

UIML에 기반한 통합 사용자 인터페이스 변환 모델에 관한 연구

박병철*, 손민우, 김강, 신동일, 신동규

세종대학교 컴퓨터공학과

e-mail: {leon, minwoo15, kimkang, dshin, shindk}@gce.sejong.ac.kr

A Study on Integrated User Interface transfer model base on UIML

Byung-Chul Park*, Min-Woo Son, Kang Kim, Dong-Il Shin,
Dong-Kyoo Shin

Dept of Computer Engineering, Sejong University

요 약

오늘날 스마트 홈, 홈오토메이션, 홈 네트워크 등의 연구 개발로 여러 기기들과 컴퓨터가 연동한다. 이에 따라 다양한 기기에 맞춰 사용자 인터페이스를 여러 번 개발해야하는 비효율적인 일이 빈번하다. 또한, 개발자에게 있어서 여러 기기에 사용되는 서로 다른 언어를 모두 익히고 개발하는 것 역시 많은 부담을 준다. 이러한 불합리함을 위해 새로운 markup language가 제시되었는데 바로 UIML(User Interface Markup Language)이다. UIML은 XML-compliant 언어로 여러 기기를 위한 사용자 인터페이스를 하나의 문서로서 구현이 가능하다. 따라서 개발자는 UIML 문서 하나만을 개발하면 HTML, WML, VoiceXML 등 다른 언어로 변환이 용이하다. 그러나 UIML도 여러 언어로의 변환을 위해 각각의 문서를 생성해야하는 번거로움이 있다. 본 연구에서는 이러한 UIML을 보완하여 통합된 사용자 인터페이스 변환의 모델을 제시한다.

1. 서론

오늘날 인터넷 도구들은 과거에는 거의 이용되지 못하였던 터치스크린, 음성/필체 인식, 음성 합성기, 소형 스크린 등 사용자 인터페이스 기술을 지원한다. 또한 스마트 홈, 홈오토메이션, 홈 네트워크 등으로 여러 가전기기들과 컴퓨터가 연동한다. 그러나 이러한 기기에서의 기술 혁신이 새로운 문제를 야기한다.

첫째, 여러 다른 기기들은 서로 다른 언어를 사용한다. PC나 Laptop에서는 HTML, 핸드폰에서는 WML(Wireless Markup Language)[1], 전화기와 같이 음성을 이용하는 기기에서는 VoiceXML[2]을 사용한다. 이에 개발자는 다양한 기기에 한 가지 정보에 해당하는 인터페이스를 전개하기 위해 여러 가지의 소스 코드를 개발해나가야 한다. 둘째, 사용자 인터페이스는 각각의 기기에 있어서 상이하게 다르고 개발자는 이러한 인터페이스들을 유지 보수해야 한다. 셋째, 개발자는 기기 사용자를 위해 각각의 기기

인터페이스에 대한 명세서를 작성해야한다.

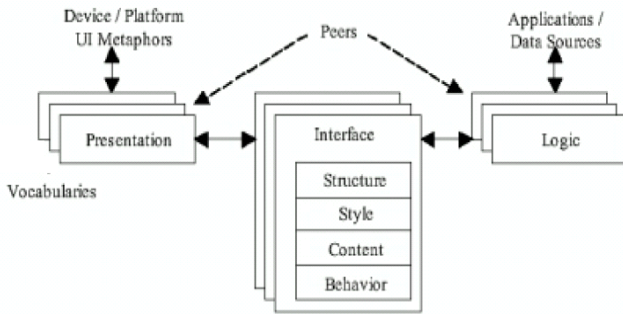
이와 같은 비효율성의 해결책은 여러 기기와 인터페이스에 대응할 수 있는 하나의 보편적인 언어를 만들어 사용하는 것이다. 그리고 이에 해당하는 언어가 UIML(User Interface Markup Language)[3]이다. UIML은 인터페이스 개발자를 여러 다른 기종간의 특성으로부터 차단시켜준다[4]. 그러나 UIML도 여러 언어로 변환을 하기 위해서 그에 상응하는 문서를 각각 작성해야하는 번거로움이 있다. 따라서 본 논문에서는 UIML을 기반으로 한 통합 사용자 인터페이스 변환 모델을 제시한다.

