pim_stereotype_name">
      <attribute name="attribute_name" value="attribute_value" />
    </source>
    <target>
      <name>pim_stereotype_name</name>
      <suffix_name>suffix_name</suffix_name>
      <target_name>replaced_name</target_name>
      <refine>check_refine</refine>
      <attribute name="attribute_name" value="attribute_value" />
    </target>
  </rule>
</mapping_rules>
```

그림 4. PIM/PSM 프로파일 대응규칙 XML 구조  
Fig. 4. XML Structure of Mapping Rule for PIM/PSM Profile

프로파일 대응규칙은 하나 이상의 규칙으로 구성한다. 하나의 규칙은 규칙명과 적용하는 대상을 클래스, 속성 또는 오퍼레이션인지를 기록하는 타입을 가지며, 대응 관계의 PIM 스테레오타입을 의미하는 source와 PSM 스테레오타입을 의미하는 target으로 구성한다. 규칙의 name은 PIM 스테레오타입 명과 PSM 스테레오타입 명을 결합하여 구성한다. 예를 들어 <<controller>>를 <<Activity>>에 대응하는 규칙의 명칭은 controller\_Activity로 명명한다. PIM의 스테레오타입을 의미하는 source에는 스테레오타입 명을 name에 기록하고 프로파일에 스테레오타입 속성 및 값이 존재할 경우 attribute를 사용하여 속성명과 값을 기록한다. attribute 값에 따라 대응하는 PSM 스테레오타입이 달라질 수 있다. target은 PSM 프로파일에서 source에 대응하는 스테레오타입을 의미하며 name에 스테레오타입 명을 기록한다. suffix\_name과 target\_name은 스테레오타입을 지정한 대상 클래스, 속성, 또는 오퍼레이션을 모바일 플랫폼의 명명 규약에 맞추어 변경 시 사용한다. 본래의 대상 명칭에 접미어를 붙이는 형식은 suffix\_name 정보를 사용하며, 명칭 대체가 필요할 경우 target\_name을 사용한다. refine는 PSM 변환 후 스테레오타입을 지정한 대상에 대하여 설계자의 추가적인 정제가 필요한지의 여부를 기록한다. attribute는 PSM 스테레오타입에 지정한 속성을 표시한다. attribute는 PSM 정제 시 활용한다.

3. 모바일 어플리케이션 PSM 템플릿 구성

본 절에서는 모바일 어플리케이션의 세부 구조를 반영하여 PSM 모델을 정제하기 위해 사용할 템플릿의 구성을 제시한다. 모바일 플랫폼은 새로운 플랫폼의 등장과 기존 플랫폼의 개선으로 인해 모델 변경이 빈번하게 발생할 수 있다. 이때 모델 변환을 위한 규칙 구조와 정보가 통합되어 있을 경우 변경한 정보를 모델 변환에 적용하기 어려울 수 있다. 이러한 문제를 해결하기 위하여 본 논문에서는 변경 규칙과 규칙에 적용하는 정보를 템플릿으로 분리하여 구성한다. 변경 정보를 구성하는 템플릿의 사용을 통해 모델 변환 정보의 변경을 용이하게 수행할 수 있으며, 모델 변환의 순방향/역방향 추적에 필요한 정보를 용이하게 획득할 수 있다.

3.1. PSM 템플릿 유형 분류

변경을 수행하는 대상에 따라 템플릿의 유형을 분류하여 구성한다. 변경 수행 대상은 스테레오타입과 스테레오타입의 속성값으로 구분한다. 이를 위해 스테레오타입 대응규칙을 적용하여 초기 PSM을 구성한 후 설계자는 스테레오타입 속성 값을 기록한다. 속성 값을 기록한 초기 PSM에 PSM 템플릿

