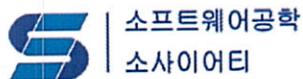




한국정보과학회
Korea Institute of Information Science and Technology

제 16 권 제 1 호
Vol. 16 No. 1



2014

제 16 회 한국 소프트웨어공학 학술대회 논문집

Proceedings of the 16th Korea Conference on
Software Engineering (KCSE 2014)

- 일시: 2014년 2월 12일(수) ~ 2월 14일(금)
- 장소: 강원도 평창 한화리조트(휘닉스파크점)

주최: 한국정보과학회, 한국정보처리학회
주관: 한국정보과학회 소프트웨어공학 소사이어티,
 한국정보처리학회 소프트웨어공학 연구회,
 한국전자통신연구원

후원: (주)비트컴퓨터, (주)다한테크, (주)솔루션링크,
 슈어소프트테크(주), 정보통신산업진흥원 SW 공학센터,
 현대엠엔소프트(주), ITRC 고품질융합소프트웨어연구센터,
 STA 테스팅컨설팅(주), TTA 소프트웨어 시험인증연구소

SW 모델 설계 1 (B1)

변수 개수 최소화를 통한 소프트웨어 모델 체킹 성능향상 연구

이낙원, 박진희, 백종문(KAIST)

Active Device Pool 기반의 IoT 디바이스 동적 연결 기법[최우수논문]

박춘우, 양신형, 라현정, 김수동(숭실대)

에너지 효율적인 모바일 컨텍스트 모니터링 기법 설계[우수논문]

김문권, 이재유, 김수동(숭실대)

시뮬레이션 모델의 검증 지원을 위한 개선된 유한 실시간 이산 사건 시스템 명세[단편논문]

윤지현, 박송이, 김단희, 이해영(서울여대)

SW 모델 설계 2 (C1)

이종 스마트폰 UI의 모델변환을 위한 메타모델 설계

손현승(홍익대), 김우열(대구교대), 김영철(홍익대)

소프트웨어 개발 조직의 지식 공유를 위한 지식관리 시스템 및 메타데이터 모델에 대한 연구

고하윤, 백원길, 이지훈, 민상윤(KAIST)

A Comparison of Model-Based Software Development Approaches

Aman jaffari, Mesfin Abebe Haile, 유철중, 신정훈(전북대)

Fillmore's Case 메커니즘 이용한 요구사항 분석:유스 케이스 모델링 방법 [단편논문]

김보연, 양효석, 손현승, 박병호(홍익대), 박용범(단국대), 김영철(홍익대)

이종 스마트폰 UI의 모델변환을 위한 메타모델 설계

손현승*, 김우열**, 김영철*

* 홍익대학교 컴퓨터정보통신공학과 소프트웨어공학연구실

세종특별자치시 조치원읍 세종로 2639

{son, bob}@selab.hongik.ac.kr

** 대구교육대학교 컴퓨터교육과

대구광역시 남구 중앙대로 219

john@dnue.ac.kr

요약: 지금까지는 앱을 안드로이드, 아이폰, 윈도우즈 폰과 같은 스마트폰 플랫폼에서 각각 개발해왔다. 이종 플랫폼의 동시 개발을 위해서는 모델변환 기법을 사용한다. 그러나 기존의 스마트폰에서 모델변환은 주로 코드생성에 초점이 맞추어져 있거나 UI를 객체 다이어그램으로 모델링하여 직관적으로 UI를 표현하지 못하는 문제점이 있다. 본 논문에서는 스마트폰의 UI에 모델변환 기법을 적용하여 이종의 플랫폼의 UI를 생성할 수 있는 방법을 제안한다. 제안한 방법은 각 플랫폼의 UI 특징을 추출한 공통 UI 모델을 안드로이드, 아이폰, 윈도우즈 폰과 같은 각 플랫폼의 네이티브 UI 모델로 변환하는 것이다. 제안한 방법의 실현을 위해서는 메타모델 설계와 모델변환 규칙 작성이 필요하지만, 본 논문에서는 모델변환의 기반인 각 플랫폼의 메타모델의 정의에 대해서 다룬다. 향후 이 메타모델을 사용하고 모델변환 규칙을 만들면 이종의 플랫폼의 UI를 모델변환을 수행할 수 있다.

핵심어: 모델변환, 스마트폰, 메타모델, 사용자인터페이스, 멀티 플랫폼

1. 서론

지난 2007년 애플의 아이폰의 탄생을 시작으로 스마트폰에 대한 관심은 계속 증가되었고 스마트폰은 생활에 필수 요소로 자리잡았다. 스마트폰의 가장 큰 매력은 PC의 다양한 소프트웨어처럼 다양한 앱을 사용할 수 있는 점이다. 기존의 일반적인 패키지 소프트웨어와 다르게 앱은 규모가 크지 않고 핵심적인 기능만 포함하며 배우지 않더라도 간단하고 쉽게 사용할 수 있는 특징이 있다.