2. 관련연구

2.1. UIML(User Interface Markup Language)의 개요

UIML은 Harmonia 사에서 여러 기기를 위한 사용자 인터페이스를 손쉽게 개발하기 위해 제안한 XML-compliant 언어이며, 2004년 3월 현재 OASIS (Organization for the Advancement of Structured

Information Standards)에서 표준화가 진행 중이다. UIML은 하나의 범용적인 언어를 통해 여러 기기들의 사용자 인터페이스를 구축하여 개발기간의 단축, 원활한 유지, 보수를 목표로 한다.



<그림 1> UIML의 메타-인터페이스 모델

(그림 1)은 인터페이스의 실제 구조와 내용을 담당하는 interface, 사용자 인터페이스를 변환할 언어의 태그를 포함하는 presentation, 인터페이스를 구현하는 어플리케이션의 논리구조를 담당하는 logic으로 구성된 UIML의 메타-인터페이스 모델을 보여준다.

<표 1>은 UIML문서의 기본 구조를 보여주는데 XML의 기본 문법을 따르며 루트 요소로 <uiml>태그를 사용한다. 그리고 자식 요소로 문서의 메타정보를 포함하는 <head>, 인터페이스의 구성요소를 포함하는 <interface>, 재사용을 가능하게 해주는 <template>, 손쉬운 확장을 담당하는 <peers>를 포함한다.

```
<?xml version="1.0"?>
<!DOCTYPE uiml PUBLIC "-//Harmonia/DTD UIML 3.0 Draft/EN" "http://uiml.org/dtds/UIML3_0a.dtd">
<uiml xmlns="http://uiml.org/dtds/UIML3_0a.dtd">
  <head> ... </head>
  <template> ... </template>
  <interface> ... </interface>
  <peers> ... </peers>
</uiml>
```

<표 1> UIML 문서의 기본구조

2.2. UIML 변환예제

다음은 UIML 문서를 WML로 변환하는 예이다. <표 2>는 사용자 인터페이스의 내용을 담고 있는 UIML 문서이고 <peers>태그에서 변환될 WML의 태그를 정의한다. <표 3>과 같은 <peers>태그의 내용이 오면 변환기에 의해 <표 4>의 WML 문서를 결과로 출력한다. 같은 방법으로 <peers>태그 안에 변환할 태그를 정의해주면 다른 언어로의 변환도 가

능하다.

```
<?xml version="1.0"?>
<!DOCTYPE uiml PUBLIC "-//Harmonia/DTD UIML 3.0 Draft/EN" "http://uiml.org/dtds/UIML3_0a.dtd">
<uiml>
  <interface>
    <structure>
      <part id="TopHello">
        <part id="hello" class="helloC"/>
      </part>
    </structure>
    <style>
      <property part-name="TopHello" name="rendering">Container </property>
      <property part-name="TopHello" name="content">Hello </property>
      <property part-class="helloC" name="rendering">String </property>
      <property part-name="hello" name="content">Hello World! </property>
    </style>
  </interface>
  <peers> ... </peers>
</uiml>
```

<표 2> 변환될 UIML 문서

```
<peers>
  <presentation name="WML">
    <component name="Container" maps-to="wml:card">
      <attribute name="content" maps-to="wml:card.title"/>
    </component>
    <component name="String" maps-to="wml:p">
      <attribute name="content" maps-to="PCDATA"/>
    </component>
  </presentation>
</peers>
```

