

UIML 을 이용한 J2ME Midlet UI 모델링

박승범*, 박기창*, 김철현*, 이재희*, 김병기*

*전남대학교 전산학과

e-mail:sbpark@chonnam.ac.kr

J2ME Midlet UI Modeling using UIML

Seung-Bum Park*, Ki-Chang Park*, Chul-Hyun Kim*, Jea-Hee Lee*, Byung-Ki Kim*
*Dept of Computer Science, Chonnam National University

요 약

UIML 은 다양한 장치에서 사용되는 UI 구성을 위한 XML 기반의 추상 메타 언어로 다양한 Renderer 를 통해 각 장치에 맞는 UI 를 생산해 낸다. 기존의 Renderer 는 C++, HTML, WML 등의 다양한 환경을 지원하지만 J2ME 기반의 UI 를 지원하는 Renderer 는 등장하지 않았다. 본 논문에서는 J2ME 기반의 UI 구성요소를 분석하고 UIML 명세로부터 J2ME 의 UI 를 생성하는 방안에 대하여 연구하였다. J2ME 의 UI 를 명세하기 위한 Vocabulary 를 제안하고, 이를 UIML 과 J2ME UI 요소간의 맵핑에 사용하여 J2ME 플랫폼의 UI 를 UIML 로부터 생성해 낼 수 있다.

1. 서론

잘 설계된 사용자 인터페이스는 측정과 분석이 어려운 consistency 와 style 과 같은 품질 특성을 갖는다.[2] 게다가 이러한 품질 특성은 다양한 장치에서 다양한 기술을 통해 구현된다. 폰을 위한 WML, 사람의 목소리를 위한 SpeechML, VoxML, PC 를 위한 HTML 등 다양하다[1]. 하지만 이러한 서로 다른 장치에서 사용자 인터페이스를 구성할 때에는 다음과 같은 문제점이 발생한다. 첫째, 각 장치에서 상이한 기술을 통해 사용자 인터페이스가 구현됨으로써 여러 장치에 대한 각각의 코드를 생성해 내야 한다. 둘째, PC 와 모바일 폰과 같이 사용자 인터페이스 환경이 심하게 다른 경우 개발자는 인터페이스 요소를 각각 고려해야 한다. 셋째, 개발자는 사라져 버릴 지도 모를 특정 장치에 대한 인터페이스를 작성하는 위험부담을 항상 갖게 된다[1]. 이러한 다양한 장치에서의 인터페이스 구현 기술에 대한 문제점을 해결하기 위해 다양한 플랫폼에서 사용이 가능한 사용자 인터페이스에 관한 연구가 진행되었다[4][5][6][7]. UIML(User Interface Modeling Language)은 Harmonia 에서 장치 독립적인 사용자 인터페이스를 개발하기 위해 만든 XML 기반의 meta-language 이다.[3] UIML 을 이용하

여 다양한 플랫폼에서 사용할 수 있는 사용자 인터페이스를 정의할 수 있다.

UIML 문서를 이용하여 실제 화면에 표현되는 인터페이스를 나타내는 과정을 Rendering 이라고 하는데 이러한 Rendering 기법에는 UIML 을 WML 이나 Java 와 같은 언어로 번역하는 Compiling 방식과 UIML 문서를 읽고 UI 를 표현 하거나 interaction 을 허용하는 API 를 호출하는 방식의 interpreting 방식이 있다. 현재 다양한 UIML Rendering Engine 이 개발되었다.[8] 하지만 모바일 응용 프로그램을 위한 Rendering Engine 은 WML 로 한정되어 그 활용범위가 다른 플랫폼에 비해 좁은 편이다. 따라서 본 논문에서는 모바일 플랫폼으로 많이 사용되는 J2ME 기반의 UIML Rendering Engine 을 개발하기 위해 UIML 을 통해 모델링 된 추상적인 인터페이스를 J2ME UI 의 구성요소에 적합한 vocabulary[9][10]를 제안하고 이를 통해 Midlet UI 를 UIML 로 모델링 하는 방안을 제시한다. 또한 Rendering Engine 을 구현함으로써 다양한 J2ME UI 를 생성해 내는 Midlet Renderer 를 설계하였다.