PC에서 소프트웨어의 실행과 제작환경인 플랫폼이 핵심기술인 것처럼 스마트폰에서도 앱을 수행할 수 있는 플랫폼이 핵심기술이다. 각 제조사들은 PC

시장의 MS 윈도우즈와 같이 스마트폰 시장에서도 플랫폼을 선점한 기업이 시장의 주도권을 잡을 수 있다고 믿고 있다. 각 스마트폰 제조 및 공급 회사들은 각각 플랫폼을 출시하였고 이를 선점하기 위해서 플랫폼 전쟁이 시작되었다[1].

플랫폼 전쟁은 안드로이드, 아이폰, 윈도우즈 폰, 타이젠 등과 같은 다양한 플랫폼을 양산하게 되었다. 플랫폼의 다양화는 소프트웨어의 개발에서 매우 좋지 않은 현상이다. 그 이유는 특정 플랫폼으로 개발된 소프트웨어는 플랫폼 종속적이며 다른 플랫폼의 소프트웨어 개발에 재사용 하지 못하기 때문이다. 즉, 플랫폼의 다양화는 같은 콘텐츠를 각 플랫폼마다 새로 만들어야 하는 중복 개발 문제를 야기 시킨다. 또한 여러 플랫폼으로 개발된 소프트웨어의 유지보수는 더 많은 비용이 소모된다.

모델변환은 MDD(Model Driven Development) [2]의 핵심 기술로 플랫폼에 독립적인 모델을 종속적인 모델로 변환하는 기술이다. 모델변환은 모델의 메타모델, 변환언어, 실행엔진으로 구성된다. 모델변환의 수행은 메타모델 설계, 변환규칙 작성, 모델변환 실행의 3 단계로 수행된다. 첫 번째 메타모델 설계 단계에서는 입력/출력하는 모델의 구조를 파악하여 메타모델을 설계한다. 두 번째 변환규칙 작성단계에서는 두 모델간의 공통점과 차이점을 분석하여 변환규칙을 만들고 이를 변환언어로 작성한다. 세 번째 모델변환 실행 단계에서는 모델, 메타모델, 변환언어를 실행엔진에 입력하고 수행하면 변환언어의 명령으로 변환된 모델을 생성한다.

우리는 이러한 모델변환을 이종의 스마트폰 개발에 적용하기 위한 연구를 수행하였다. 최초의 시도는 ATL (ATLAS Transformation Language) [3]를 사용하여 윈도우즈 모바일 어플리케이션을 개발에 적용한 것이다 [4]. 플랫폼 독립적인 모델과 종속적인 모델을 UML 클래스 다이어그램으로 표현하였고 이 둘을 자동으로 변환하였다. 기본 구조 생성, 클래스 변경, 메소드 변경 3 가지 카테고리로 분리되는 것을

[†]본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업원천기술개발사업[10035708, 고신뢰 자율제어 SW를 위한 CPS(Cyber-Physical Systems) 핵심 기술 개발]과 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2013R1A1A2011601).

확인하였다. 이것을 기반으로 안드로이드, 아이폰에도 적용하여 하나의 독립 모델을 사용하여 이종의 플랫폼 종속적인 모델로 변환 가능한 것을 확인할 수 있었다[5-9]. 그리고 이렇게 만들어진 플랫폼 종속 모델을 Acceleo[10]를 사용하여 코드 템플릿을 만들고 실행하여 코드까지 생성해 보았다[11]. 결론적으로 모델로부터 코드 단계까지 부분적인 자동화가 가능한 것을 확인해 볼 수 있었다. 그리고 최근에는 시퀀스 다이어그램에 적용하였고[12] 모델변환언어를 연구 중이다[13].

기존 연구결과 완벽한 스마트폰 플랫폼 환경을 위해서는 화면을 구성하는 UI, 동작을 수행하는 코드, 개발 환경을 구성하는 프로젝트의 3 가지 요소에 대한 변환방법이 필요하다. 그러나 아직 스마트폰의 모델변환적용은 주로 코드생성에 초점이 맞추어져 있다. 최근에 타 연구에서는 UML의 프로파일을 이용하여 클래스 다이어그램으로 코드 생성하고 객체 다이어그램으로 UI를 생성하는 시도도 있었다[14]. 하지만 프로젝트에 관한 부분이 없고 UI를 객체 다이어그램으로 모델링하기 때문에 직관적으로 UI를 표현하지 못하는 문제점을 가지고 있다.

본 논문에서는 기존 연구 내용의 보완 및 확장을 위해서 스마트폰의 UI에 모델변환 기법을 적용한 방법을 제안한다. 제안한 방법은 각 플랫폼의 UI 특징을 추출한 공통 UI 모델을 이용하여 안드로이드, 아이폰, 윈도우즈 폰과 같은 각 플랫폼의 네이티브 UI 모델로 변환하는 것이다. 공통 UI 모델 개발을 통해서 기존의 객체 다이어그램이 가지는 가시성 문제를 해결하였다. 제안한 방법의 실현을 위해서는 메타모델 설계와 모델변환 규칙 작성이 필요하지만, 본 논문에서는 모델변환의 기반인 각 플랫폼의 메타모델을 정의에 대해서 다룬다. 향후 이 메타모델을 사용하고 모델변환 규칙을 만들면 이종의 플랫폼의 UI를 모델변환을 수행할 수 있다.