을 적용하여 PSM을 정제한다. 표 5는 PSM 프로파일의 스테레오타입에 적용하는 PSM 템플릿의 유형이다.

표 5. PSM 템플릿 유형  
Table 5. Category of PSM Templates

대상	유형	설명
Class	중재 클래스 (mediate)	본래 클래스와 연관 관계 클래스 사이에서 중재 역할을 하는 클래스
	보조 클래스 추가 (assist)	본래의 클래스에 서비스를 제공하는 클래스
	연관 클래스 추가 (associate)	클래스 사이의 연관 관계를 정의하는 클래스
Operation	보조 오퍼레이션 추가 (assist_operation)	본래의 오퍼레이션 기능을 보조하는 추가 오퍼레이션
	보조 클래스 추가 (assist_class)	오퍼레이션을 소유하는 보조 클래스
Attribute	보조 속성 추가 (assist_attribute)	본래의 속성을 보조하는 속성
	속성 오퍼레이션 추가 (assist_operation)	속성을 다루기 위해 필요한 오퍼레이션

스테레오타입의 대상이 클래스인 경우 모바일 플랫폼의 특성을 반영하여 중재 클래스(mediate), 보조 클래스(assist), 그리고 연관을 표현하는 클래스(associate)를 추가할 수 있다. 예를 들어 안드로이드 기반 PSM의 <<View>> 스테레오타입으로 지정한 클래스는 Layout 관련 클래스가 필요하다. 그러므로 초기 PSM에서 <<View>> 스테레오타입을 지정한 클래스를 대상으로 Layout 클래스를 추가하는 템플릿을 구성할 수 있다. 클래스 대상의 경우 오퍼레이션과 속성의 유형도 적용할 수 있다. 스테레오타입의 대상이 오퍼레이션인 경우 템플릿에 의해 오퍼레이션을 추가하거나 오퍼레이션을 담당하는 클래스를 추가하는 템플릿을 구성할 수 있다. 안드로이드 플랫폼의 사용자 이벤트 관련 오퍼레이션은 이벤트를 처리 시 이벤트 핸들러 클래스를 구성하여 사용한다. 따라서 초기 PSM의 사용자 이벤트 관련 스테레오타입으로 지정한 오퍼레이션은 이벤트 핸들러 클래스를 추가하는 템플릿을 구성하여 적용할 수 있다. 스테레오타입 대상이 속성인 경우 보조 속성을 추가하거나 속성 관련 오퍼레이션을 추가하는 템플릿을 구성할 수 있다.

3.2. PSM 템플릿의 구조

PSM 템플릿은 PSM 스테레오타입 지정 대상을 모바일 플랫폼의 특성을 반영하여 보다 구체적으로 변환하기 위한 정

보로 구성한다. 그림 5는 PSM 템플릿의 구조이다.

template은 name과 표 5의 템플릿 유형을 기록하는 type 속성을 갖는다. template은 적용 대상 스테레오타입을 표현하는 target과 template의 유형에 따라 필요한 class, operation, attribute를 포함하며, 추가한 정보를 지정하는 stereotype 정보를 포함한다. 적용 대상 스테레오타입을 표현하는 target은 name과 attribute의 조합으로 결정하며, 복수의 템플릿을 적용할 경우의 우선순위를 priority에 기록한다. 클래스를 추가하는 경우 스테레오타입을 지정한 클래스와의 연관 관계를 source\_association에 기록하며, 스테레오타입 지정 클래스가 다른 클래스와 갖고 있던 연관 관계를 새로 추가한 클래스와의 target\_association으로 대체한다. 연관 관계에는 연관의 다중성과 참여역할을 기술한다. 오퍼레이션 추가의 경우 명칭과 반환 타입, 그리고 매개변수 타입을 기록하며, 속성 추가 시에는 추가할 속성의 명칭과 타입을 기록한다. 템플릿에 의해 추가한 클래스, 오퍼레이션 및 속성의 스테레오 타입을 기록한다. 템플릿은 템플릿에 의해 추가한 구성요소에 반복 적용하는 것도 가능하다.

