
장치 독립적 디지털 콘텐츠 제공을 위한 CC/PP 프로파일 생성 시스템 설계 및 구현

변영철* · 강철웅** · 이상준*

Design and Implementation of CC/PP Profiling System
for Providing of Device Independent Digital Contents

Yung-Cheol Byun · Chul-Ung Kang · Sang-Joon Lee

이 논문은 2004년도 제주대학교 발전기금 국외파견 연구지원계획에 의하여 연구되었음

요 약

유비쿼터스 컴퓨팅 환경에서 서버가 다양한 유형의 단말기에 대해 각각의 장치에 적합한 콘텐츠를 효과적으로 제공하기 위해서는 단말기 정보 및 관련 상황 정보를 알고 있어야 한다. CC/PP 표준 스펙은 하드웨어, 소프트웨어, 네트워크, 사용자 취향 정보 등을 기술하고 전송하기 위한 규약으로서, RDF을 이용하여 단말기 프로파일 정보를 기술한다. 한편, WAP 포럼에서는 클라이언트가 이러한 CC/PP 프로파일 정보를 스스로 서버에게 전송할 수 있도록 하기 위한 프로토콜로서 W-HTTP 프로토콜을 정의하였다. 하지만 기존의 HTTP 프로토콜을 이용하는 클라이언트의 경우 CC/PP 프로파일 정보를 서버에게 제공하지 못하는 문제가 발생한다. 본 논문에서는 W-HTTP 프로토콜 클라이언트뿐만 아니라 HTTP 프로토콜을 이용하는 클라이언트에 대해서도 CC/PP 프로파일을 생성하여 서버에게 제공하기 위한 미들웨어 시스템을 설계하고 구현한다.

ABSTRACT

The server system of digital contents has to know the context information about client devices to provide the appropriate contents for each device effectively. CC/PP standard specification is an agreement for describing and transmission of the information related with a client device. In this case, the information on device hardware and software, networks and user's preference is included here. In the mean time, WAP forum defined W-HTTP protocol to transmit the CC/PP profile information of a client device to a server system. However, the client devices which use existing HTTP protocol to transmit their information cannot provide the CC/PP profile information to a server. In this paper, we propose an effective method to create and provide the CC/PP profile in the clients which use not only HTTP protocol but also W-HTTP protocol to transmit device information.

키워드

CC/PP Profiling, Device Independence, Contents Transformation, Mobile Devices

* 제주대학교 통신컴퓨터공학부 컴퓨터공학전공

접수일자 : 2006. 8. 4

** 제주대학교 기계에너지시스템공학부

I. 서 론

인터넷 기술과 모바일 컴퓨팅 기술의 발달로 인하여 일상생활에서의 컴퓨팅 환경은 언제 어디서든지 컴퓨터 기술을 활용할 수 있는 유비쿼터스 환경으로 변화가고 있다. 이러한 변화의 한 가운데에는 PDA, 스마트폰, 텔레매틱스 기기와 같은 유연하고 이동성이 있는 최첨단 기기들이 자리 잡고 있으며, 이에 따라 현재의 단말기뿐만 아니라 앞으로 출현할 새로운 기능과 제약을 갖는 단말기들에 대해 최적화된 서비스를 효과적으로 제공할 수 있는 방법 및 인증 방법이 요구되고 있다[1, 2].

한편, 다양한 형태의 이기종 모바일 단말기들이 증가함에 따라 각 단말기들이 갖는 서로 다른 입출력 기능과 하드웨어, 소프트웨어, 네트워크 상의 특성으로 인하여 단말기에 최적화된 서비스 개발의 복잡성은 증가한다. 이상적으로 볼 때 서비스 제공자들은 한 가지 버전의 서비스를 개발하여 모든 단말기에 동일하게 제공할 수 있기를 바라지만, 현실적으로 볼 때 단말기들의 다양한 특성으로 인하여 각각의 단말기에 적합한 서비스를 중복으로 개발해야만 한다.