이 논문의 구성은 다음과 같다. 2 장에서는 관련연구로 모델변환에 대해서 언급한다. 3 장에서는 제안한 UI 모델변환 방법에 대해서 설명하고, 4 장에서는 메타모델 설계 방법을 보여준다. 5 장에서는 적용사례를 통해 설계된 메타모델을 검증한다. 마지막으로 6 장에서는 결론 및 향후 연구에 대해서 언급한다.

2. 관련연구



모델변환의 주요 개념은 그림 1과 같다. 그림은 입력 모델에서 출력 모델로의 변환에 대한 간단한 시나리오를 보여준다[15]. 입력/출력 양쪽의 모델은 각각의 메타모델을 따른다. 메타모델은 모델링의 표기법의 추상적인 구문을 일반적으로 정의한 것이다. 모델변환은 메타모델을 참고하여 작성된 변환언어에 따라 수행된다. 변환언어는 변환규칙을 통해서 작성될 수 있는데, 변환규칙은 두 모델간의 공통점과 차이점을 자연어 형태로 작성한 것을 의미하고 변환언어는 프로그램 언어와 같이 변환엔진을 수행할 수 있는 명령어들로 구성되어 있다. 모델변환을 수행하기 위해서는 메타모델, 변환언어, 변환엔진이 요구된다.

3. 스마트폰 UI의 모델변환

우선 UI의 모델변환의 수행을 위해서는 UI 모델에 대한 이해가 필요하다. 그림 2는 안드로이드, 아이폰, 윈도우즈 폰에서 버튼을 만들었을 때 생성되는 UI 파일의 예이다.

```

...(생략)...
<AbsoluteLayout>
    <Button android:id="@+id/button1"
    android:layout_width="50dp"
    android:layout_height="40dp" android:layout_x="38dp"
    android:layout_y="230dp" android:text="Btn1"/>
</AbsoluteLayout>
...(생략)...
  
```

(가) 안드로이드의 activity_main.xml

```

...(생략)...
<object class="IBUIButton" id="3169532110">
...(생략)...
<string key="NSFrame">[206, 79], [75, 45]</string>
...(생략)...
<string key="IBUINormalTitle">Btn1</string>
...(생략)...
</object>
...(생략)...
  
```

(나) 아이폰의 ViewController.xib

```

...(생략)...
<Grid x:Name="ContentPanel" Grid.Row="1"
Margin="12,0,12,0">
    <Button Height="90" HorizontalAlignment="Left"
Margin="311,103,0,0" Name="button1" Content="Btn1"
VerticalAlignment="Top" Width="133"
Click="Button_click" />
</Grid>
...(생략)...
  
```

(다) 윈도우즈 폰의 MainPage.xaml
그림 2 각 플랫폼 별 UI 파일의 예

여기서 각 플랫폼의 UI는 XML로 표현된다. 그리고 이들의 UI 파일 구조는 전혀 다르다는 것을 알 수 있다. 이 예에서 공통점을 추출해 보면 버튼의 이

름과 좌표 정도가 된다. 이름과 좌표는 그림에 볼드체로 표현하였다. 각 플랫폼마다 기능과 구조는 다르지만 공통적으로 갖는 정보가 있기 때문에 이러한 정보를 추상화하여 우리는 공통 UI 모델을 이용한다. 이 모델을 통해서 각각의 플랫폼에 해당하는 모델로 변환하면 하나의 UI 를 통해서 각종의 플랫폼 UI 들을 생성할 수 있다. 또한 모델변환을 수행시 생성되는 파일들은 XMI 로 표현되므로 UI 모델 변환은 XMI [16]를 XML 로 변환하는 단계도 필요하다.

이러한 각 플랫폼들의 UI 의 구조와 정보를 분석하여 그림 3 과 같이 변환방법을 제안한다. 제안한 방법은 모델에서 모델로 변환하는 단계와 XMI 파일을 XML 로 변환하는 두 단계로 구성한다. 첫 번째 모델에서 모델 변환 단계에서는 공통 UI 모델로부터 모델 데이터를 입력 받아 모델변환을 통해 각 플랫폼 별로 UI 모델들을 생성한다. 이렇게 생성된 파일들은 모두 XMI 파일로 생성된다. 두 번째 단계에서는 이 XMI 파일을 XML 변환하여 각 플랫폼 별 개발도구에서 데이터를 읽을 수 있도록 한다.