<표 3> WML 태그를 정의하는 <peers>태그

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.0/EN" "http://www.wapforum.org/DTD/wml.xml">
<wml>
  <card title="Hello">
    <p>Hello World!</p>
  </card>
</wml>
```

<표 4> 변환된 WML 문서

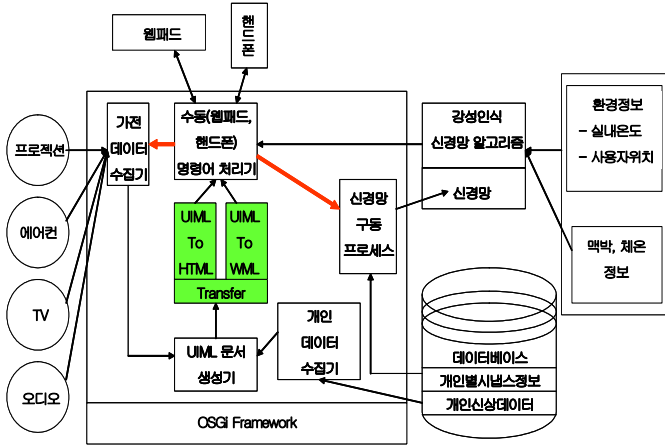
2.3. UIML의 보완해야 할 점

앞선 예와 같이 UIML은 동일한 정보를 나타내는 사용자 인터페이스를 여러 언어로 변환할 수 있다. 그러나 <peers>태그 안에서 변환될 WML의 태

그들이 정의되어야만 한다. 비록 UIML 문서 하나로 여러 변환될 사용자 인터페이스 구축이 가능하지만 결국 변환될 언어의 태그로 매핑하는 작업이 필요하다. 이는 개발자에게 다양한 언어의 습득을 요한다.

3. 통합 사용자 인터페이스 변환기

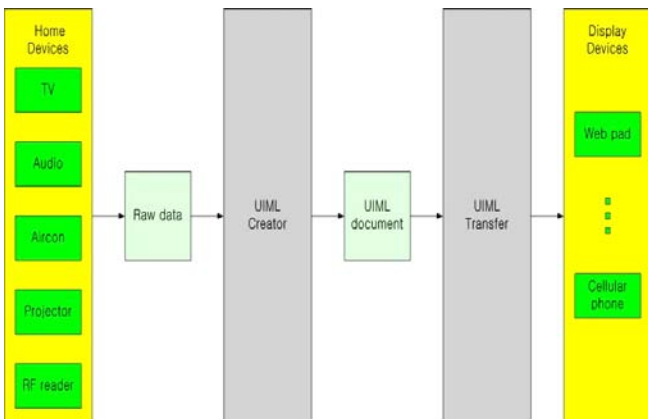
3.1. 통합 사용자 인터페이스 변환기 적용 시스템



(그림 2) 스마트 홈 아키텍처

(그림 2)는 스마트 홈[5]의 아키텍처이다. 스마트 홈 시스템은 사용자의 감성을 인식하여 사용자의 편의에 맞추어 반응하는 시스템이다. 여기서 통합 사용자 인터페이스 변환기의 역할은 가전기기들로부터의 상태 정보를 웹패드와 핸드폰으로 출력할 수 있게 사용자 인터페이스를 구성하는 것이다.

3.2. 통합 사용자 인터페이스 변환기 아키텍처



(그림 3) 통합 사용자 인터페이스 변환기 아키텍처

통합 사용자 인터페이스 변환기는 (그림 3)과 같은 아키텍처를 구성하며 가전기기들로부터의 상태 정보인 raw data를 사용자가 쉽게 확인해 볼 수 있는 웹패드나 핸드폰에 출력한다. 통합 사용자 인터페이스 변환기는 두 부분으로 구성되는데 UIML 문서를 생성하는 Creator와 UIML 문서를 원하는 언어로 변

환해주는 Transfer이다. 변환기의 순차적 처리과정을 정리하면 다음과 같다:

1. 가전기기로부터 raw data 생성
2. raw data를 기반으로 UIML 문서 생성
