

유비쿼터스 프로파일의 효율적인 관리를 위한 프레임워크의 설계

김진성, 김경식, 이재동
단국대학교 컴퓨터과학과
e-mail : ckjs001@empal.com

A Design of Framework for Effective Management of Ubiquitous Profile

Jin-Sung Kim, Kyung-Sik Kim and Jae-Dong Lee
Dept. of Computer Science, Dankook University

요약

유비쿼터스 프로파일은 유비쿼터스 환경에서 콘텐츠 적응화를 위해 CC/PP를 확장한 프로파일로서, 프로파일을 구성하는 정보들은 네트워크에 분산되어 존재하며, 생성 시 게이트웨이의 프로파일 관리 저장소에서 통합되게 된다. 이와 같이 프로파일들이 네트워크에 분산되어 존재하기 때문에 통합 및 관리에 어려움을 가지고 있다. 본 논문에서는 효율적인 유비쿼터스 프로파일들을 통합 운영할 수 있는 프레임워크를 설계한다. 제안한 프레임워크는 프로파일 운영 아키텍처와 프로파일 처리 기능을 중심으로 설계하였으며, 통합 및 관리를 할 수 있는 기능을 포함하고 있다. 프레임워크는 운영되는 장소에 따라 클라이언트 프레임워크(CF), 유비쿼터스 프레임워크(UF), 저장소 프레임워크(RF)로 구분되며, 특성에 맞는 기능을 가지고 있다. 제안된 프레임워크는 프로파일 생성, 전송, 통합 관리 기능을 가지고 있어, 유비쿼터스 프로파일 운용 시스템 구축에 효율적이다.

1. 서론

정보통신 기술의 발전과 인터넷의 급속한 확산으로 인하여, 사용자들은 다양한 디바이스를 사용하여 인터넷에 존재하는 다양한 콘텐츠들을 이용하고 있다. 그러나 다양한 디바이스들은 자기 디바이스에 맞게 또는 비슷하게 만들어진 콘텐츠만을 이용할 수 있어, 다양한 종류의 콘텐츠 이용에는 한계는 가지고 있다. 이러한 문제점을 해결하기 위해서 프로파일을 이용한 콘텐츠 변환이 대두었다. 프로파일을 이용한 콘텐츠 변환은 프로파일에 사용자 정보, 디바이스 정보, 콘텐츠 정보 등 다양한 정보들을 이용하여 프로파일을 구성한 후 콘텐츠 변환 시 적용함으로써 사용자에게 최적화된 콘텐츠를 제공할 수 있도록 하는 방법이다.

유비쿼터스 프로파일은 콘텐츠 적응화를 위해 W3C(The World Wide Web Consortium)[9]의 표준인 CC/PP(C

omposite Capability/Preference Profiles)[7]를 확장하여 구성된 프로파일이다. 유비쿼터스 프로파일은 생성 시 네트워크에 존재하는 프로파일 저장소들에서 필요한 정보를 수집하여 프로파일을 생성하게 된다. 그러나 프로파일 구성에 필요한 정보들이 네트워크에 분산되어 존재하기 때문에 프로파일의 통합 및 관리에 어려움을 가지고 있다.

본 논문에서는 분산되어 존재하는 정보들을 효율적으로 통합 관리할 수 있는 프레임워크를 설계한다. 프레임워크 설계를 위해 유비쿼터스 프로파일 운영 아키텍처를 기반으로 하며, 프로파일의 처리 기능에 따라 기능을 분류하였다. 분류된 특성 및 운영되는 장소에 따라 프레임워크를 다음과 같이 설계하였다. 클라이언트의 사용자 디바이스에 위치하며, 사용자와 관련된 프로파일 생성 및 전송을 담당하는 클라이언트 프레임워크 CF(Client Framework), 사용자 정보 저장소, 디바이스 정보 저장소, 콘텐츠 정보 저장소, 상황정보 저장소에 위치하며 저장소에 따른 정보들을 관리하고 유비쿼터스 프로파일 구성에 필요한 정보를 제공하는 저장소 프레임워크 RF(Repository Framework), 게이트웨이 프로파일 저장소에 위치하며, 클라