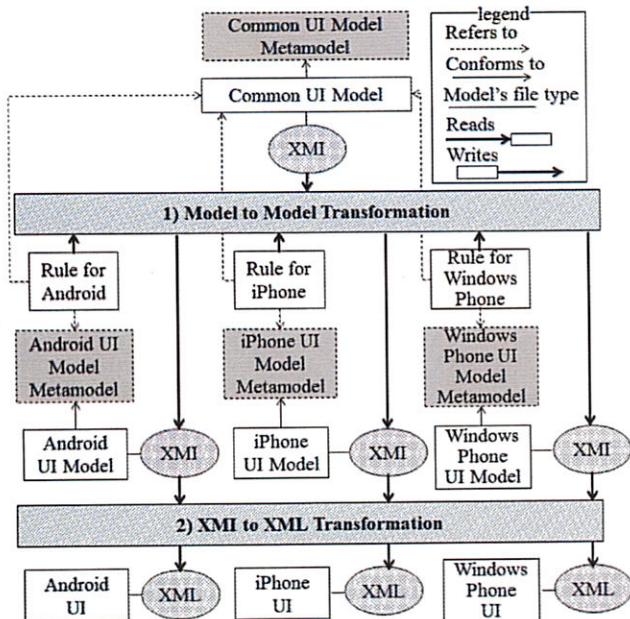


그림 3 UI의 모델변환 과정

제안한 방법을 실현하기 위해서는 메타모델과 변환규칙이 필요하다. 하지만 본 논문에서는 모델변환 수행의 기반이 되는 메타모델에 대해서만 다룬다.

4. 메타모델 설계

메타모델 설계는 모델변환을 수행하기 위해 반드시 필요한 단계이다. 그러나 각 플랫폼에 대한 메타모델은 제공되지 않으므로 UI 데이터를 보고 역설계를 해야 한다. 본 장에서는 공통 UI 모델의 메타모델

설계와 안드로이드, 아이폰, 윈도우즈 폰의 메타모델의 역설계 방법에 해서 설명한다.

4.1 공통 UI 모델 설계

공통 UI 모델은 각 플랫폼들의 UI 들의 특징들을 분석하여 추상화하여 만든 새로운 모델이다. 우리는 표 1 과 같은 종류의 UI 모델을 사용할 수 있도록 적용하였다. 각 플랫폼의 UI 파일들을 분석한 결과 UI 컴포넌트들은 이름, 위치(x,y), 넓이, 높이, 이벤트 리스너 정보들이 공통적으로 사용되는 것을 확인 할 수 있었다.

표 1 공통 UI 모델

이름	표기법	설명
Static	Example	문자 출력
Edit		텍스트를 입력상자
Button		버튼
Slider		슬라이더
Check		체크 박스
Radio		라디오 버튼

이러한 정보를 들을 기반으로 그림 4 와 같이 공통 UI 모델의 메타모델을 설계 하였다. UIModel 은 모델의 최상위 루트노드이다. UIElement 는 각 UI 를 의미하는데, UI_Type 과 이름 리스너를 속성으로 갖는다. UIElement 하위에 있는 OutSize 는 UI 의 크기 정보를 나타내며, DefSize 는 UI 가 가지는 최소 크기를 의미한다.

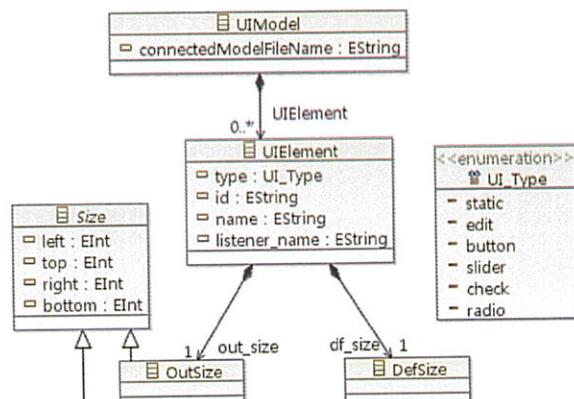


그림 4 공통 UI 모델의 메타모델

공통 UI 모델은 그림 5 와 같이 자체 개발한 UI 편집 도구로 설계 가능하다.

