

# 모바일 단말정보 저장소 시스템 설계 및 구현

이영일\* · 인민교\*\* · 이강찬\*\* · 이승윤\*\* · 김창수\*\*\* · 정희경\*

\*배재대학교 컴퓨터공학과 · \*\*한국전자통신연구원 · \*\*\*청운대학교 인터넷학과

## Design and Implementation of Mobile Device Description Repository System

Young-il Lee\* · Min-Kyo In\*\* · Kang-Chan Lee\*\* · Seung-Yun Lee\*\*

· Chang-Su Kim\*\*\* · Hoe-Kyung Jung\*

\*Dept. of Computer Engineering, Paichai University · \*\*ETRI

· \*\*\*Dept. of Internet, Chungwoon University

E-mail : \*{lyibest, hkjung}@pcu.ac.kr, \*\*{mkin, chan, syl}@etri.re.kr, \*\*\*ddoja@chungwoon.ac.kr

### 요 약

모바일 단말의 성능 향상으로 다양한 콘텐츠 서비스에 대한 요구가 증대되고 있다. 하지만 이기종 모바일 단말간의 콘텐츠 표현성능이 하드웨어나 소프트웨어 등의 측면에서 차이가 있어 단말에 종속적인 콘텐츠 개발을 강요하게 된다.

이를 해결하기 위해서는 콘텐츠를 단말의 성능에 맞추어 재구성하는 적응화(Adaptation) 과정이 필요하다. 이때 적응화 과정에서 사용되는 단말의 성능 정보를 단말정보 표현 언어를 통해 기술하는데, 이 단말정보 데이터를 저장하고 콘텐츠 제공 서버(Content Provider)에 제공해 주는 역할을 하는 단말정보 저장소 시스템(Device Description Repository)의 부재 및 단말정보 데이터의 확보는 여전히 문제로 남아있다.

이에 본 논문에서는 단말정보 데이터를 확보하기 위해 단말정보의 변환 방법을 제시하고 단말정보 저장소 시스템에 접근하기 위한 표준 인터페이스를 확장하여 정의하였으며, 이를 기반으로 단말정보 저장소 시스템을 설계 및 구현하였다.

### ABSTRACT

Due to the improvement of mobile device capabilities, a demand for various contents service is increasing. But the differences from heterogeneous mobile devices force to develop contents which is dependent on a mobile.

To solve this problem, adaptation course that reorganize contents by the device capability is needed. In this course, device description is described using device description language but there are some problems that is absence of device description repository and the way to get device description data.

In this paper, I suggest device description acquisition module and define the interfaces to access device description repository that I designed and implemented.

### 키워드

DDR, 단말정보 저장소, 모바일

## 1. 서 론

모바일 단말을 이용한 무선 통신에서 웹을 사용하기 위한 노력이 최근 각 표준화 단체 및 사설 업체에서도 활발히 진행 중 이다. 이러한 노력

의 중심에는 모바일 단말의 성능 및 사용자의 선호도 정보 등, 모바일 단말을 통해 콘텐츠를 제공받기 위해 필수적인 정보를 표준화된 방법을 통해 정의하고, 이를 여러 콘텐츠 제공자(Content Provider)가 공유하여 사용자 단말에 최적화된 서

비스를 제공하는데 있다.

먼저 각 모바일 단말을 식별하고 세부 성능정보를 확보하기 위해 단말정보 표현언어(DDL : Device Description Language)가 발표되었다. 이를 통해 단말정보를 기술하고 참조할 수 있는 기반이 마련된 것이다.

하지만 생성된 단말정보 데이터를 저장하고 관리하며, 콘텐츠 제공자 등 단말정보 요청자의 요구에 따라 단말정보를 검색하여 전송하는 서비스를 제공하는 단말정보 저장소(DDR : Device Description Repository)가 필요하게 된다.

이에 본 논문에서는 W3C의 DDR Simple API 표준에 기반을 둔 인터페이스를 제공하는 단말정보 저장소를 설계 및 구현하였다. 또한 단말정보 데이터를 확보하기 위한 방안이자 공존하는 단말정보 표현언어간의 상호 호환성 확보를 위해 단말정보의 변환 방법을 제시한다.

