

GUI 위젯을 이용한 위피 코드 생성 저작도구 설계

이동수*, 박기창*, 김철현*, 이상준**, 김병기*
전남대학교 전자컴퓨터공학과*
전남대학교 경영학부**

e-mail:{tnsqnffn,kcpark,ch-kim,s-lee,bgkim}@chonnam.ac.kr

Design of Authoring Tool for generating WIPI Codes using GUI Widget

Dong-Su Lee*, Ki-Chang Park*, Chul-Hyun Kim*
Sang-Jun Lee**, Byung-Ki Kim*

*Dept. of Electronic & Computer Engineering, Chonnam National University

**College of Business Administration, Chonnam National University

요 약

현재 위피 저작도구는 위피 코드 작성의 편의성이 부족하고, 시각화를 배제한 문법 중심의 프로그래밍 도구가 대부분이다. 이로 인해 규모가 작은 위피 응용프로그램의 경우에도 직접 코딩해서 개발하고 있다. 따라서 위피 상위레벨(High level) 컴포넌트를 GUI 위젯(Widget)으로 시각화하고, 이를 모바일 레이아웃에 쉽게 적용한 후, 위피 코드로 렌더링(Rendering) 해주는 저작도구에 대한 연구가 필요하다. 본 논문에서는 위젯, 기본 위피 코드, 에플레이터 관리를 위해 W2PM(Widget and WIPI Platform Management) 을 제안하고, 이를 시스템 기능 요구사항에 반영하여 GUI 위젯을 이용한 위피 저작도구를 설계한다. 제안한 도구는 개발자가 Look & Feel을 자유롭게 조정하고, 중·소규모의 위피 응용프로그램을 제작하는데 효율적이다.

1. 서론

위피(WIPI : Wireless Internet Platform for Interoperability)의 등장으로 게임, 멀티미디어, 비즈니스 등에 적용되는 다양한 모바일 응용프로그램의 개발이 진행되고 있다[5]. 무선인터넷 콘텐츠의 경우 사용자의 요구사항이 점점 다양해지고 있기 때문에 시장 경쟁력을 가지기 위해서는 새로운 형태의 콘텐츠를 신속하게 저렴한 비용으로 개발할 수 있어야 한다[9].

현재 연구된 위피 저작도구는 위피 코드 작성 시 문법 중심으로 설계되어 있어서 개발사 입장에서는 개발 기간 및 비용이 증가하고, 개발자는 편의성을 제공받지 못하고 있다. 따라서 시각화 중심의 위피 저작도구에 대한 연구가 필요하다.

본 연구에서는 기존 저작도구를 이용하여 위피 응용프로그램을 저작할 때의 문제점을 보완할 수 있는 GUI 위젯을 이용한 위피 코드 생성 저작도구를 분석·설계하고, 이를 기반으로 시스템을 구성한다. 이 시스템은 위피 상위레벨 및 하위레벨 GUI를 위젯으로 시각화하여, 이를 Drag&Drop, Look&Feel 기능을 이용하여 모바일 레이아웃에 빠르게 적용할 수 있고, 레이아웃에 디스플레이된 위젯을 위피 렌더러를 통해 위피 코드로 변환해 준다. 이 시스템을 이용하여 개발자는 위피 응용프로그램을 보다 쉽게 개발할 수 있고, 개발 기간을 단축하는데도 효율적이다.

2. 관련연구

2.1 GUI 위젯

오늘날 상업적 목적의 어플리케이션들을 사용할 때, 사람들은 어플리케이션 기능들을 불러오고 피드백을 얻기 위한 방법으로 그래픽 기능을 활용한 컨트롤(control)들에 크게 의존한다[1]. 위젯은 이러한 그래픽 기능을 활용한 컨트롤 들을 가능하게 하는 그래픽 유저 인터페이스(GUI)의 단위 요소로써, 윈도우나 텍스트박스과 같이 사용자와의 상호작용을 위한 인터페이스(interface, panel)로 정의된다. 위젯은 활용되는 플랫폼에 따라 GUI 위젯, 웹 위젯, 모바일 위젯 등으로 구분된다. GUI 위젯은 버튼, 라디오 버튼, 체크 상자, 슬라이더, 텍스트 상자 등이 있고, (그림 1)은 GUI 위젯의 인터페이스이다.