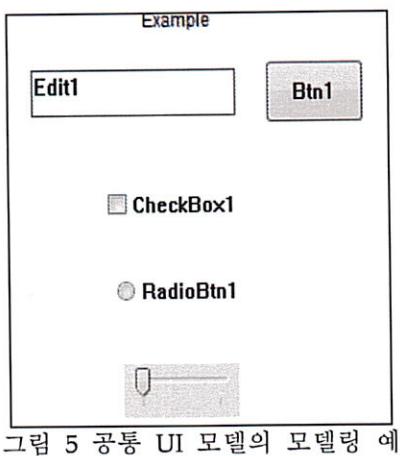


그림 5 의 모델을 XMI 데이터로 표현하면 그림 6 과 같다. XMI 에 태그를 보면 메타모델에서 정의한 요소들이 있음을 확인할 수 있다. 이렇게 메타모델은 모델에 대한 스키마 정보로 객체지향 모델링으로 표현되고 XMI 형태로 데이터가 저장된다.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<UIModel connectedModelFileName="CDM1.cdm">
  <UIElement type="static" id="4104862c-545f-470c"
    name="Example" listener_name="">
    <out_size left="123" top="48" right="191" bottom="72"/>
    <df_size left="50" top="52" right="90" bottom="66"/>
  </UIElement>
  <UIElement type="edit" id="6d1004b5-7bc7-4d2b"
    name="Edit1" listener_name="">
    <out_size left="42" top="84" right="191" bottom="119"/>
    <df_size left="27" top="89" right="87" bottom="115"/>
  </UIElement>
  <UIElement type="button" id="9b194bc6-dd40-46b4"
    name="Btn1" listener_name="Button_click">
    <out_size left="206" top="79" right="281" bottom="124"/>
    <df_size left="210" top="80" right="260" bottom="120"/>
  </UIElement>
  <UIElement type="slider" id="b3ed12e5-76f8-4833"
    name="Slider1" listener_name="slider_click">
    <out_size left="105" top="270" right="192" bottom="310"/>
    <df_size left="59" top="230" right="139" bottom="256"/>
  </UIElement>
  <UIElement type="check" id="9a6ead45-715c-40e4"
    name="CheckBox1" listener_name="Checkbox_click">
    <out_size left="95" top="153" right="206" bottom="190"/>
    <df_size left="13" top="179" right="93" bottom="205"/>
  </UIElement>
  <UIElement type="radio" id="2a23c49d-f510-4617"
    name="RadioBtn1" listener_name="Radio_click">
    <out_size left="101" top="210" right="198" bottom="243"/>
    <df_size left="85" top="213" right="165" bottom="239"/>
  </UIElement>
</UIModel>
```

그림 6 공통 UI 모델의 XMI 파일의 내용

4.2 각 플랫폼 UI의 메타모델 역설계

공통 UI 모델은 기존에 없는 모델이기 때문에 UI 데이터를 기반으로 설계자 임의로 메타모델을 설계하면 되지만 각 플랫폼의 UI 메타모델은 그렇게 할 수 없다. 기존의 UI 파일의 데이터의 스키마에 해당하는 메타모델을 임으로 설계한다면 모델과 메타모델의 관계가 불일치 되기 때문이다. 그러므로 UI 데이터를 기반으로 메타모델을 역으로 설계해야 한다.

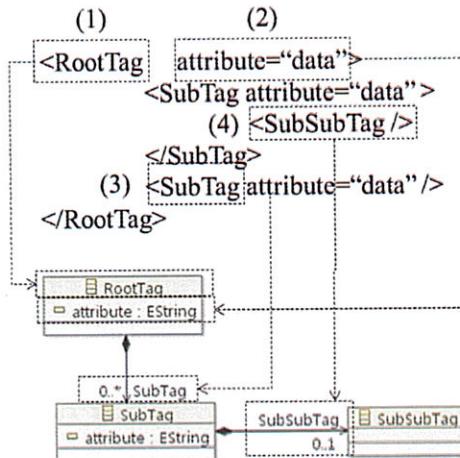


그림 7 메타모델의 역설계 방법 1

그림 7 은 XMI 데이터를 기반으로 메타모델을 역설계하는 방법이다. 기본적으로 4 가지 형태로 구분할 수 있다. (1)은 최상위 요소로 대부분 하나의 값을 가진다. 그림에서 (1)은 메타모델의 최상위 모델이 된다. 그렇기 때문에 XMI의 태그 이름과 같다. (2)는 XMI의 속성 값이다. 속성의 이름은 클래스의 속성 이름으로 표기되고 속성의 값은 데이터의 형태가 입력된다. 예를 들어 문자열은 EString, 숫자는 Eint, 실수는 Efloat 으로 데이터의 형이 입력된다. (3)은 서브 요소로 루트 요소와 합성관계로 표현된다. (4)는 서브의 서브요소로 (3)과 같이 합성관계로 표현된다. 합성관계에서 데이터의 개수에 따라 0..1 또는 0..* 와 같이 UML의 클래스다이어그램의 개수 표현 방식처럼 사용된다.