```

<?xml version="1.0" encoding="UTF-8"?>
<templates>
  <template name="template_name" type="template_type">
    <target name="target_psm_stereotype">
      <attribute name="attribute_name" value="attribute_value" />
      <priority>priority_value</priority>
    </target>
    <class name="class_name">
      <source_association multiplicity="" role="" />
      <target_association multiplicity="" role="" />
      <operation name="operation_name">
        <return_type>operation_return_type</return_type>
        <parameter type="" name="" />
      </operation>
      <attribute name="" type="" />
      <stereotype name="stereotype_name">
        <attribute name="attribute_name" value="attribute_value" />
      </stereotype>
    </class>
    <operation name="operation_name">
      <return_type>operation_return_type</return_type>
      <parameter type="" name="" />
      <stereotype name="stereotype_name">
        <attribute name="attribute_name" value="attribute_value" />
      </stereotype>
    </operation>
    <attribute name="" type="" />
    <stereotype name="stereotype_name">
      <attribute name="attribute_name" value="attribute_value" />
    </stereotype>
  </template>
</templates>
    
```

그림 5. PSM 템플릿의 구조  
Fig. 5. Structure of PSM Template

4. 프로파일 기반 모델 변환 자동화 도구

제한한 기법을 적용하여 생성한 PIM/PSM UML 프로파일과 대응규칙, 그리고 템플릿을 기반으로 모바일 어플리케이션의 PIM에서 PSM을 자동으로 생성하는 변환 도구를 개발하였다. 그림 6은 자동화 도구와 관련 산출물의 입출력 흐름을 나타낸다.

개발한 도구는 UML 모델링 도구를 사용하여 생성한 XMI 형식의 PIM과 XML 형식의 PIM/PSM 프로파일을 입력으

로 하여 초기 PSM을 XMI 형식으로 생성하는 초기 PSM 생성기(Initial PSM Generator), 그리고 XMI 형식의 PSM과 XML 형식의 PSM 템플릿을 입력으로 받아 정제된 PSM을 XMI 형식으로 생성하는 정제 PSM 생성기(Refined PSM Generator)로 구성된다. 입출력을 표준 모델 교환 형식인 XMI를 사용하므로 기존의 모델링 도구와 연동하여 사용이 가능하며, 플랫폼 변경 등에 따른 모델 정보의 변경 시 해당 정보와 관련한 템플릿의 수정만으로 모델 변환 정보의 변경이 가능하다.

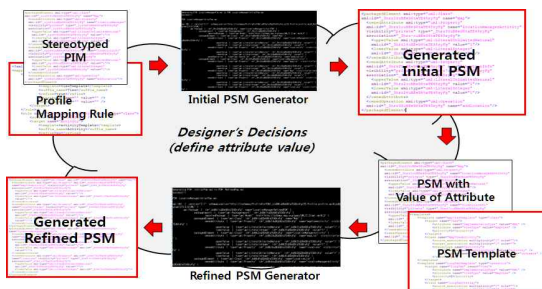


그림 6. 모델 변환 자동화 도구와 산출물의 입출력 흐름  
Fig. 6. Automation Tool for Model Transformation and I/O Flow of Artifacts

## VI. 사례 연구 및 분석

본 장에서는 모바일 어플리케이션 PIM/PSM UML 프로파일 구성 기법과 이를 활용한 모델 변환 기법의 유용성을 확인하기 위하여 위치 기반 서비스 모바일 어플리케이션을 안드로이드 기반으로 개발하여 제안한 기법의 유용성을 확인하며, 기존의 MDD 지원 모바일 어플리케이션 개발 기법과의 비교 분석을 수행한다. 사례 어플리케이션은 사용자의 현재 위치에서 지도의 특정 영역을 기록하여 지도상에 표시하며 목록 형태로 저장한 위치의 위도 및 경도 정보를 표시한다.

### 1. 모바일 어플리케이션의 UML 프로파일 구성

본 논문에서 제안한 UML 프로파일 구성 기법을 적용하여 위치 기반 모바일 어플리케이션을 적용할 PIM/PSM 용 UML 프로파일을 개발하였다. 모바일 어플리케이션의 아키텍처는 MVC 패턴[16]을 적용한 아키텍처 구조에 적합하므로[17], MVC 구성요소별로 스테레오타입을 지정하여 그림 1과 같은 UML 프로파일로 구성하였다. 더불어 Longtab 이벤트와 위치 정보 제공자를 표현하기 위한 스테레오타입을 추가하여 PIM용 UML 프로파일을 구성하였다. PSM 프로파일

의 경우 안드로이드 플랫폼에서 사용하는 어플리케이션 아키텍처를 분석하여 그림 2와 같은 UML 프로파일을 구성하였다.

### 2. 프로파일 대응규칙 및 PSM 템플릿 구성

프로파일을 적용한 PIM과 PSM 모델 사이의 변환을 수행하기 위해 프로파일 대응규칙과 PSM 스테레오타입 별 템플릿을 4장에서 제안한 기법을 적용하여 구성하였다.

안드로이드 어플리케이션 아키텍처를 분석하여 MVC 패턴에 대응하는 요소를 분석한 후 PIM에 적용한 MVC 요소를 표현하는 스테레오타입과의 대응관계로 대응규칙을 결정하였으며, 기타 요소는 각 구성요소의 의미에 적합하게 대응관계를 설정하여 규칙에 반영하였다. 그림 7은 위치 기반 어플리케이션에 적용한 프로파일 대응규칙의 일부를 나타낸다.

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping_rules>
  <rule name="view_View" target_type="class">
    <source name="view" />
    <target name="View">
      <suffix_name>View</suffix_name>
      <refine>true</refine>
      <attribute name="" value="" />
      <attribute name="" value="" />
    </target>
  </rule>
  <rule name="controller_Activity" target_type="class">
    <source name="controller" />
    <target name="Activity">
      <suffix_name>Activity</suffix_name>
      <refine>no</refine>
      <attribute name="activityType" value="Activity" />
      <attribute name="MandatoryOperation" value="onCreate" />
    </target>
  </rule>
  <rule name="model_Ext" target_type="class">