본 논문에서는 GUI 위젯을 이용하여 위피 상위레벨 컴포넌트를 시각화하고, Drag & Drop, Look & Feel 기능을 이용하여 이를 모바일 레이아웃에 빠르게 적용 가능하도록 저작도구를 설계한다.



(그림 1) GUI 위젯 인터페이스

2.2 위피

국내의 서로 다른 모바일 플랫폼(SKVM, BREW, Ez-i) 환경을 극복하고자 이동통신사(SK, KTF, LGT) 간 호환이 가능한 위피 플랫폼이 제정되었다[4]. 위피의 등장으로 게임, 멀티미디어, 비즈니스 등에 적용되는 다양한 모바일 응용프로그램의 개발이 진행되고 있다[5].

위피 플랫폼을 개발하기 위한 응용프로그램에는 C로 작성하는 Clet과 JAVA로 작성하는 Jlet이 있다[4]. 일반적으로 타 플랫폼으로 호환성이 용이한 Jlet으로 개발을 하지만, 최근에는 빠른 처리속도, 응답속도를 위해 Clet으로 개발하는 응용프로그램이 늘어나고 있다.

위피 생명주기, 기본코드 분석, 이동통신사의 위피 API 및 에뮬레이터 분석은 위피 콘텐츠 전용 저작도구[2,3]의 연구에서 기술하였다.

현재 위피 플랫폼은 각 이동통신사마다 API가 다른 위피 플랫폼을 적용하고 있는데, 기본 위피 API에 과금정책, 카메라 기능 등의 확장 API를 플러그인 한 형태로 탑재하고 있다. 따라서 개발자가 각 이동통신사마다 다른 확장 API를 적용하여 각각 개발해야 하므로, 개발시간 및 개발비용이 증가한다. 이를 위해서는 위피 응용프로그램을 빠르게 개발할 수 있고, 시각적으로 위피 컴포넌트를 지원해주는 저작도구에 대한 연구가 필요하다.

2.3 위피 연구 현황

국내의 위피에 관한 연구는 에뮬레이터, 플랫폼 컨버팅(Converting), 코드 제너레이션(Generation) 등이 진행되었다. CNU 위피 에뮬레이터[4]와 GVM-to-WIPI[5]의 연구에서는 다양한 운영체제에서 사용가능한 CNU 위피 에뮬레이터의 구현 및 실행 성능, 안정성에 대해 접근하고 있으며[4], GVM[6]-to-WIPI에서는 플랫폼 컨버팅을 통한 개발비용, 개발시간의 효율성[5]에 대해 연구되었다[2]. 하지만, 연구의 전체적인 관점이 빠른 위피 코드 개발, 각 통신사 에뮬레이터 설정 등의 편의성에 있기 때문에 기존 연구는 이러한 관점을 해결하기에 한계가 있다. 위피 코드 제너레이션[2,3]의 연구에서는 위피 기본 코드를 자동생성해주고, Drag & Drop 기능을 이용하여 위피 API를 쉽게 코드에 적용하며, 마법사 기능을 지원하여 빠른 코드 개발을 하도록 연구되었다. 하지만 시각화를 배제한 문법 중심의 개발로 인해 개발자가 쉽게 접근하기 어렵다. 따라서 위젯을 이용하여 위피 컴포넌트를 시각화 해주고, 이를 모바일 레이아웃에 적용하여 Look & Feel 조정을 통해 위피 코드로 렌더링 해주는 저작도구가 필요하다.