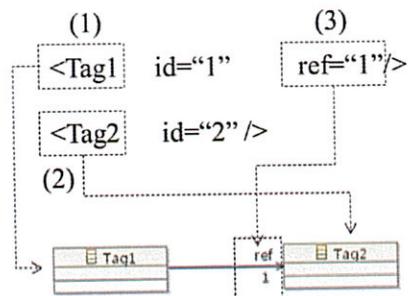


그림 8 메타모델의 역설계 방법 2

메타모델에서의 참조관계는 그림 8 과 같다. 참조 관계를 위해서는 (1)과 (2)처럼 각 태그요소에 해당하는 메타모델의 클래스가 있어야 한다. 그리고 참조 관계에서는 XMI에 ID가 존재해야 한다. ID가 참조하는 요소를 나타내기 때문이다. (3)은 참조할 속성의 이름으로 그림의 예에서는 ref라는 이름으로 정의하였고 이것은 메타모델에서는 참조관계에 이름으로 표현된다.

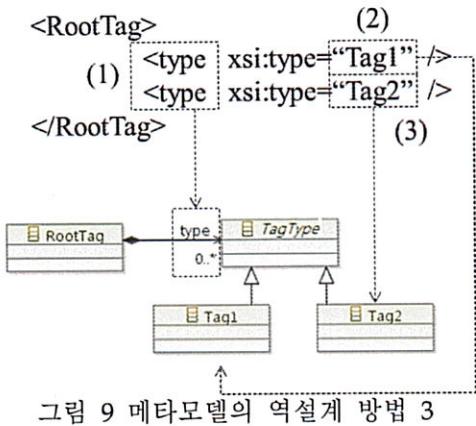


그림 9 메타모델의 역설계 방법 3

그림 9 와 같이 하나의 정의로 태그 안에서 여러 개로 선택해야 할 때도 있다. 이러한 경우에는 (1)처럼 합성관계를 먼저 표현해 주고 데이터의 구분은 (2)와 (3)처럼 xsi:type 으로 구분한다. 메타모델은 추상클래스를 만들고 선택해야 하는 두 개의 클래스는 일반화 관계로 표현한다.

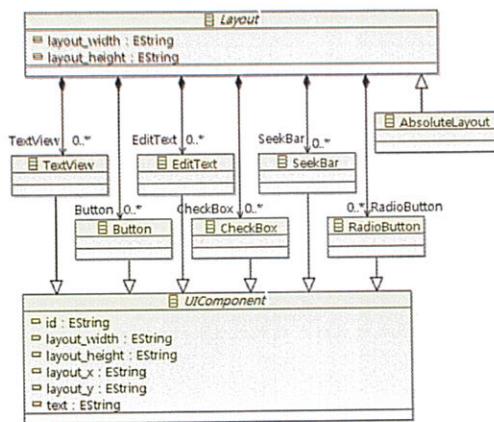


그림 10 안드로이드 UI 모델의 메타모델

우리는 앞에서 제시한 메타모델의 역설계 방법을 이용하여 각 플랫폼 별로 메타모델을 설계하였다. 그림 10은 안드로이드의 UI의 메타모델을 설계한 것이다. 안드로이드의 전체 UI에 대한 메타모델은 이보다 복잡하지만 본 논문에서는 공통 UI와 관련된 모델들의 데이터에 해당하는 부분만 메타모델로 설계하였다. 안드로이드 Layout을 하위로 UI 컴포넌트

들이 구성되고 각 UI 컴포넌트는 TextView, EditText, Button, CheckBox, SeekBar, RadioButton의 이름으로 구성된다.

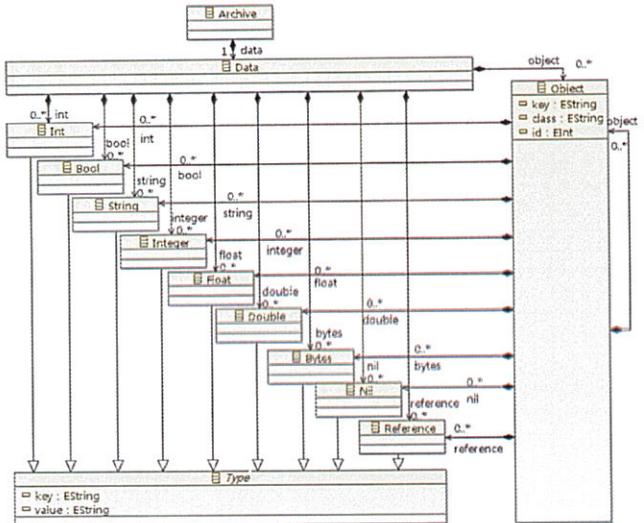


그림 11 아이폰 UI 모델의 메타모델

그림 11 은 아이폰의 메타모델이다. 아이폰은 UI 컴포넌트의 이름이 Object 클래스의 속성인 class에 표현된다. 각 Object 와 Data 는 여러 가지 형태의 데이터 타입을 가질 수 있다. 그림은 복잡해 보이지만 메타모델의 구성은 간단하다. 하지만 데이터의 타입을 지정하기 때문에 UI 파일의 내용은 복잡하고 양이 많은 특성을 가진다.

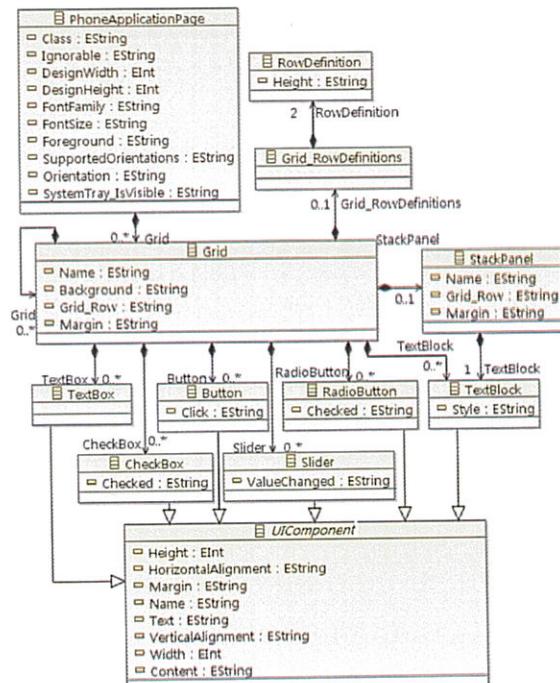


그림 12 윈도우즈 폰 UI 모델의 메타모델

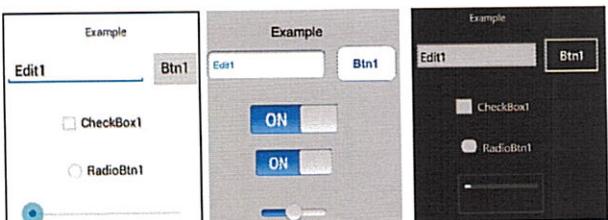
그림 12는 윈도우즈 폰의 메타모델이다. UI를 구성하기 위한 속성데이터가 많아 복잡해 보이지만 기본적인 형태는 안드로이드와 유사하다. 안드로이드의 Layout은 윈도우즈 폰에서 Grid로 사용되고 유사한 형태의 UI 컴포넌트들을 가지고 있다. 차이점은 안드로이드에는 리스너 함수를 입력할 수 없지만 윈도우즈 폰은 리스너 함수이름을 입력 가능하다. 그리고 메타모델의 이름에 “.”은 인식하지 못하기 때문에 “_”으로 변경하여 입력하였다. 예를 들어 “Grid_row”는 윈도우즈 폰에서는 “Grid.row”로 표현된다.

5. 적용사례

설계된 메타모델은 이클립스의 EcoreTools[17]와 EMF[18]를 이용하면 데이터가 정상적으로 읽을 수 있는지 검증할 수 있다. EcoreTools은 메타모델 설계 도구이다. EMF는 EcoreTools로 만든 메타모델을 자바 코드로 생성하여 프로그래밍을 할 수 있거나 이클립스 플러그인으로 모델 에디터를 생성하여 모델에 데이터를 입력해 볼 수 있는 도구이다. 설계한 메타모델의 검증과정은 다음과 같다.

1. EcoreTools로 메타모델을 설계 후 메타모델 파일(*.ecore)을 EMF 프로젝트로 로드한다.
2. 생성된 EMF 프로젝트에서 자동 생성 버튼을 눌러 모델 에디터를 생성한다.
3. UI XML 파일을 XMI로 변환한다.
4. 생성된 모델 에디터로 XMI 파일을 읽고 파일이 로드 되면 “validation”을 눌려 문제가 없는지 확인한다.

1~2 과정은 앞에서 설계한 메타모델을 EcoreTools로 모델링하는 것이다. 3 과정은 XMI 변환기를 사용하여 변환한다[19]. XMI 변환기는 시뮬링크를 위해 만들어 겼지만 XML을 XMI로 변환할 수 있으므로 프로그램의 약간의 수정으로 사용할 수 있다. 수정되는 정보는 각 메타모델의 네임스페이스 이름이다. 4 과정에서 성공 메시지가 나오면 메타모델과 데이터가 일치됨을 확인 할 수 있다.