```

그림 7. 위치 기반 어플리케이션의 프로파일 대응규칙  
Fig. 7. Mapping Rule of Profile for Location Manager

대응규칙 구성 후 PSM에 적용한 스테레오타입 적용 대상의 구체화를 위한 템플릿을 구성하였다. 초기 PSM 생성 후 스테레오타입의 속성 정보를 설계자가 결정하였을 때 결정된 속성 정보에 따라 변경하여야 하는 아키텍처 구조와 스테레오타입의 의미에 따라 추가하여야 하는 구조를 고려하여 각 스테레오타입 별로 템플릿을 구성하였다. 그림 8은 PSM 프로파일의 스테레오타입 별 템플릿의 예를 나타낸다.

```
<templates>
  <template name="mapViewTemplate" type="class">
    <target name="View">
      <attribute name="implementationType" value="XML" />
      <attribute name="viewType" value="MapView" />
      <priority>1</priority>
    </target>
    <class name="MapViewActivity">
      <source_association multiplicity="1" role="" />
      <target_association multiplicity="1" role="" />
      <stereotype name="Activity">
        <attribute name="activityType" value="Activity" />
        <attribute name="mandatoryOperation" value="onCreate" />
      </stereotype>
    </class>
  </template>
  <template name="longTabTemplate" type="operation">
    <target name="longTab" owner="View">
      <attribute name="implementationType" value="XML" />
      <attribute name="viewType" value="MapView" />
      <priority>2</priority>
    </target>
  </template>