3. 시스템 요구사항 분석 · 설계

3.1. 시스템 요구사항

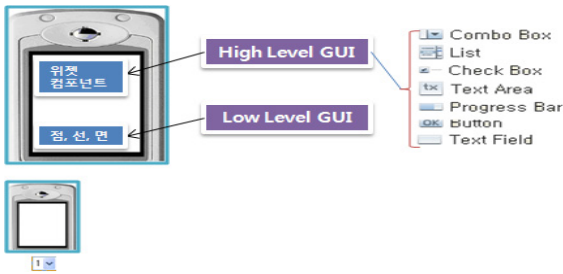
위피 콘텐츠 제작 시 개발자의 편의성을 향상시키고 빠른 위피 응용프로그램 개발을 위해서 다음과 같은 기본 요구사항을 반영해야 한다.

- 첫째, 위피 콘텐츠 제작 시 응용프로그램에 반드시 포함되는 생명주기가 포함된 기본 위피 코드의 자동생성이 용이해야 한다.
- 둘째, 시각화를 배제한 문법 중심의 프로그래밍의 단점을 보완하기 위해 상위레벨 컴포넌트를 GUI 위젯으로 시각화 해야 한다.
- 셋째, 저작도구에서 제공하는 모바일 레이아웃에 구현할 상위레벨 컴포넌트를 마우스 Drag & Drop으로 적용하고, Look & Feel 기능으로 컴포넌트를 리사이즈 할 수 있어야 한다.
- 넷째, 적용한 컴포넌트 및 메소드 속성 값, 리턴 형, 보안레벨 등 설정의 편의성을 제공해야 한다.
- 다섯째, 모바일 레이아웃에 디스플레이 된 GUI를 위피 코드로 렌더링 할 수 있도록 시스템 렌더러를 제공해야 한다.
- 여섯째, 모바일 레이아웃에 구현된 디스플레이를 위피 코드로 렌더링 해주는 변환 규칙을 명세해야 한다.
- 일곱째, 모바일 레이아웃에 적용과 위피 코드로 렌더링 할 때 발생하는 변환 규칙을 UIDL(User Interface Description Language)[6,7,8]로 명세하여 메타데이터를 관리한다. UIDL을 이용하여 저장하기 및 불러오기 등의 프로젝트 관리를 할 수 있어야 한다.
- 여덟째, 제작된 응용프로그램을 각 이동통신사 에뮬레이터 구동 및 연동을 통해서 빠르게 디버깅 작업을 할 수 있어야 한다.

3.2. 모바일 레이아웃 구성요소

위피 플랫폼에는 위피 응용프로그램을 개발하는데 필요한 클래스와 인터페이스가 포함되어 있다. 인터페이스에는 Display, Card, Graphics 클래스 정보가 담긴 org.kwis.msp.lcdui 패키지가 있고, 상위레벨 컴포넌트 객체를 담고 있는 org.kwis.msp.lwc 패키지가 있다. Jlet의 화면 표시를 위해서는 lcdui 패키지내의 Display를 구현한 후에 Card를 생성하고, pushCard 함수를 호출하여 Display에 Card를 담는다. Card의 paint 함수에서 그려지는 내용이 화면에 출력되는데, Graphics 클래스를 이용하여 점, 선, 면 등의 기능을 paint 함수 내부에 작성하여 화면을 출력하는 방식이 하위레벨(Low level) GUI 구성 방식이다. 반대로 lwc 패키지내의 미리 정의된 클래스의 객체를 이용하여 화면을 구성하는 방식이 상위레벨 GUI 구성 방식이다.

저작도구는 상위레벨 및 하위레벨 GUI를 시각화하기 위해서 2개의 모바일 레이아웃을 제공해야 한다. 하나는 편집 및 Look & Feel 기능을 지원하기 위한 레이아웃이고, 다른 하나는 제작한 디스플레이의 각 프레임을 나타내는 레이아웃이다. (그림 2)는 저작도구가 제공한 모바일 레이아웃에 상위레벨 및 하위레벨 GUI를 디스플레이하고, 제작한 디스플레이의 각 프레임 레이아웃을 나타낸다.