(가) 안드로이드 (나) 아이폰 (다) 윈도우즈 폰
그림 13 각 플랫폼 별 UI 모델의 데이터

각 플랫폼 별로 메타모델 설계가 정상적으로 만들

어졌다면, 각 플랫폼의 UI 도구로 생성한 파일을 모델 에디터에서 데이터를 읽을 수 있고 유효성 검사를 때 성공 메시지가 나타나야 한다. 그림 13은 각 플랫폼 별 입력데이터로 사용한 UI 화면들이고 그림 14는 UI 파일을 모델 에디터로 로드하고 검증한 결과이다. 모두 메타모델과 일치하는 것을 확인하였다.

(a) 안드로이드

(나) 아이폰

(다) 윈도우즈 폰

그림 14 각 플랫폼 별 UI 모델의 검증

6. 결 론

모델변환은 이종의 플랫폼을 동시에 개발할 수 있는 방법이다. 그러나 기존의 스마트폰 모델변환의 적용은 주로 코드생성에 초점이 맞추어져 있다. 그러나 이종 플랫폼으로 완벽한 변환 수행을 위해서는 화면을 구성하는 UI, 동작을 수행하는 코드, 개발 환경을 구성하는 프로젝트의 3 가지 요소의 변환이 구축되어 있어야 한다. 최근 타 연구에서 UML 의 프로파일을 이용하여 코드와 UI 를 다루고 있지만 프로젝트에 관한 부분이 없고 UI 를 객체 디자인그램으로 모델링하기 때문에 직관적으로 UI 를 표현하지 못하는 문제점을 가지고 있다.

본 논문에서는 기존 연구의 이러한 내용의 보완 및 확장을 위해서 UI 의 모델변환 방법을 제안하였다. 제안한 방법은 각 플랫폼의 UI 특징을 추출한 공통 UI 모델을 이용하기 때문에 기존의 객체 디자인그램이 가지는 가시성 문제를 해결 하였고 모델변환 실현을 위해서 기존의 UI 데이터에서 메타모델을 역으로 설계 가능한 방법을 제시하였다. 또한 제시한 역설계 방법을 통해서 각 플랫폼의 UI 메타모델을 설계하고 적용사례를 통해서 UI 메타모델들을 검증하였다.

UI 모델변환을 수행하기 위해서는 메타모델뿐만 아니라 변환규칙에 대한 정의도 필요하다. 향후 연구로 각 스마트폰 UI 의 특성을 분석하여 모델변환 규칙에 대해 연구 중이다. 변환규칙을 이용하면 이종의 플랫폼의 UI 를 모델변환을 수행할 수 있을 것으로 본다.

참고문헌

- [1] Brad Cho, Platform wars, BOOK21, 2013.
- [2] Bran Selic, "The pragmatics of model-driven development", Software, IEEE, Vol. 20, Issue 5, pp. 19-25, 2003.
- [3] Wikipedia, ATL, http://en.wikipedia.org/wiki/ATLAS_Transformation_Language
- [4] Woo Yeol Kim, Hyun Seung Son, Jae Seung Kim, Robert YoungChul Kim, " Development of Windows Mobile Applications using Model Transformation techniques", Journal of KIISE : Computing Practices and Letters, Vol. 16, No. 11, pp. 1091-1095, 2010.
- [5] Wooyeol Kim, Hyunseung Son, Junbeom Yoo, Young B. Park, R. Youngchul Kim, "A Study on Target Model Generation for Smartphone Applications using Model Transformation Technique", International Conference on Internet (ICONI) 2010, Vol. 2, pp. 557-558, Dec. 2010.
- [6] Woo Yeol Kim, Hyun Seung Son, Jae Seung Kim, Robert Young Chul Kim, "Adapting Model Transformation Approach for Android Smartphone Application", Advanced Communication and Networking, Springer CCIS 199, pp. 421-429, Aug. 2011.
- [7] Woo Yeol Kim, "Model Transformation Framework for Heterogeneous Mobile Embedded Platforms", Hongik University thesis 2011.