## II. 관련 연구

### 2.1 단말정보 표현언어 표준

단말정보 표현언어는 상이한 단말의 성능 및 사용자의 선호도를 기술하는 언어를 표준으로 정의한 것으로 단말정보의 표현 기술간의 상호 운용성의 부재를 해결하기 위해서 개발되었으며, 대표적인 단말정보 표현 언어로는 UAProf(User Agent Profile)과 WURFL(Wireless Universal Resource File)이 있다. UAProf은 관련된 단말의 속성(Property)정보를 5가지 컴포넌트로 나누어 기술하고 있으며 WURFL은 여러 개의 그룹단위로 기술하고 있다. 그러나 이 두 표현언어 간에는 속성 이름 및 값, 값의 범위가 서로 일치하지 않아 서로 호환을 이루지는 못한다[1,2,3].

### 2.2 단말정보 저장소 요구사항

단말정보 저장소에서 제공해야 하는 기본적인 서비스 및 기능적 요구사항을 제시한다[4].

- CP로부터 전달받은 delivery context를 인식
- 주어진 요청을 이용해 단말정보를 검색
- 단말정보를 검색할 수 있는 API를 제공
- 단말정보의 반환 값에 맞는 단위를 제공
- 예외처리가 가능
- 일정 범위의 검색 지원
- 전달받은 검색 키 값이 불충분 할 경우
  - 주어진 정보에 가장 근접한 결과를 반환
  - 검색되는 모든 값을 반환
  - 예외처리로 종료

### 2.3 DDR Simple API

W3C에서 발표한 표준 문서로 단말정보 저장소 내부 데이터에 액세스하여 서비스를 제공받기 위한 기초적인 인터페이스를 정의하고 있다. 그림 1은 W3C에서 제시한 표준 인터페이스의 구성도이다[5].

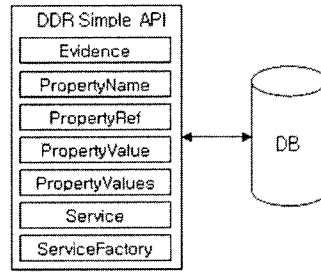


그림 1. DDR Simple API 구조

제시된 인터페이스를 사용하여 DDR 시스템을 구현 가능하며, 이를 통해 DDR 상호간의 호환성을 확보할 수 있게 된다.

## 2.4 DDR Core Vocabulary

모바일 웹에서 콘텐츠의 적응화 과정을 위해 필수적인 속성(Property)에 대해 정의 하고 있다. 이는 DDR 시스템에서 프로파일을 구성하는 기본 단위로 사용된다[6]. 단말정보 표현언어는 기본적으로 본 표준에 정의된 속성들을 지원할 것을 권하고 있다. 또한 단말정보 저장소에서도 해당 속성을 지원해야 한다.

## III. 단말정보 저장소 시스템 설계

본 논문에서 제시하는 단말정보 저장소 시스템은 크게 3부분으로 나뉜다. 단말정보 데이터의 확보를 위한 단말정보 변환부와 단말정보 DB에서 특정 프로파일을 생성하는 단말정보 저장소 인터페이스, 마지막으로 생성된 단말정보를 전송하기 위한 단말정보 전송부이다. 그림 2에서 단말정보 저장소 시스템의 전체 구성을 표현하였다.

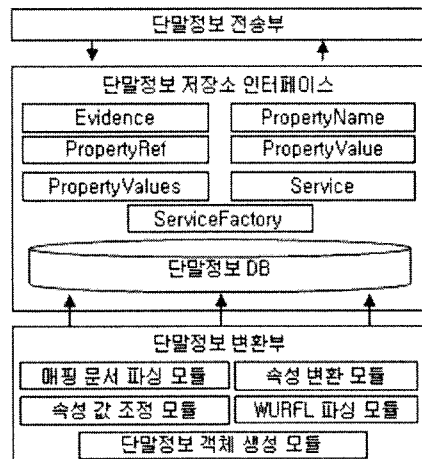


그림 2. 단말정보 저장소 시스템 구조

### 3.1 단말정보 변환부

단말정보 저장소의 데이터 확보와 단말정보 표현언어간의 호환성 확보를 위해 WURFL 데이터를 UAProf로 변환하는 기능을 제공한다. 다음 그림 3은 단말정보 변환부의 모듈 구성도이다.

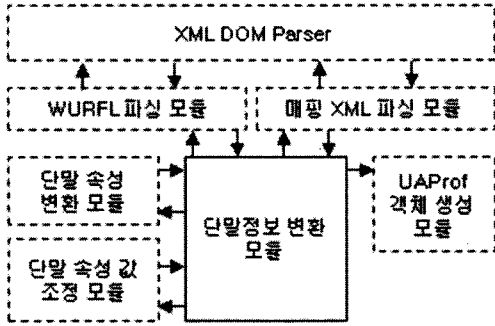


그림 3. 단말정보 변환 모듈 구성

단말정보 변환 모듈은 변환에 필요한 데이터를 파싱하는 과정에서 파싱된 데이터를 비교하여 변환하는 과정을 진행한다. 먼저 XML DOM Parser로부터 WURFL과 매핑 XML 파일을 파싱한다.