(그림 2) 모바일 레이아웃 구성

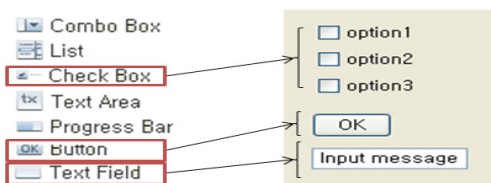
3.3. 위젯 상위레벨 컴포넌트

상위레벨 컴포넌트를 이용하여 라벨, 버튼, 텍스트필드, 체크박스 등 이미 구현된 컴포넌트 객체 클래스를 적절하게 조합, 배치함으로써 화면을 구성할 수 있다. 주요 컴포넌트의 기능과 규칙은 <표 1>에 정의한다.

<표 1> 주요 컴포넌트의 기능과 규칙

ShellComponent 기능 및 규칙
<ul style="list-style-type: none"> 비즈니스용 응용프로그램을 작성할 때에는 가장 먼저 ShellComponent 사용한다. 그 위에 FormComponent를 얹은 후, 다른 컴포넌트들을 배치한다. ShellComponent는 화면에 출력되는 기본 단위가 되는 Card와 연결되는 데, Card 내부에 존재하는 paint를 통해서 컴포넌트들을 화면에 보여준다.
FormComponent 기능 및 규칙
<ul style="list-style-type: none"> 대부분의 컴포넌트들은 FormComponent 클래스 위에 놓여진다. FormComponent는 여러 종류의 컴포넌트들의 위치를 배열하고 스크롤을 관리하여 보여준다.
ButtonComponent 기능 및 규칙
<ul style="list-style-type: none"> 버튼 컴포넌트는 선택한 버튼을 눌렀다가 떼면 액션 처리를 위해 등록된 ActionListener를 호출하게 된다. 버튼 안에 들어갈 수 있는 것은 문자열과 이미지이다.
CheckBoxComponent 와 CheckBoxGroup 기능 및 규칙
<ul style="list-style-type: none"> CheckBoxComponent 클래스는 리스트 항목 중에서 선택 버튼이 있는 리스트 항목을 만드는 데에 사용된다. 항목들을 묶을 때 CheckBoxGroup 클래스를 이용한다.
ListComponent와 ListItemComponent 기능 및 규칙
<ul style="list-style-type: none"> 리스트는 사용자에게 여러 메뉴를 보여 주고, 이중에 하나 또는 여러 개의 항목을 선택할 수 있도록 해 준다.

상위레벨 컴포넌트는 정의된 기능 및 규칙에 의해서 GUI 위젯으로 시각화 한다. (그림 3)은 상위레벨 컴포넌트를 GUI 위젯으로 시각화한 그림이다.



(그림 3) GUI 위젯으로 시각화

3.4. GUI 위젯 렌더링

GUI 위젯을 시스템의 렌더러를 통해 위피 코드로 렌더링하는 것은 위피 응용프로그램을 개발하는데 효율적이다. 위피 코드로 렌더링 하기 위해서 다음과 같은 기본 규칙을 정의한다.

- 첫째, Jlet 구성요소를 위해 필요한 라이브러리 클래스 패키지를 소스 코드에 추가하고, Jlet 클래스로부터 상속받은 자식 클래스를 정의한다. 만약 하나 이상의 이벤트 요소가 존재한다면, 클래스 선언부에 Listener 인터페이스를 구현하는 부분이 추가된다.
- 둘째, Jlet의 생명주기와 연관된 메소드(startApp(), pauseApp(), destroyApp(), resumeApp())를 추가한다.
- 셋째, Jlet 화면을 구성하는 상위레벨 및 하위레벨 GUI를 소스 코드로 변환하고, 위피 기본 코드와 함께 initialize() 메소드에 정의하고, 이 메소드를 다시 startApp() 메소드에서 호출한다.