This research was conducted by the research fund of the CT(Culture technology) support program supervised by the KOCCA(Korea Culture and Content Agency)

이언트 프레임워크와 저장소 프레임워크에 프로파일 교환을 통해 유비쿼터스 프로파일을 생성 및 관리를 담당하는 유비쿼터스 프레임워크 UF(Ubiquitous Framwork)등이 있다. 제안한 프레임워크는 유비쿼터스 프로파일 운영을 위한 다양한 기능들을 가지고 있어, 프로파일 운영 시스템 구축 시 효율적이다.

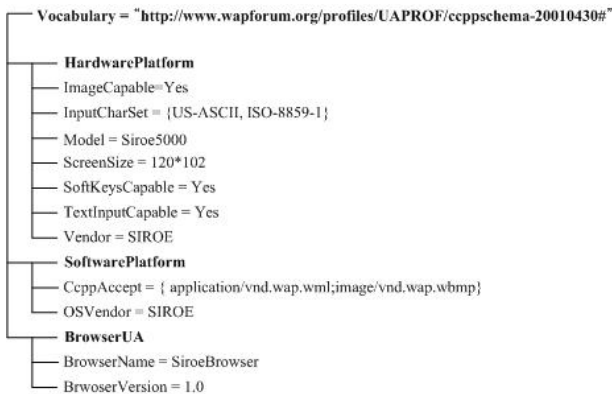
본 논문은 다음과 같이 구성된다. 2장에서는 콘텐츠 적응화를 위한 프로파일 연구들에 대해서 알아본다. 3장에서는 유비쿼터스 프로파일의 구성 및 설계 방법에 대해서 알아보도록 한다. 4장에서는 유비쿼터스 프로파일을 운영을 위한 아키텍처에 대해서 살펴본다. 5장에서는 4장의 운영 아키텍처와 기능을 중심으로 프레임워크를 설계한다. 6장에서는 결론 및 향후 연구 방향에 대해서 알아본다.

2. 관련연구

2.1 콘텐츠 적응화를 위한 프로파일

콘텐츠 적응화를 위해 제안된 프로파일로는 W3C의 Independent Group의 CC/PP가 있다. CC/PP는 사용자 정보 및 선호도 정보를 표현하기 위해 제안되었으며 RDF(Resource Description Framework)[5]를 이용하여 표현한다. CC/PP는 Component, Attributes, Values의 구조로 되어 있으며, 각 컴포넌트에 대한 Default 값들을 가질 수 있다. CC/PP는 이러한 구조로 되어 있어 상호교환, 분해, 균일, 확장에 좋다. 그러나 CC/PP는 단순한 구조를 가지고 있어 복잡한 프로파일 구성에는 부적합하다[1].

(그림 1)은 CC/PP의 간단한 예제로서 HardwarePlatform, SoftwarePlatform, Browser로 구성된 프로파일을 나타낸다.



(그림 1) CC/PP 예제

UAProf(User Agent Profile)[8]은 Open Mobile Alliance에서 WAP(wireless application protocol)용 모바일 디바이스를 위해 제안된 프로파일이다. UAProf는 CC/PP를 기반으로 하고 있으며 RDF를 이용하여 표현하였다. UAProf는 WAP(Wireless Application Protocol)용 모바일 단말기에 최적화하기 위해 Hardware Platform, Software Platform, Browser UA, Network Characteristic, WAP Characteristic, Push Characteristic 등으로 컴포넌트를 구성하였다. 또한 각 컴포넌트의 속성에 대한 제약사

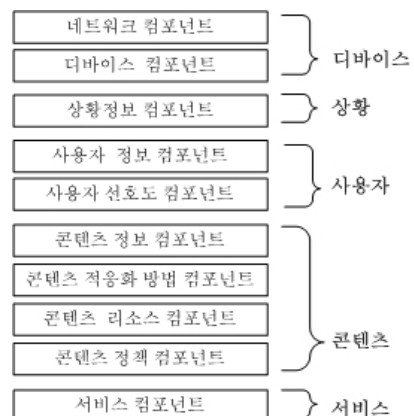
항을 기술하기 위해 Schema를 이용하고 있다.