이때 매핑 XML 파일은 그림 4와 같이 WURFL의 각 속성 정보가 UAProf에서 표현되는 형식을 기술할 수 있도록 구성되어 있다.

```

<?xml version="1.0" encoding="UTF-8"?>
<IELEMENT ConvertData (Uaprof*)>
<IATTLIST ConvertData
  source CDATA #REQUIRED
  target CDATA #REQUIRED>
<IELEMENT Uaprof (Match?)*>
<!ATTLIST Uaprof
  name CDATA #REQUIRED
  component CDATA #REQUIRED
  datatype CDATA #REQUIRED>
<IELEMENT Match (wurfl)*>
<!ATTLIST Match
  decorator CDATA #REQUIRED>
<IELEMENT wurfl EMPTY>
<!ATTLIST wurfl
  name CDATA #REQUIRED
  datatype CDATA #REQUIRED
  value CDATA #IMPLIED>
    
```

그림 4. 매핑 XML 스키마 구조

매핑 XML 파일에 기술된 단말정보 프로파일 간 속성들의 매핑 정보를 이용해 단말정보 속성 변환 모듈에서 각각 해당하는 속성으로 이름을 변경하는 과정을 거친다. 단말 속성 값 조정 모듈은 해당 단말에 값으로 사용될 정보를 모아 UAProf에 정의된 방식으로 조정한다. 이렇게 완

성된 단말정보는 UAProf 객체로 생성되어 저장되며 생성된 객체는 데이터베이스의 입력자료로 사용된다.

저장될 데이터베이스의 구조는 그림 5와 같이 uaprof\_devices와 uaprof\_capabilities의 두 개 테이블로 이루어져 있다.

```

CREATE TABLE 'uaprof_capabilities' (
'id' int(11) NOT NULL auto_increment,
'uaprof_device_id' int(11) NOT NULL,
'property' varchar(255) NOT NULL,
'value' varchar(255) NOT NULL,
'container_type' varchar(255) default NULL,
PRIMARY KEY ('id'),
KEY 'uaprof_device_id' ('uaprof_device_id')
)

CREATE TABLE 'uaprof_devices' (
'id' int(11) NOT NULL auto_increment,
'device_id' varchar(20) NOT NULL,
PRIMARY KEY ('id')
)
    
```

그림 5. 단말정보 저장소 DB 스키마

### 3.2 단말정보 저장소 인터페이스부

W3C에서는 단말정보 저장소의 인터페이스를 DDR Simple API에 기술하고 있다. 본 논문에서는 표준에 기술된 인터페이스를 준수하여 설계 및 구현하였다.

Evidence Interface는 단말정보 검색의 단서가 되는 Delivery context를 작성하고 관리한다. 여기서 입력 정보는 단말의 모델명 등 단말을 식별할 수 있는 유일한 값이 된다. PropertyName Interface는 저장소에 저장된 단말 속성 정보를 참조할 수 있다. 즉, 본 시스템의 UAProf 단말정보에서 사용되는 property의 목록을 반환하는 인터페이스이다. PropertyRef Interface는 PropertyName Interface와 다르게 property의 목록을 특정 그룹 단위로 참조하는 것으로 여기서는 UAProf에 정의된 각 콤포넌트에 속한 property만을 참조할 수 있다. PropertyValue Interface는 실제로 property에 해당하는 값 정보를 생성할 때 사용되며 반환할 값의 각 타입정보를 지정할 수 있다. PropertyValues Interface는 다수 property의 값 정보의 생성을 지원한다. 마지막으로 Service Interface는 일련의 단말정보 저장소 서비스를 제어하는 인터페이스이다. 이를 통해 각 evidence정보를 설정하고 해당 property 정보를 검색하여 반환하는 전체 과정에 관여한다.

### 3.4 단말정보 전송부

단말정보 전송부는 콘텐츠 제공자의 요청을 받고 요청에 따라 단말정보 저장소에서 생성된 프로파일을 다시 전송하는 역할을 한다. 단말정보 저장소에 정보를 요청하는 시스템의 플랫폼 및

구현언어와 네트워크 제한 등의 환경요인에 독립성을 확보하기 위해 SOAP을 이용한 통신 방법을 이용해 구현하였다.