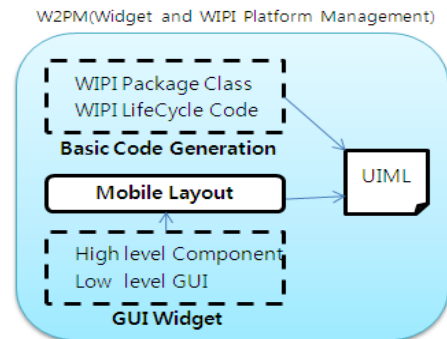
4. GUI 위젯을 적용한 시스템 설계

W2PM의 기능에 대해 알아보고, 시스템 렌더러 프로세스를 정의한다. 이를 토대로 위젯을 이용한 위피 코드 생성 저작도구의 시스템 구성을 제안한다.

4.1 W2PM

UIML(User Interface Markup Language)[6]은 가장 대표적인 UIDL로 자바 AWT, Swing, Visual Basic 등과 같은 다양한 형태의 UI 명세를 위한 메타언어이다.

본 논문에서는 위피 기본 코드 정보 및 모바일 레이아웃에 디스플레이 된 정보를 UIML에 저장하여 관리하는 W2PM을 제안한다. 위피 기본 코드에는 위피 패키지 파일과 위피 응용프로그램에 반드시 들어가야 할 생명주기 코드가 포함된다. 모바일 레이아웃에는 상위레벨 및 하위레벨 GUI 위젯이 디스플레이 된다. (그림 4)는 W2PM의 기능을 나타낸다.

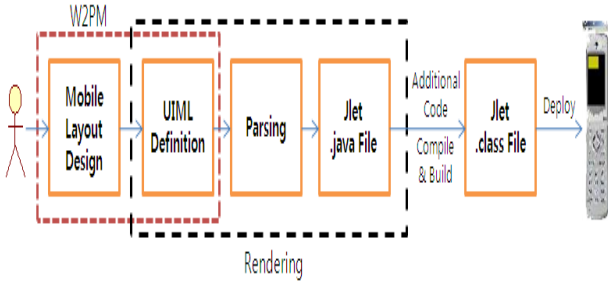


(그림 4) W2PM 기능

4.2. 시스템 렌더러 프로세스

W2PM에 의해서 메타정보가 담긴 UIML이 생성되면,

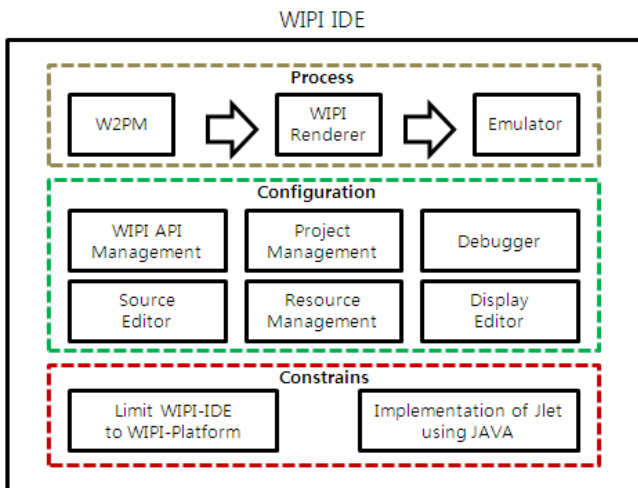
파싱을 통해서 위피 소스 코드로 렌더링 한다. 렌더링 프로세스가 끝나면 추가 코드 작성 및 컴파일을 실행 하고, 에뮬레이터로 구현을 확인 할 수 있다. (그림 5)는 위피 렌더러 프로세서를 나타낸다.