그러나 UAProf는 모바일 단말기 기능에 초점을 맞춰 프로파일을 설계하여 다른 디바이스에서 사용하기에는 부적합하며, 이 프로파일을 사용하기 위해서는 단말기 및 Proxy 서버에 프로파일 처리 모듈이 필요하다.

UPS(Universal Profiling Schema)[2][3][4]는 INRIA의 Opera 프로젝트 일부로서 콘텐츠 교섭 및 멀티미디어 콘텐츠 적응화를 위해 제안되었다. UPS는 CC/PP를 기반으로 하고 있으며 RDF를 사용하여 표현하였다. UPS는 Client Profile Schema, Document Instance Profile Schema, Resource Profile Schema, Adaption Method Profile, Client Resource Schema, Network Profile Schema로 컴포넌트를 구성하였다. UPS는 프로파일 구성 시 사용자에게 대한 감성/상황 정보, 사용자 선호도 등을 프로파일 구성에 포함하지 않고 있어, 사용자에게 최적화된 콘텐츠를 제공하기 위한 프로파일로는 부적합하다.

3. 유비쿼터스 프로파일

유비쿼터스 프로파일은 유비쿼터스 환경에서 사용자가 소유하고 있는 디바이스에 적응화된 콘텐츠를 제공하기 위해 제안 되었다. 유비쿼터스 환경에서 콘텐츠 적응화를 위해서는 사용자의 다양한 정보들이 유기적으로 결합되어 있어야 한다. 유비쿼터스 프로파일은 유비쿼터스 환경에서 콘텐츠 적응화에 대한 필요한 요소들을 디바이스, 상황, 사용자, 콘텐츠, 서비스 등으로 분류하고 네트워크, 디바이스, 상화정보, 사용자 정보, 사용자 선호도, 콘텐츠 정보, 콘텐츠 적응화 방법, 콘텐츠 리소스, 콘텐츠 정책, 서비스, 동적구성 정보 컴포넌트로 구성하였다. (그림 2) 유비쿼터스 프로파일의 구성 정보를 보여주고 있다.



(그림 2) 유비쿼터스 프로파일

3.1 유비쿼터스 프로파일 구성

네트워크 컴포넌트는 사용자 디바이스와 게이트웨이간 네트워크 정보를 표현 한다. 네트워크 컴포넌트에는 프로토콜, 전송속도, Qos(Quality of Services)로 구성된다. 디바이스 컴포넌트들은 하드웨어에 관련된 사항들을 표현한다. 상황정보 컴포넌트는 사용자가 존재하는 환경에 대한 정보를 표현한다. 사용자 컴포넌트는 사용자를 식별할 수

있도록 하는 정보를 표현한다. 사용자를 식별할 수 있는 대표적인 방법인 아이디와 패스워드를 이용한다. 사용자 선호도 컴포넌트는 사용자가 이용하는 콘텐츠와 서비스들을 데이터베이스에 저장 한 후 기본 서비스 시 사용자에게 자동으로 맞춤화하여 서비스할 수 있는 콘텐츠에 대한 종류, 콘텐츠 적응화 방법, 서비스 퀄리티 등으로 구성된다. 콘텐츠 정보 컴포넌트는 사용자가 콘텐츠를 요청 시 서비스에 대한 정보를 표현한다. 콘텐츠 정보 컴포넌트에는 콘텐츠의 주소, 콘텐츠 제공 업체 정보 등으로 구성된다. 콘텐츠 적응화 방법 컴포넌트는 콘텐츠 적응화를 위해 필요한 사항을 표현한다. 콘텐츠 리소스 컴포넌트는 콘텐츠를 구성하는 콘텐츠의 종류, 크기, 제작 년도 등 리소스에 대한 사항 등으로 구성된다. 콘텐츠 정책 컴포넌트는 사용자에게 콘텐츠에 대한 접근 권한, 접근 허용, 접근 금지 등 콘텐츠에 대한 사용권한을 표현한다. 서비스 컴포넌트는 서비스되고 있는 정보들을 표현한다. 콘텐츠 이용 시간, 콘텐츠 이용 우선순위, 콘텐츠 퀄리티, 과금 정보 등으로 구성된다. 구성 정보 컴포넌트는 프로파일을 구성하는 컴포넌트들에 메타 정보를 표현한다.