```

그림 8. 스테레오타입 별 템플릿의 예  
Fig. 8. Example of Template for Stereotype

3. 모바일 어플리케이션 모델 변환 절차의 적용

위치 기록 어플리케이션의 초기 PIM에 PIM 프로파일을 적용하여 그림 9의 스테레오타입을 기록한 PIM으로 재구성하였다. 사례 어플리케이션의 구성요소 역할을 분석하여 PIM UML 프로파일의 스테레오타입 <<model>>, <<view>>, 그리고 <<controller>>로 구성요소를 각각 지정하고, 위치 정보 구성요소는 <<device>>로 지정 후 속성을 'gps'로 지정하였다. 스테레오타입을 지정한 후 모델링 도구에서 XMI 형식으로 변환 저장하였다. 그림 9는 UML PIM 프로파일의 스테레오타입을 적용하여 설계한 위치 기반 어플리케이션의 PIM을 나타낸다.

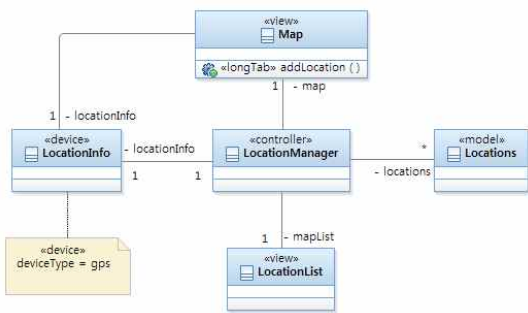


그림 9. 스테레오 타입을 적용한 위치 기록 어플리케이션의 PIM  
Fig. 9. Stereotyped PIM of Location Manager

PIM의 XMI 저장 결과와 그림 7의 프로파일 간 대응규칙을 초기 PSM 생성기에 입력하여 초기 PSM을 생성하였다. 초기 PSM은 PIM과 동일한 구조를 가지며 스테레오타입 부분을 대응규칙에 따라 PSM UML 프로파일의 스테레오타입으로 대체한 형태로 구성되었다. 초기 PSM 생성기는 XMI 형식의 초기 PSM을 생성한다. 그림 10은 초기 PSM 생성기를 통해 자동 생성한 초기 PSM의 XMI를 모델링 도구에서 확인한 결과를 나타낸다.

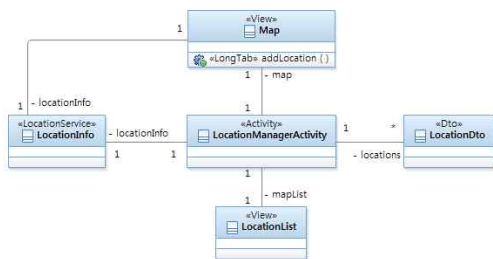


그림 10. 자동 생성한 초기 PSM  
Fig. 10. Automatic Generated Initial PSM

초기 PSM의 각 스테레오타입 별로 설정해야 할 속성 값을 설계 의도에 따라 결정하여 모델링 도구에서 기록한 후 XMI 형식으로 저장하였다. Map 클래스에 지정한 <<View>> 스테레오타입의 viewType 속성을 MapView로 설정하였으며, LocationInfo 클래스에 지정한 <<LocationService>> 스테레오타입의 경우 ServiceType 속성을 Gps로 지정하였다. 각 스테레오타입 별 속성 값을 지정한 후 모델링 도구에서 XMI 형식으로 변환한 결과와 스테레오타입별 템플릿을 정제 PSM 도구에 입력하여 XMI 형식의 정제 PSM을 생성하였다. 정제한 PSM은 템플릿에 의해 MapActivity, LongTabOverlay, MapActivityIntent, 그리고 BaseAdapter 클래스가 자동으로 추가되었으며, 클래스 사이의 연관 관계가 재정의 되었다. 정제한 PSM은 모델링 도구를 통해 코드로 변환할 수 있었다. 그림 11은 정제 PSM 생성기를 통해 생성한 XMI 형식의 정제한 PSM의 모델과 이를 사용하여 구현한 안드로이드 기반 사례 어플리케이션의 실행 화면을 나타낸다.