(그림 5) WIPI Renderer 프로세서

4.3. 시스템 구성

GUI 위젯을 이용한 위피 코드 생성 저작도구의 시스템은 (그림 6)과 같이 전체 프로세스, 구성, 제약사항으로 세분화하여 나타내었다. 위피 응용프로그램 개발 프로세스는 다음과 같다. W2PM에서 위피 기본 코드 정보, GUI 위젯 정보가 저장된 UIML을 생성하고, 이를 위피 렌더러를 통해 위피 코드로 렌더링 하여 에뮬레이터에서 디버깅, 구현 및 테스트를 한다. 시스템은 위피 API 관리, 프로젝트 관리, 디버거, 소스 편집기, 리소스 관리, 저작도구 디스플레이 편집기로 구성된다. 저작도구의 제약사항은 위피 플랫폼만을 대상으로 개발할 수 있고, 저작도구를 통해 위피 응용프로그램 개발 시 자바를 사용하여 Jlet을 구현한다.



(그림 6) 시스템 구성

5. 결론 및 향후 연구

위피 응용프로그램 개발자의 편의성 향상과 개발기간 단축 및 개발비용을 줄이기 위해서 위피 전용 저작도구는 꼭 필요한 부분이다. 하지만 기존 저작도구는 시각화를 배제한 문법 중심의 시스템으로써 위피 응용프로그램을 쉽

게 제작하기 어려웠다.

본 논문에서는 이러한 단점을 보완하기 위해서 위피 상위레벨 컴포넌트를 위젯으로 구성하여 시각화하고, 저작도구에서 제공하는 모바일 레이아웃에 위젯을 적용할 수 있으며, 렌더러를 통해서 위피 코드로 렌더링 해주는 시스템을 제안하고 설계하였다. 또한 GUI 위젯을 시스템 렌더러를 통해 위피 코드로 렌더링 하기 위한 몇 가지 규칙을 제안하였다.

제안한 시스템은 위피 응용프로그램 개발 시 문법 중심의 프로그래밍 한계를 보완해주고, 개발자의 편의성을 향상시킬 수 있다.

향후 연구에서는 위피 비즈니스 응용프로그램 개발 시 자주 쓰이는 상위레벨 컴포넌트를 선별하고, 모바일 레이아웃에 적용과 위피 코드로 렌더링 할 때 발생하는 변환 규칙을 UIDL로 명세한 후, 위피 저작도구를 설계·구현한다.

참고문헌

[1] Saul Greenberg and Michael Boyle, "Customizable Physical Interfaces for Interacting with Conventional Applications," Proc.15th UIST2002, 2002.
 [2] 이동수 외 3인, "위피 콘텐츠 개발을 지원하기 위한 통합개발환경," 한국정보처리학회 춘계학술발표대회 제14권 제1호, pp.160-163, 2007.
 [3] 이동수, 김병기, "위피 콘텐츠 전용 저작도구 설계 및 구현," 한국 소프트웨어공학기술 합동 워크샵 2007, pp.72-76, 2007.
 [4] 한국무선인터넷표준화포럼, Available at URL: <http://www.kwisforum.org/>.
 [5] 이상윤 외 2인, "한국 무선 인터넷 표준 플랫폼(WIPI)의 표준화 현황 및 발전 전망," 한국정보과학회 학회지 제22권 제1호, pp.16-23, 2004.
 [6] M. Abrams and C. Phanouriou, "UIML: An XML Language for Building Device-Independent User Interface," XML'99, 1999.
 [7] Abstract User Interface Markup Language Toolkit, <http://www.alphaworks.ibm.com/tech/auiml>.
 [8] XIML Forum, <http://www.ximl.org>.
 [9] 유용덕 외 3인, "위피 응용프로그램 개발환경 설계 및 구현," 한국정보처리학회 논문지 C, pp.749-756, 2005.