3.2 유비쿼터스 프로파일

유비쿼터스 프로파일은 RDF, RDF Schma[6], CC/PP Structure, 표준 Vocabulary 등을 적용하여 설계되었다. <표 1> 상황 정보를 기술하고 있는 유비쿼터스 프로파일을 나타내고 있다.

<표 1> 상황정보를 기술하고 있는 유비쿼터스 프로파일

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <rdf:rdf xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#" xmlns:upc="http://www.itpioneer.org/schema/ubiquitouschema/context#" xmlns:upd="http://www.itpioneer.org/schema/ubiquitouschema/context#" >
3. <rdf:description rdf:id="Context">
4. <ccpp:component>
5. <rdf:description rdf:id="environment">
6. <rdf:type rdf:resource="http://www.itpioneer.org/schema/ubiquitouschema/context#environment"/>
7. <upc:stability>stable</upc:stability>
8. <upc:sound>silent</upc:sound>
9. <upc:light>norma</upc:light>
10. </rdf:description>
11. <rdf:description rdf:id="sensibility">
12. <rdf:type rdf:resource="http://www.itpioneer.org/schema/ubiquitouschema/context#sensibility"/>
13. <upc:visual>stable</upc:visual>
14. <upc:hearing>silent</upc:hearing>
15. <upc:bloodpress>norma</upc:bloodpress>
16. </rdf:description>
17. </ccpp:component>
18. </rdf:description>
19. </rdf:rdf>
    