이러한 문제점을 해결하기 위하여 다양한 단말기들을 효과적으로 지원할 수 있는 장치 독립적(device-independent) 접근 방식에 대한 연구가 진행 중이다[3, 4]. 장치 독립적 접근 방식을 사용하면 서비스 제공자는 하나의 서비스만을 개발하고, 이를 단말기의 특성에 최적화된 형태로 변환하여 제공함으로써 서비스 중복 개발에 따른 비용을 절감할 수 있다.

단말기의 특성을 기반으로 장치 독립적 서비스 제공을 위해서는 먼저 상황기반 정보가 필수적으로 요구된다. 상황기반 정보에는 단말기의 특성뿐만 아니라 네트워크의 특성, 사용자의 기호(preference), 사용자의 위치나 사용하는 언어 등 애플리케이션에 특화된 정보도 포함된다.

서버가 상황기반 정보를 바탕으로 사용자의 단말기에 최적화된 서비스를 제공할 수 있도록 하기 위해서는 단말기 정보를 어떻게 서버로 제공할 것인가 하는 콘텐츠 협의(content negotiation) 문제가 발생한다. 현재 표준 HTTP 헤더에는 콘텐츠 협의를 위한 몇 가지 정보가 포함되어 있다[5]. 그러나 HTTP는 브라우저 특성에 대한 기술만을 대상으로 설계되어 있기 때문에 단말기 정보를 충분히 표현할 수 없는 한계를 가지고 있다.

CC/PP(Composite Capabilities / Preference Profile) 표준

스펙은 하드웨어, 소프트웨어, 네트워크, 지원하는 서비스 정보 등 일련의 단말기 특성 정보 및 제한 조건과 사용자 취향에 관한 정보를 기술하고 전송하기 위한 규격으로 RDF (Resource Description Framework)로 정보를 기술한다[6, 7, 8, 9]. CC/PP 기술은 현재의 사용되는 다양한 단말기뿐만 아니라 앞으로 개발될 새로운 기능의 단말기들이 자신의 프로파일 정보를 기술하고 전송하도록 함으로써 장치 독립적 서비스 제공을 위한 콘텐츠 협의의 문제는 쉽게 해결한다[10].

이에 본 논문에서는 유비쿼터스 환경 하에서 CC/PP 프로파일을 지원하는 단말기뿐만 아니라 지원하지 않는 단말기에 대해서도 단말기 특성 정보를 추출, 분석할 수 있도록 한다.

본 논문의 구성은 다음과 같다. II 장에서는 CC/PP를 중심으로 한 관련 연구와 문제점을 살펴보고, III 장에서는 본 논문에서 제시한 CC/PP 프로파일 생성 방법에 대해 살펴본다. 그리고 IV 장에서는 본 논문에서 제안한 방법을 구현하여 테스트한 결과를 살펴보고, 마지막 V 장에서 본 연구의 결론에 대해 설명한다.

II. 관련 연구 및 기술 현황

2.1. HTTP 요청 헤더와 프로파일 정보

기존의 방법에서는 클라이언트 단말기의 특성을 파악하기 위해서 HTTP 요청 헤더에 있는 필드 중 User-Agent 필드와 Accept 필드를 분석한다. User-Agent 항목은 다음과 같은 형식으로 구성된다.

```
Browser / version (platform ;
security-level ; OS-or-CPU description)
첫 부분에 브라우저 이름과 버전 정보가 있으며, 괄호 안에 단말기의 플랫폼과 보안 레벨, 운영체제 또는 CPU에 대한 정보를 세미콜론으로 구분하여 표시한다. 실제 예는 다음과 같다.
```

```
Mozilla/4.04 (X11; I; SunOS 5.4 sun4m)
Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1)
Mozilla/1.22 (compatible; MSMB13;
LG-KP6100; CellPhone)
```

User-Agent 필드를 통하여 클라이언트 단말기의 특성을 파악하는 방법은 크게 두 가지로 구분될 수 있다. 첫째 방법은 단말기 특성과 적용 콘텐츠의 명시적인 연결 방식으로서 프로그램이 User-Agent 항목을 검사하여 적용되는 콘텐츠를 선택하는 방식이다. 즉 User-Agent 필드에 Nokia가 있는 경우와 MSIE가 있는 경우 제공하는 콘텐츠를 달리하는 방식이다. 이러한 방식은 단말기 특성에 관한 명시적인 내용 없이 단말기의 식별 번호로 제공할 콘텐츠를 결정한다.