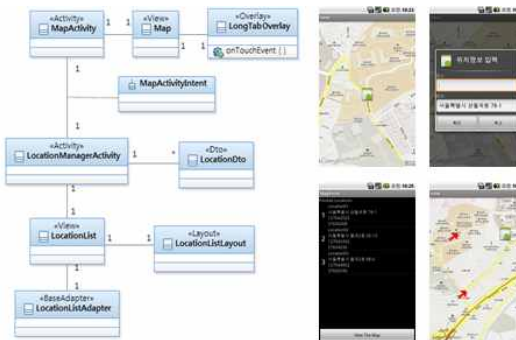


그림 11. 자동 생성한 PSM과 구현 결과  
Fig. 11. Automatic Generation Result of PSM and Result of Implementation.

4. 기존 연구와의 비교 분석

본 논문에서 제안한 모바일 어플리케이션의 UML 프로파일 구성 기법과 이를 활용한 모델 변환 기법을 기존의 MDD 적용 모바일 어플리케이션 개발 연구와 비교 분석하였다. 제안 기법의 유용성은 MDD 설계의 지원, 모델 변환 지원, 그리고 기법을 통해 얻어지는 산출물 등의 활용성으로 구분하여 평가하였다.

MDD 설계 지원 항목은 MDD를 적용한 설계를 위해 필요한 프로파일의 구성 지침 및 기법, 그리고 어플리케이션 아키텍처의 표현성으로 세분하였으며, 해당 항목을 직접 지원할

표 6. MDD 기반의 모바일 어플리케이션 모델 변환 기법의 유용성 비교  
 Table 6. Usability Comparison between Model Transformation Techniques of Mobile Applications based on MDD

평가 항목	기법	시각 모델 기반의 개발 기법	ATL 기반 모델 변환 기법	XOZ 기반 모델 변환 기법	제안 기법
MDD 설계 지원	PIM 프로파일 구성	-	×	×	○
	PSM 프로파일 구성	-	△	×	○
	어플리케이션 아키텍처 표현	△	○	△	○
모델 변환 지원	변환 규칙 구조	×	○	○	○
	모델 자동 변환	○	○	○	○
	소스 코드 생성	○	△	○	△
활용성	표준 형식 사용	×	△	△	○
	정보 변경 용이성	×	△	△	○
	추적 정보 획득 용이성	×	△	△	○

○ : 지원, △ : 부분지원, × : 미지원, - : 관련 없음

경우 '지원', 일부분만 지원할 경우 '부분지원', 지원하지 않을 경우 '미지원', 별도의 방법을 적용하는 경우 '관련없음'으로 평가하였다. 모델 변환 지원 항목의 경우 해당 평가 항목을 연구에서 직접 지원할 경우 '지원', 일부분만 지원할 경우 부분지원, 그리고 지원하지 않을 경우 '미지원'으로 평가하였다. 활용성의 표준 형식 사용 항목의 경우 모델과 모델 변환 관련 정보에 XML과 같은 표준으로 정해진 정보 교환 양식을 사용할 경우 '지원', 표준 교환 양식과 별도의 양식을 함께 사용할 경우 '부분지원', 독자적인 형식을 사용할 경우 '미지원'으로 평가하였다. 정보 변경 용이성의 경우 모델 정보의 변경을 변환 규칙과 변환 정보에 반영할 때 변환 규칙을 변경하지 않고 반영이 가능할 경우 '지원', 변환 규칙의 변경을 동반할 경우 '부분지원', 별도의 정보 변경 방법을 제공하지 않을 경우 '미지원'으로 평가하였다. 추적 정보 획득 용이성의 경우 모델 정보의 추적 시 모델 정보와 변환 정보에서 추적 정보 획득이 가능할 경우 '지원', 모델 정보 및 변환 정보 이외에 규칙을 구성하는 정보를 분석하여야 추적 정보 획득이 가능할 경우 '부분 지원', 모델 추적에 필요한 세부 정보를 지원하지 않을 경우 '미지원'으로 평가하였다. 표 6은 비교 결과를 나타낸다.