#### IV. 단말정보 저장소 시스템 구현

본 단말정보 저장소 시스템은 Linux Ubuntu 8.0.4 운영체제를 기반으로 개발하였고, JDK 1.5와 Eclipse 3.4, Mysql 5.0의 개발 환경에서 구현하였다.

단말정보 처리 전체과정을 테스트하기 위해 임의의 모바일 단말을 설정하여 콘텐츠 제공서버에 접속하도록 하였으며, 콘텐츠 제공 서버가 단말정보 저장소 시스템에 접속하여 단말정보를 제공받는 기본 시나리오로 구현하였다. 그림 6은 본 시스템 구현에서 제시된 시나리오를 나타낸 것이다.

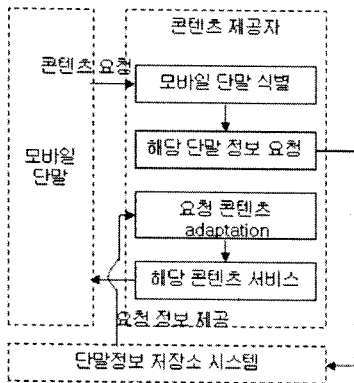


그림 6. 구현 시나리오

그림 7은 단말정보 저장소에서 콘텐츠 제공자의 요청에 따라 검색하여 생성된 단말정보 프로파일의 일부를 나타낸 것이다.

```

- <prf: CcppAccept-Charset>
- <rdf: Bag>
  <rdf: li>ISO-8859-1</rdf: li>
  <rdf: li>UTF-8</rdf: li>
</rdf: Bag>
</prf: CcppAccept-Charset>
- <prf: CcppAccept-Language>
- <rdf: Seq>
  <rdf: li>de</rdf: li>
  <rdf: li>en</rdf: li>
  <rdf: li>es</rdf: li>
  <rdf: li>fr</rdf: li>
  <rdf: li>it</rdf: li>
  <rdf: li>pt</rdf: li>
</rdf: Seq>
</prf: CcppAccept-Language>
    
```

그림 7. 단말정보 저장소 검색 결과

단말정보 저장소 시스템에서는 저장소 내부의 단말정보 데이터를 검색한 후 UAProf의 형식으로 RDF 파일을 생성하여 단말정보 요청자에게 서비스하게 된다.

#### V. 결 론

모바일 단말의 성능향상으로 각종 콘텐츠에 대한 요구가 증대되었다. 이에 다양한 콘텐츠를 모바일 단말에 최적화된 형식으로 제공하기 위한 적응화 과정이 필요하다. 이를 위해 단말의 성능 및 사용자의 선호 정보를 기술하는 단말정보 표현언어가 개발되었지만 하나의 표준으로 통일되지 않아 단말정보 데이터를 통합하기 어려운 점과 기록된 데이터를 효율적으로 저장하고 관리할 수 있는 단말정보 저장소의 부재로 단말정보 데이터를 서비스 받는데 문제가 있었다.

이에 본 논문에서는 W3C의 표준 인터페이스에 부합하는 단말정보 저장소 시스템을 설계 및 구현하였다. 단말정보 저장소 사용자는 표준 인터페이스를 통해 특정 단말에 속한 속성 정보를 개별 혹은 그룹 단위로 요청하여 참조할 수 있다.

또한 단말정보 표현언어 상호 호환성의 부재 및 저장소 내부에서 사용될 단말정보 데이터의 확보를 위한 방안으로 프로파일의 변환 방법을 제시하였다. 이는 단말정보 확보를 용이하게 하여 보다 정확한 데이터를 요청자에게 제공하여 소비자가 질 높은 콘텐츠를 서비스 받을 수 있을 것으로 기대된다.

향후 연구로는 현재 본 시스템에서 지원하는 WURFL-UAProf 간의 변환에 추가로 변환 가능한 단말정보 프로파일의 지원 범위를 넓히고, 단말정보 데이터를 검색하는 과정의 효율성을 높이기 위한 연구가 필요하다.

#### 참고문헌

- [1] OMA, User Agent Profile 2.0, 2007
- [2] WURFL, Wireless Universal Resource File <http://wurfl.sourceforge.net>
- [3] 김태현, "모바일 단말정보 처리시스템에 관한 연구", 2007
- [4] W3C, DDR Requirements 1.0, Working Group Note 2007
- [5] W3C, DDR Simple API, Working Draft, 2008
- [6] W3C, DDR Core Vocabulary, Working Draft 2007