2. UIML Document Structure

UIML 은 하나의 인터페이스 명세를 다양한 장치에서 구현하기 위해 Harmonia 에서 개발한 XML 기반 인터페이스 모델링 언어이다. UIML 을 통하여 장치 독립적인 인터페이스를 명세할 수 있다. (그림 1)은 UIML 명세로부터 각 각 PC, Handheld PC, Cell Phone 에서 Rendering 된 예이다.[1]



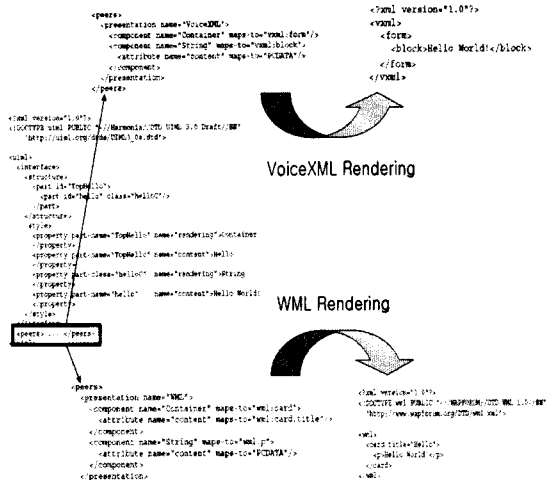
(그림 1) 다양한 장치의 사용자 인터페이스

UIML document 의 root element 는 <uiml>이다. uiml element 는 metadata 정보를 제공하는 <head>, reuse 를 위한 <template>, UI 를 구성하는 <interface>, application 과의 mapping 을 표현하는 <peers>등 4 개의 child element 를 갖는다. 주요 Skeleton Code 는 Fig. 2 와 같다.

```
<?xml version="1.0" >
<uiml>
  <interface>
    <structure>..</structure>
    <style>..</style>
    <content>..</behavior>
    <behavior>..</behavior>
  </interface>
  <peers>
    <logic>..</logic>
    <presentation>..</presentation>
  </peers>
</uiml>
```

(그림 2) UIML 의 구조

<interface>는 <structure>, <style>, <content>, <behavior>등 4 개의 element 를 포함한다. <structure> element 는 다양한 플랫폼을 위한 UI 구성 요소를 표현하고, <style> element 는 Interface 구성요소와 관련된 다양한 properties 를 정의한다. <content> element 는 interface parts 와 연관된 words, sounds 그리고 images 를 제공하며, <behavior> element 는 UI events 를 정의한다. <peers> element 는 특정 플랫폼에 대해 작성 가능하며 Rendering 을 통해 원하는 UI 문서를 얻을 수 있다.[2]

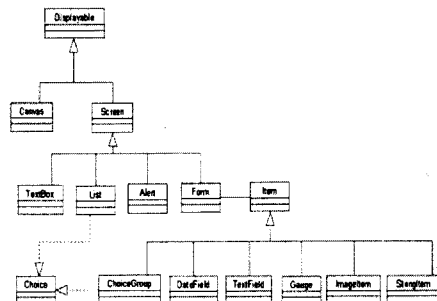


(그림 3) Rendering 의 예

3. UIML 을 이용한 J2ME MIDlet UI 모델링

J2ME 플랫폼에서 기존의 AWT 나 Swing 과 같은 UI 를 사용하는 것은 장치 특성상 많은 제약이 따른다. 결과적으로 기존의 AWT 와 유사한 lcdui 라는 사용자 인터페이스가 설계 되었다. Midlet 에서 사용하는 GUI 클래스들은 javax.microedition.lcdui 패키지에 정의되어 있다. lcdui 패키지에 정의된 클래스 사이의 관계는 (그림 4)와 같은 Class Diagram 으로 표현 할 수 있다.