- [8] Woo Yeol Kim, Hyun Seung Son, Robert Young Chul Kim, "A Study on UML Model convergence Using Model Transformation Technique for Heterogeneous SmartPhone Application", Software Engineering, Business Continuity, and Education, CCIS 257, pp. 292-297, Dec. 2011.
- [9] Woo Yeol Kim, Hyun Seung Son, R. Young Chul Kim, "Rule Extraction Method for Model Transformations in Heterogeneous Smartphone Applications", Information Journal, Vol.16, No.1(B), pp. 615-626, Jan. 2013.
- [10] Obeo, Acceleo User Guide, <http://www.acceleo.org/>
- [11] Woo Yeol Kim, Hyun Seung Son, Robert Young Chul Kim, "Design of Code Template for Automatic Code Generation of Heterogeneous Smartphone Application", Advanced Communication and Networking, Springer CCIS 199, pp. 292-297, Aug. 2011.
- [12] Hyun Seung Son, Woo Yeol Kim, R. Young Chul Kim, "Concretization of the Structural and Behavioral Models based on model Transformation Paradigm for Heterogeneous Mobile Software", International Journal of Software Engineering and Its Applications, Vol. 7, No. 4, pp. 389-399, Jul. 2013.
- [13] Hyun Seung Son, Jae Seung Kim, R. Young Chul Kim, "SMTL Oriented Model Transformation Mechanism for Heterogeneous Smart Mobile Models", International Journal of Software Engineering and Its Applications, Vol.7, No.3, pp. 323-331, May. 2013.
- [14] Jin Lee, Jungsun Kim, "A MDA-based Application Atuhoring Tool Architecture for Smart Cross-Platform", Journal of KIISE : Computer Systems and Theory, Vol. 40, No. 3, pp. 139-148, Jun. 2013.
- [15] Krzysztof Czarnecki, Simon Helsen, "Feature-Based Survey of Model Transformation Approaches", IBM Systems Journal, Vol. 45 No. 3, pp. 621-645, 2006.
- [16] OMG, "MOF 2.0/XMI Mapping, v2.1.1", OMG Available Specification, 2007.
- [17] EcoreTools, <http://www.eclipse.org/ecoretools/>
- [18] Dave Steinberg, Frank Budinsky, Marcelo

- Paternostro, Ed Merks, "EMF Eclipse Modeling Framework", Addison-Wesley, 2009.
- [19] Hyun Seung Son, Woo Yeol Kim, In-geol Chun, Jae Ho Jeon, R. Young Chul kim, "Development of Simulink XMI Translator for ECML of Cyber-Physical Systems", Proceedings of the 15th Koera Conference on Software Engineering (KCSE 2013), Vol. 15, No. 1, pp. 170-175, Jan. 2013.