연구 [9]의 시각 모델 기반 모바일 어플리케이션 개발 기법은 MDD의 개념을 적용하여 시각 모델을 기반으로 어플리케이션을 자동 생성하는 방법을 제공하고 있으나, MDD 설계를 위한 구체적인 지침 및 관련 정보는 제공하고 있지 않다. 연구 [10]의 ATL 기반 모델 변환 기법은 스테레오타입을 사용한 모델 설계를 지원하고 ATL을 활용하여 타겟 독립 모델로부터 모바일 어플리케이션의 아키텍처를 구체적으로 변환

할 수 있는 장점이 있다. 그러나 해당 연구에서 타겟 독립 모델로 표현하는 플랫폼 독립 모델과 종속 모델 사이의 변환 방법을 제공하고 있지 않으며, 모델 설계 시 변환 대상을 지정하기 위해 사용하는 스테레오타입 구성을 위한 구체적인 방법은 제공하고 있지 않다. 연구 [11]의 XOZ 기반 모델 변환 기법은 MDD를 적용하여 하나의 모델로부터 여러 모바일 플랫폼 상의 모바일 어플리케이션 사용자 인터페이스를 생성하는 기법을 제안하였다. 모델 변환을 위한 규칙을 제시하고 있으며 이를 통해 플랫폼의 네이티브 코드를 생성할 수 있는 장점을 갖고 있으나 모바일 어플리케이션 아키텍처 전체가 아닌 사용자 인터페이스 부분만으로 적용 범위가 제한되어 있으며, 표준 형식인 XMI로부터 XOZ라는 해당 연구에서 제안한 별도의 형식으로 변환하는 노력이 요구되고 있다. 본 논문에서는 PIM/PSM 설계에 적용할 프로파일의 구성을 위한 구체적인 방법을 제시하였고 이를 활용한 변환 방법을 제시하였다. 더불어 플랫폼 변경 등에 따른 정보 변경을 모델 변환에 용이하게 적용하기 위한 템플릿을 제시하였으며, 이를 활용하여 모델 변환 정보와 변환 규칙을 분리 구성하여 모델 변환 정보 변경의 용이성을 향상을 기대할 수 있으며, XML 기반의 산출물을 활용하므로 정보 접근성 및 추적성을 향상시켰다.

## VII. 결론

모바일 어플리케이션 개발에 MDD를 적용하기 위해서는 모바일 어플리케이션의 특징을 반영한 UML 프로파일을 사용하여 PIM과 PSM을 설계한 후 모델 변환 규칙에 따라 변

환을 수행하여야 한다. 이에 본 논문에서는 모바일 어플리케이션의 UML 프로파일 구성 기법과 이를 활용한 모델 변환 기법을 제안하였다.

모바일 어플리케이션의 특징을 반영하여 모델에 적용할 스테레오타입의 유형을 분류하였고 이에 기반한 프로파일 구성 지침을 제시하였다. 모델 변환을 위해 프로파일 대응규칙 구조를 제시하였고 PSM에 모바일 플랫폼의 세부적인 특성을 반영하기 위한 방안으로 PSM 템플릿을 제시하였으며, 이를 활용한 모델 변환 절차를 제시하였다. 또 제안한 기법을 적용할 수 있는 자동화 도구를 개발하였다. 제안 기법의 유용성을 확인하기 위하여 기법을 적용하여 안드로이드 기반 위치 기록 어플리케이션 개발하여 결과를 확인하였으며, 기존 연구와의 비교를 수행하였다.

제안 기법은 모델 정보를 표준 모델 정보 교환 형식을 적용하였으므로 기존 모델링 도구의 활용이 가능하며, 프로파일 대응규칙 및 PSM 템플릿은 XML 기반으로 구성하였으므로 변환 정보 구성이 용이하고 다른 도구와의 통합 및 정보 교환이 용이하다. 또, 템플릿을 활용하여 변환 정보를 변환 규칙으로부터 분리하여 구성하였으므로 템플릿 변경만으로 모바일 플랫폼 변경 사항을 모델 변환에 반영할 수 있다.