Midlet 에서 UI 를 구성하는 방법은 화면에 표시되는 장치는 Displayable 클래스로부터 상속받은 Canvas 클래스와 Screen 클래스로 추상화 되어있다. Canvas 클래스는 다양한 그래픽 요소를 이용한 그리기 작업을 할 수 있는 유일한 클래스이다. Screen 클래스는 TextBox, List, Alert, Form 과 같은 객체들을 하나로 묶는 역할을 한다. Item 클래스로부터 상속 받은 ChoiceGroup, DateField, TextField 와 같은 클래스는 독립적으로 화면에 보여질 수 없고 Form 클래스에 포함되어 화면에 표현된다.



(그림 4) lcdui 패키지 클래스 다이어그램

UIML 에 등장하는 모델링 요소를 특정 플랫폼기반의 UI 로 변경하기 위한 변환 프로세스를 구성하기 위해서는 기존의 Vocabulary[12, 14]와 대응되는 J2ME UI 를 <표 1>과 같이 구성하였다.

<표 1> Vocabulary

Generic User Interface Element	Java (SWING)	Palm OS	WML (tag)	J2ME
GButton	JButton	Command Button	<do>	Command
GLabel	JLabel	Label	<p>	StringItem
GIcon	Icon	Bitmap		ImageItem
GSLTestRegion	JTextField	Field	<input>	TextField
GList	JList	Popup List	<select>	List
GDialog	JDialog	Alert Dialog	-	Alert
GMenu	JMenu	Menu	-	-
GArea	JFrame Window etc	Window	<card>	Screen Card
GTopContainer	JFrame	Form	<wml>	Displayable
GMLTextRegion	JTextArea	Field	<textarea>	TextField
GCheckbox	JCheckBox	Checkbox	<input type="checkbox">	List
GRadioButton	JRadioButton	Pushbutton	<input type="radio">	List

<표 2> UI 클래스의 속성과 값의 예

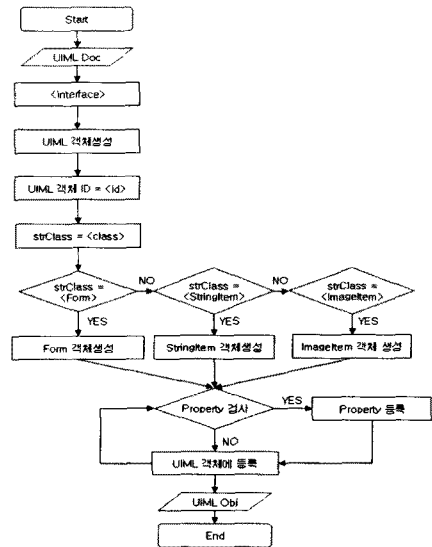
클래스	속성	값
Alert	title	문자열
	alertText	문자열
	alertImage	이미지 파일경로, URL
	alertType	ALAM, CONFIRMATION, INFO, WARNING
ChoiceGroup	label	문자열
	choiceType	EXCLUSIVE, MULTIPLE, IMPLICIT, POPUP
Command	longLabel	문자열
	shortLabel	문자열
	commandType	BACK, EXIT, HELP, ITEM, OK, SCREEN, STOP
	priority	양의 정수
DateField	label	문자열
	mode	DATE, DATE_TIME, TIME
	timeZone	Time Zone ID
Display	imageType	ALERT, CHOICE_GROUP, LIST

	color	BACKGROUND, BORDER, FOREGROUND, HIGHLIGHTED_BACKGROUND, HIGHLIGHTED_BORDER, HIGHLIGHTED_FOREGROUND
	vibrate	양의 정수
Displayable	title	문자열
	visible	True 또는 False
Font	face	MONOSPACE, PROPORTIONAL, SYSTEM
	size	LARGE, MEDIUM, SMALL
	style	PLAIN, BOLD, ITALIC, UNDERLINED
	specifier	STATIC, INPUT