```

라인 3~19까지는 상황정보를 구성하는 항목들에 대한 내

용을 나타내고 있다.

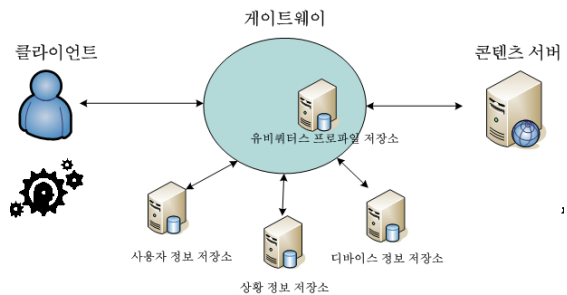
4. 유비쿼터스 프로파일 운영을 위한 아키텍처

본 장에서는 유비쿼터스 프로파일 운영을 위한 아키텍처에 대해서 알아본다.

4.1 유비쿼터스 프로파일 운영을 위한 아키텍처

유비쿼터스 아키텍처는 클라이언트, 게이트웨이, 콘텐츠 서버로 구성된다. 게이트웨이는 클라이언트와 콘텐츠서버 사이의 중간 매개체 역할을 담당하면서 다음과 같은 기능을 가지고 있다 첫째, 유비쿼터스 프로파일을 관리한다. 유비쿼터스 프로파일은 게이트웨이 안에 있는 유비쿼터스 프로파일 저장소에서 프로파일 생성, 삭제, 추가, 업데이트 기능을 수행한다. 둘째, 프로파일 구성 시, 필요한 정보를 가지고 있는 서버들과 교환을 통해 필요한 정보를 수집한다. 운영에 필요한 서버들은 상황정보를 저장하는 있는 상황정보 저장소, 사용자에게 대한 정보, 선호도 정보를 가지고 있는 사용자 정보 저장소, 디바이스 스펙에 대한 정보를 가지고 있는 디바이스 정보 저장소로 구성된다.

콘텐츠 서버는 사용자에게 제공되는 콘텐츠와 콘텐츠에 대한 메타데이터를 가지고 있다. 메타데이터에는 콘텐츠 적응화 방법, 리소스 정보, 콘텐츠 정책 정보 등을 가지고 있다.



(그림 3) 유비쿼터스 프로파일 운영을 위한 아키텍처

5. 프레임워크 설계

본 장에서는 4장에서 제시한 프로파일 아키텍처를 기반으로 프레임워크를 설계한다. 콘텐츠 적응화를 위한 프레임워크의 일부분으로 유비쿼터스 프로파일을 교환만을 고려하여 기능을 분석하고 설계한다. 프로파일을 구성하는 정보들은 네트워크에 분산되어 존재하기 때문에 존재하는 위치에 따라 제공하는 기능이 다르다. 따라서 본 논문에서는 기능에 따라 클라이언트 프레임워크, 유비쿼터스 프레임워크, 저장소 프레임워크를 분류하였다. 본 논문에서는 프레임워크를 구성하는 기능들을 간단히 알아본다.

5.1 클라이언트 프레임워크 (Client Framework)

클라이언트 프레임워크는 클라이언트의 사용자 디바이스에 위치하며 사용자 정보, 콘텐츠 정보, 디바이스 정보, 서비스 정보, 네트워크 정보, 등과 사용자 프로파일을 생성하여 유비쿼터스 프레임워크에 전송하는 역할을 담당하게 된다. 클라이언트 프레임워크에서 생성되는 프로파일

은 네트워크 전송 및 처리의 효율성을 위해 메타데이터로 구성되어 있다. 프로파일을 생성하는 기능과 전송하는 기능이 필요하다. 사용자 정보는 개인의 프라이버시에 관련된 내용을 포함하고 있기 때문에 암호화와 관련된 기능이 필요하다.

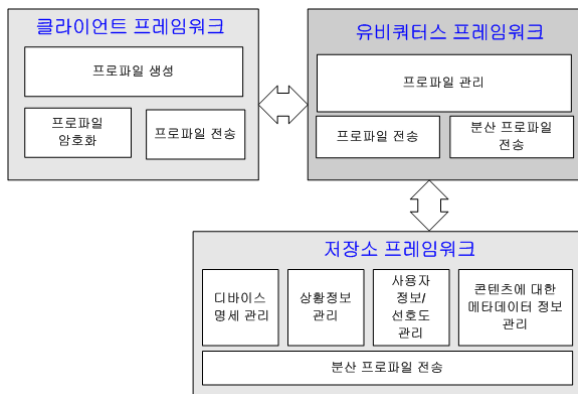
5.2 유비쿼터스 프레임워크 (Ubiquitous Framework)

유비쿼터스 프레임워크는 게이트웨이의 유비쿼터스 프로파일 저장소에 위치하며 클라이언트 프레임워크에서 전송 받은 프로파일들을 분석하여 유비쿼터스 프로파일을 구성한다. 클라이언트 프레임워크에서 전송받은 프레임워크는 크기를 줄이기 위해서 메타데이터로 구성되었기 때문에 유비쿼터스 프레임워크에서는 메타데이터를 이용하여 사용자 정보 저장소, 상황정보 저장소, 디바이스 정보 저장소, 콘텐츠 서버 등에 필요한 프로파일을 수집하여 프로파일을 구성한다. 그러므로 유비쿼터스 프레임워크는 프로파일 분석 기능, 관리 기능, 클라이언트 통신 기능, 저장소와 통신 기능등을 필요로 한다.

5.3 저장소 프레임워크 (Repository Framework)

저장소 프레임워크들은 저장소에 존재하는 프로파일에 정보들을 제공하는 기능을 담당하게 된다. 유비쿼터스 프로파일 구성을 위해 필요한 저장소는 디바이스 정보 저장소, 상황 정보 저장소, 사용자 정보 저장소, 콘텐츠 정보 저장소등이다. 이러한 저장소들은 유비쿼터스 프레임워크의 요청을 따라 프로파일을 제공하기 위해 분산 프로파일 전송 기능과 각 프로파일을 관리하는 기능을 가지고 있어야 한다.

(그림 4)는 제안한 유비쿼터스 프로파일 운용을 위한 프레임워크를 나타내고 있다.