두 번째 방법은 단말기 특성과 해당 단말기에 적용할 콘텐츠를 명시적으로 연결하는 방식으로서 단말기 특성 데이터베이스를 이용하는 방법이다. 콘텐츠 요청을 받으면 서버는 단말기의 특성을 결정하기 위하여 단말기의 식별 번호를 데이터베이스에 넘기고, 데이터베이스는 단말기 식별 번호에 맞는 해당 단말기의 특성, 즉 데이터베이스에 명시적으로 기술되어 있는 단말기의 특성을 반환한다. 물론 이 방식도 단말기의 식별 번호를 통해 특성을 결정하고는 있지만 제공할 콘텐츠 결정에 있어서는 단말기의 식별 번호를 통해 직접 결정하는 것이 아니라 명시되어 있는 단말기의 특성 정보를 통하여 제공할 콘텐츠가 결정되는 방식이다. 사실 이러한 방식은 다음 절에서 설명할 CC/PP와 별 반 다를 것이 없어 보인다. 하지만 한 가지 주요한 차이점이 존재하는데, 그것은 CC/PP 지원 단말기인 경우에는 자신의 특성 정보를 서버에게 스스로 전송하기 때문에 새로운 단말기가 나왔을 때 단말기 특성 데이터베이스를 주기적으로 업데이트 할 필요가 없다는 것이다.

HTTP 요청 헤더는 User-Agent 필드 이외에도 Accept, Accept-Charset, Accept-Encoding, Accept-Language 등이 있다. 다음은 HTTP 요청 시 헤더에 인코딩되는 필드의 예이다.

```
Accept: image/gif, image/x-xbitmap,
image/jpeg, image/pjpeg,
application/x-shockwave-flash,
application/vnd.ms-excel,
application/vnd.ms-powerpoint,
application/msword, /*
Accept-Language: ko
Accept-Encoding: gzip, deflate
```

위의 Accept 요청 헤더 필드를 분석함으로써 해당 브라우저 혹은 단말기는 한국어어를 기본으로 사용하고, 인코딩으로는 gzip 형식의 압축 파일과 평문을 지원하며, gif, x-xbitmap, jpeg, pjpeg 형식의 이미지 파일을 표현할 수 있고, 플래시와 마이크로소프트사의 엑셀, 파워포인트, 워드 형식의 파일을 지원할 수 있음을 알 수 있다.

콘텐츠를 제공하는 웹 서버는 이러한 내용들을 이용하여 단말기의 특성을 어느 정도까지 예측할 수 있다. 그러나 많은 단말기, 혹은 브라우저들이 Accept 문에 어떠한 MIME 타입과도 매치될 수 있는 확장자인 */*를 기술함으로써 모호한 정보를 제공하고 있는 경우도 있다.

2.2. CC/PP

W3C에서 표준으로 내세운 CC/PP 기술은 하드웨어, 소프트웨어, 네트워크, 지원하는 서비스 정보 등 단말기와 관련된 일련의 특성 및 제한 조건과 사용자 기호에 관한 정보 기술 및 전송에 대한 규격으로서, 콘텐츠 제공 서버로 전달되어 클라이언트에게 클라이언트의 요구사항에 맞추어진 콘텐츠를 제공할 수 있도록 한다.

2.1.1 CC/PP와 UAProf

CC/PP 프로파일은 RDF(Resource Description Framework)로 기술되며, 크게 두 단계의 계층 구조를 갖는다. CC/PP 프로파일은 하나 이상의 컴포넌트(component)를 가지며, 각 컴포넌트는 하나 이상의 속성(attribute)을 갖는다. 컴포넌트는 단말기의 하드웨어 플랫폼, 소프트웨어 플랫폼, 브라우저와 관련된 정보를 표현하며, 속성은 이름과 값의 쌍으로 이루는 집합으로서 해당 컴포넌트에 속하는 세부 특성 정보를 나타낸다.