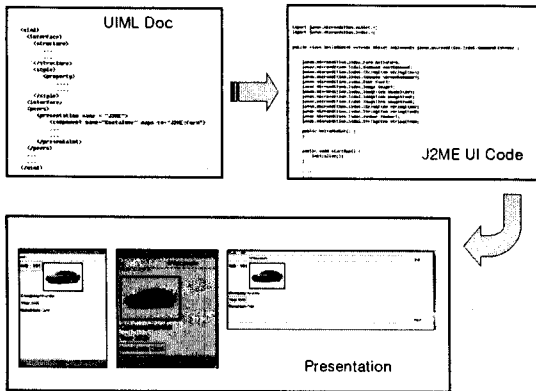
<표 2>에서 제시한 클래스, 속성, 값을 사용하여 명세된 UIML 문서로부터 실제 MIDlet UI 를 생성하기 위한 알고리즘은 (그림 5)와 같이 표현할 수 있다. UIML 문서로부터 UI 객체를 추출한 후 객체식별, 속성 식별, 값 적용등의 단계를 거쳐 MIDlet UI 객체와 대응하게 된다.

4. Rendering

UIML 문서에서 J2ME UI 로 Rendering 하기 위해 3장에서 제시한 클래스와 클래스의 속성과 값을 사용하여 UIML 을 명세한다. J2ME Render 는 UIML 문서를 입력받아 <peers> element 를 분석하여 J2ME UI 와 mapping 하여 코드를 생성해 낸다.



(그림 5)Rendering 과정 순서도



(그림 6) Rendering 과정

5. 결론 및 향후연구

UIML 은 사용자 인터페이스를 명세하는데 유용하게 사용할 수 있다. 현재 다양한 플랫폼을 위한 Vocabulary 와 Renderer 가 개발되었다. 본 논문에서는 모바일 플랫폼의 일종인 J2ME 의 MIDlet UI 를 UIML 로 명세하기 위한 Vocabulary 를 제안하고 이를 이용한 Rendering 과정을 제시하였다. 이를 통해 기존의 개발자 중심의 코딩이 아닌 명세 기반의 UI 를 얻을 수 있다. 향후 연구과제로 MIDlet UI 의 이벤트 관련 명세를 추가하고 이를 이용한 화면 전환을 자동화하는 방향에 대한 연구가 필요하다.

참고문헌

- [1] Abrams, M., Phanouriou, C., Batongbacal, A., Williams, S., Shuster, J.: UIML: An Appliance-Independent XML User Interface Language, Proceedings of the 8th WWW conference, 1999.
- [2] User Interface Markup Language(UIML) Specification, Harmonia, 2002
- [3] Guruduth B., James B., Eugene G., Jonathan M., Jeremy S., Deborra Z.: Challenges: An Application Model for Pervasive Computing, ACM/IEEE International Conference on Mobile Computing and Networking, 2000
- [4] Silvia B., Giulio M., Fabio P., Carmen S.: A Transformation-based Environment for Designing Multi-Device Interactive Applications, ACM IUT'04, pages 352-353, January 2004.
- [5] Josef J., Pavel., S.: GUI Generation from Annotated Source Code, PAMODIA'04, ACM TAMODIA'04, pages 129-136, November 2004
- [6] Stina N., Markus B., Annika W.: Ubiquitous service access through adapted user interfaces on multiple devices, Springer-Verlag London Limited 2005
- [7] Yves G., Lawrence D. B., Tessa L.: MORE for Less: Model Recovery from Visual Interfaces for Multi-Device Application Design, IUI'03
- [8] UIML Tools, <http://www.uiml.org/tools>
- [9] Rainer S., Michael J., Florian W.: A Generic UIML Vocabulary for Device-and Modality Independent User Interfaces, ACM WWW 2004, pages 434-435, May 2004
- [10] Mir F., Mark A.: Simplifying Construction of Multi-Platform User Interface Using UIML, Harmonia. Inc.
- [11] Qusay H. Mahmoud: MIDP GUI Programming: Programming the Phone Interface, <http://developers.sun.com/techtopics/mobility/midp/articles/ui/>, December 2000
- [12] Michael Juntao Yuan: Enterprise J2ME DEVELOPING MOBILE JAVA APPLICATION, Prentice Hall PTR, 2004
- [13] Jonathan E. Shuster: Introduction the User Interface Markup Language, The Journal of Defense Software Engineering, pages 15-19, January 2005