3. UIML 문서를 웹패드(HTML)나 휴대폰(WML)에 적합하게 변환
4. 변환된 정보를 영상기기 화면에 출력

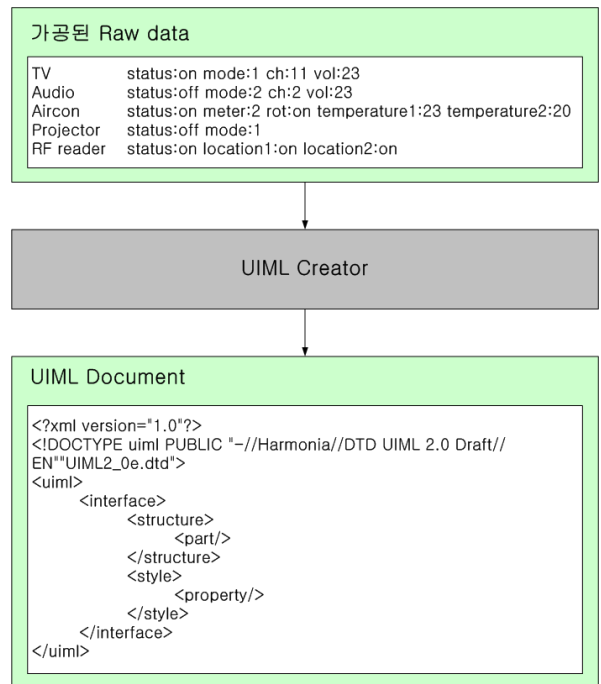
위와 같은 과정을 통해 각각의 영상기기를 위한 사용자 인터페이스 변환이 이루어지며, 이를 위해 다음 세 가지 사항이 선행되어야 한다:

1. raw data를 일관된 형식에 맞게 가공
2. UIML template 구축
3. transfer vocabulary 설정

가전기기로부터의 상태정보인 raw data의 형식은 기기마다 제각각이므로 모든 사용자 인터페이스에 대응하는 단일한 UIML 문서를 생성하기 위해서는 일관된 형식으로 변환이 필요하며 이는 (그림 2)의 가전 데이터 수집기에서 수행한다. 또한 UIML 문서도 모든 사용자 인터페이스로 변환이 용이한 형식의 template을 토대로 생성한다. 마지막으로 기존 <peers>에서 변환될 언어의 태그로 매핑하는 transfer vocabulary 설정이 필요하다.

4. 통합 사용자 인터페이스 변환기 프로세스

4.1 통합 문서 생성기(UIML Creator)

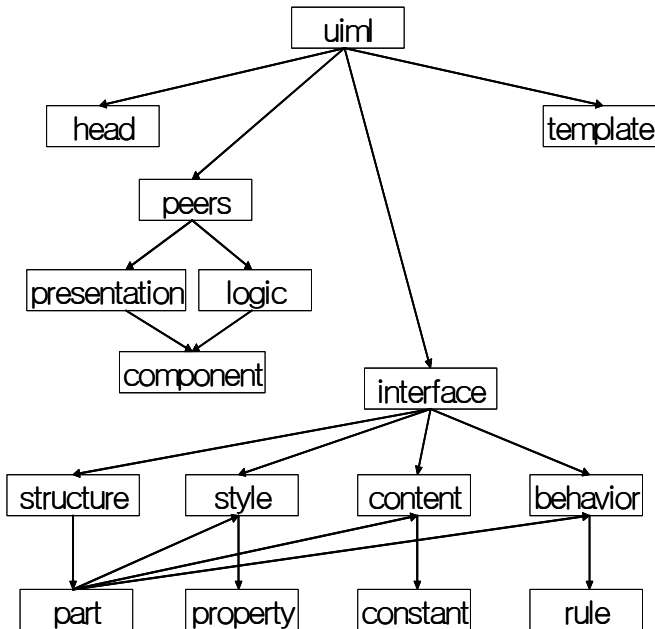


(그림 4) 통합 문서 생성기 프로세스

(그림 4)는 통합 문서 생성기가 가공된 raw data

를 통하여 UIML문서를 생성하는 과정을 보여준다. 생성된 UIML문서는 여러 가지 사용자 인터페이스로 변환이 용이한 template를 기본으로 삼는다.

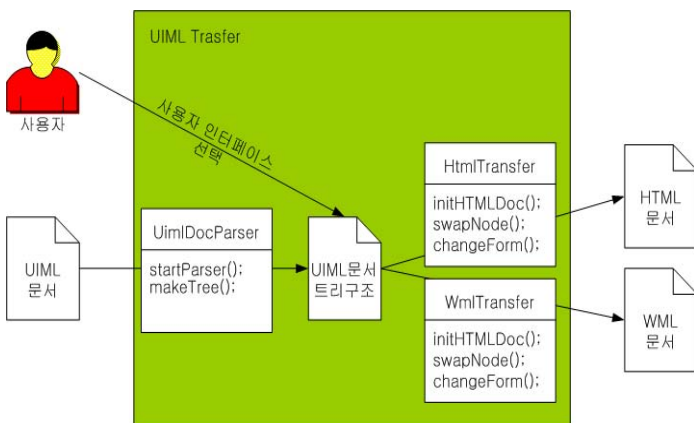
4.2 통합 문서 변환기(UIML Transfer)



(그림 5) uiml 문서 구조 트리

통합 문서 생성기로 생성된 UIML문서를 파서를 통하여 파싱한 결과는 (그림 5)와 같은 트리구조를 갖는다. 통합 문서 변환기는 이 트리구조에 사용자가 원하는 기기에 알맞은 언어로 변환을 담당한다. (그림 6)은 통합 문서 변환기의 구조를 보여주며 다음과 같은 프로세스 단계를 갖는다:

1. 사용자가 원하는 사용자인터페이스 선택
2. UIML문서 파싱
3. 파싱된 트리구조의 문서에 사용자 인터페이스를 구현하는 언어의 태그를 매핑