향후에는 사례연구에서 적용한 안드로이드 플랫폼 이외의 iOS, Windows Mobile과 같은 멀티 플랫폼으로 확장하여 프로파일 구성 지침과 변환 규칙 및 템플릿을 보완하는 연구와 제안한 기법을 지원하는 도구의 추가적인 개발이 필요하다. 또 제안한 기법으로 생성한 프로파일, 대응규칙과 템플릿을 활용한 모델 변환의 추적 기법 연구가 필요하다.

## 참고문헌

- [1] MDA Guide Version 1.0.1, <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- [2] Colin Atkinson, Thomas Kühne, "Model-Driven Development: A Metamodeling Foundation", IEEE Software, Vol. 20, Issue: 5, pp.36-41, Sept. 2003
- [3] H. G. Min, S. D Kim, "A UML Profile for Specifying Component Design as MDA/PIM", Journal of KISS : Software and Applications, Vol.32, no.3, pp.153-162, Mar. 2005
- [4] Tuukka Ritala, Seppo Kuikka, "UML Automation Profile: Enhancing the Efficiency of Software Development in the Automation Industry", Industrial Informatics, 2007 5th IEEE International Conference, Vol.2, pp.885-890, June 2007
- [5] A. Kleppe, J. Warmer, and W. Bast. "MDA Explained", Pearson Education, 2003
- [6] Stephen J. Mellor, Anthony N. Clark, Takao Futagami, "Model-Driven Development", IEEE Software, Vol.20, Issue: 5, pp.14-18, Sept. 2003
- [7] Hassan Charaf, "Keynote: Developing Mobile Applications for Multiple Platforms", Engineering of Computer Based Systems (ECBS-EERC), pp.2-2, Sept. 2011
- [8] Unified Modeling Language, [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#)
- [9] Florence T. Balagtas-Fernandez, Heinrich Hussman-n, "Model-Driven Development of Mobile Applications", Automated Software Engineering, 2008. ASE 2008. 23rd IEEE/ACM International Conference on, pp.509-512, Sept. 2008.
- [10] W. Y. Kim, H. S. Son, J. S. Kim, and Y. C. Kim, "Development of Windows Mobile Applications using Model Transformation Techniques", Journal of KISS(C): Computing Practices and Letters, vol.16, no.11, pp.1091-1095, Nov. 2010.
- [11] S. T. Ryu, C. H Park, E. S. Lee, "Automatic Generation of GUI Source code in Mobile Applications for Multi-platform", Journal of KISS : Software and Applications, vol.38, no.4, pp.198-206, April 2011.
- [12] MOF 2 XMI Mapping, <http://www.omg.org/spec/XMI/2.4.1/PDF>
- [13] Meta Object Facility, <http://www.omg.org/spec/MOF/2.4.1/PDF>
- [14] iOS App Programming Guide, <https://developer.apple.com>
- [15] Android User Interface, <http://developer.android>

com/guide/topics/ui/index.html

- [16] J. E. Rumbaugh, "Modeling models and viewing views: A look at the model-view-controller framework," JOOP : Journal of Objectoriented programming, vol. 7, no. 2, pp. 14-20, 29, May 1994.
- [17] Iulia-Maria, T., Ciocarlie, H., "Best practices in iPhone programming: Model-view-controller architecture-Carousel component development", International Conference on Computer as a Tool, 2011. EUROCON 2011, pp.1-4, April 2011

### 저 자 소 개



최 윤 석

1997: 숭실대학교

소프트웨어공학과 공학사.

1999: 숭실대학교 일반대학원

컴퓨터학과 공학석사.

2008: 숭실대학교 일반대학원

컴퓨터학과 공학박사

현 재: 동덕여자대학교

컴퓨터학과 전임강사

관심분야: Mobile Software, MDA,

AOP, Software Architecture

Email : cooling@dongduk.ac.kr