(그림 4) 유비쿼터스 프로파일 처리를 위한 프레임워크

6. 결론 및 향후 연구 방향

본 논문에서는 콘텐츠 적응화를 위해 제안된 유비쿼터스 프로파일의 효율적인 관리를 위한 프레임워크를 제안하였다. 유비쿼터스 프로파일은 네트워크에 분산되어 존재하기 때문에 통합 및 운영에 어려움을 가지고 있다. 이러한 문제점 해결하기 위해 프로파일 운영 아키텍처와 프로파

일 처리 기능을 중심으로, 위치하는 장소에 따라 클라이언트 프레임워크, 유비쿼터스 프레임워크, 저장소 프레임워크를 설계하였다. 클라이언트 프레임워크는 사용자의 디바이스에 위치하면서 사용자 정보, 사용자 선호 정보, 콘텐츠 정보들을 수집하여 유비쿼터스 프레임워크에 전송하는 역할을 담당한다. 유비쿼터스 프레임워크는 클라이언트 프레임워크, 저장소 프레임워크들과 프로파일 교환을 통해 유비쿼터스 프로파일을 생성하게 된다. 저장소 프레임워크는 디바이스 정보 저장소, 상황정보 저장소, 사용자 정보 저장소, 콘텐츠 저장 정보 저장소에 위치하며 유비쿼터스 프레임워크에게 프로파일을 제공하고, 프로파일들을 관리하는 역할을 담당한다. 제안한 프레임워크는 프로파일 전송을 위한 기능과 통합 관리 기능을 가지고 있고, 존재하는 위치에 따라 특징적인 기능을 가지고 있도록 설계함으로써 유비쿼터스 프로파일 운영 시스템 구축에 효율적이다

향후에는 본 논문에서 제안한 프레임워크들을 구현하고 유비쿼터스 프로파일 운영에 대한 실험을 진행한다. 실험을 통해 프레임워크의 미비점을 분석하고, 보완하여 콘텐츠 적응화를 위한 프레임워크로 발전시킨다.

참고 문헌

[1] P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen, J. E. Malm, "Managing context information in mobile devices", Pervasive Computing. IEEE, vol 2, issue 3, pp. 42-51, July-sept. 2003

[2] T. Lemlouma, N. Layaïda, "Encoding Multimedia Presentation for User Preferences and Limited Environments", IEEE International Conference on Multimedia & Expo (ICME), pp. 165-168, July. 2003.

[3] T. Lemlouma, N. Layaïda, "Content Adaptation and Generation Principles for Heterogeneous Clients", W3C Workshop on Device Independent Authoring Techniques, Sept. 2002.

[4] T. Lemlouma, N. Layaïda, "The Negotiation of Multimedia Content Services in Heterogeneous Environments", In the MMM 2001: the 8th International Conference on Multimedia Modeling, pp. 187-206, Nov. 2001.

[5] B. Bridge, Hewlett-Packard Laboratories, "RDF Primer", W3C, Feb 2004

[6] B. Bridge, "RDF Vocabulary Description Language 1.0 :RDF Schema", Feb 2004

[7] G. Klyne, et al, "Composite Capability/Preference Profiles(CC/PP) : Structure and Vocabularies 1.0", W3C, Jan 2004

[8] Open Mobile Alliance, "UAProf(User Agent Profile)", Open Mobile Alliance, May 2003

[9] The World Wide Web Consortium(W3C), "http://www.w3c.org"