(그림 6) 통합 문서 변환기의 구조

5. 결론 및 향후 연구과제

현재 스마트 홈이나 홈 네트워크 등 지능을 갖춘 집의 개발이 한창이다. 이러한 스마트 홈에서는 여러 가전제품을 포함 기기들이 연동되지만 기기들과 사용자를 연결해주는 사용자 인터페이스가 각각의 기기마다 따로 개발되어야 한다. 이는 개발기간이 길어지고 사용자 인터페이스의 유지, 보수에도 많은 비용을 초래하며 개발자에게도 여러 언어의 습득을 요구한다. 본 논문에서는 UIML을 기반으로 한 통합 UIML 변환 모델을 제시하여 해법을 모색하였다. 통합 UIML문서를 생성함으로써 여러 다른 언어로 구현되는 사용자 인터페이스의 구축이 가능하다.

향후 연구과제로는 현재까지 연구된 통합 변환 모델을 토대로 통합 사용자 인터페이스 변환기의 프로세스 검증을 통한 성능 향상이 필요하겠다.

참고문헌

[1] WML(Wireless Markup Language), <http://www1.wapforum.org/tech/terms.asp?doc=WAP-191-WML-20000219-a.pdf>

[2] VoiceXML(Voice eXtensible Markup Language), <http://www.voicexml.org/specs/VoiceXML-100.pdf>

[3] UIML(User Interface Markup Language), <http://www.w3.org/TR/2003/CR-voicexml20-20030128/>

[4] M. Abrams; C. Phanouriou; A. L. Batongbacal; S. M. Williams; J. E. Shuster; "UIML: An appliance-independent XML user interface language". Computer Networks, vol. 31, Amsterdam: Elsevier Science, pp. 1695 - 1708, 1999

[5] D. Valtchev; I. Frankov; "Service gateway architecture for a smart home", Communications Magazine, IEEE, Volume: 40, Issue: 4, Pages:126 - 132, April 2002.