CC/PP 프로파일을 구성하는 컴포넌트와 속성들의 이름 및 제약사항, 그리고 데이터 형은 프로파일 내에 URI(Uniform Resource Identifier)에 명시되어 있는 어휘(vocabulary)에 의해 결정된다. CC/PP 표준에서는 특정 어휘를 지정하지 않고, 어떠한 애플리케이션도 자신만의 어휘를 정의하여 프로파일 기술에 사용할 수 있도록 하고 있는데, 현재 CC/PP를 지원하는 휴대폰 기기들은 자신의 프로파일 기술을 위해 WAP 포럼에서 개발한 UAProf 어휘를 사용하고 있다. UAProf 어휘는 표 1과 같이 크게 6가지의 컴포넌트를 정의하고 있다.

표 1. UAProf 어휘
Table 1. UAProf Vocabulary

Component	기술하는 정보
HardwarePlatform	하드웨어 특성 정보
SoftwarePlatform	장치 운영 환경 정보
BrowserUA	장치의 브라우저 정보
WapCharacteristics	WAP 관련 장치 성능 정보
NetworkCharacteristics	네트워크 특성 정보
PushCharacteristics	장치의 푸시 특성 정보

그림 1은 표 1의 UAProf 어휘를 따르는 CC/PP 프로파일의 예이다. 프로파일을 보면 HardwarePlatform, SoftwarePlatform, BrowserUA 등의 컴포넌트들은 URI로 지정된 어휘에 정의된 컴포넌트들이며, 이러한 각각의 컴포넌트에는 ImageCapable, ScreenSize, BrowserName과 같은 속성들이 그들의 데이터 형에 맞게 기술되어 있다.

컴포넌트를 구성하는 각 속성들은 속성 유형, 집합(collection) 유형, 그리고 분석(resolution) 규칙 등으로 표현된다. 속성 유형은 BrowserUA 컴포넌트의 BrowserName 속성과 같은 일반적인 문자열의 값을 갖는 String 형, HardwarePlatform 컴포넌트의 ImageCapable과 같이 Yes/No 중 하나의 값을 가질 수 있는 Boolean 형, BrowserVersion과 같이 숫자를 값으로 갖는 Number 형, 그리고 ScreenSize와 같이 양수의 쌍으로 나타내어지는 Dimension 등이 있다.

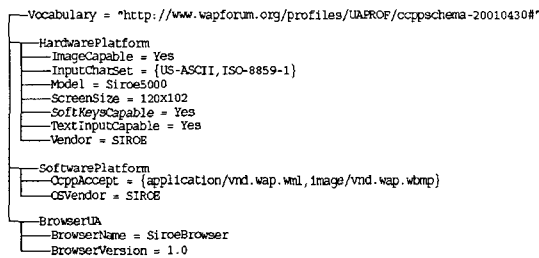


그림 1. UAProf 기반 CC/PP 프로파일 구조
Fig.1. The Structure of CC/PP Profile

집합 유형은 하나의 값을 갖는 Simple 형, 순서가 없는 여러 개의 값을 가질 수 있는 Bag 형, Bag과 같이 여러 개의 값을 가질 수 있으나 해당 값들이 순서적으로 우선 순위를 갖는 Seq 형이 있다.

분석 규칙은 프로파일 내에 같은 이름의 속성이 둘 이

상 선언될 경우의 분석 방법, 또는 단편화된 여러 개의 프로파일을 하나로 합병할 때 같은 이름의 속성들을 어떻게 처리할 지에 대한 규칙을 정의한다. 분석 규칙이 Locked로 정의된 속성은 첫 번째로 정의된 속성 값을 사용하고, 그 후에 나오는 같은 이름의 속성 값들은 무시한다. 분석 규칙이 Override로 지정되면, Locked와 달리 제일 마지막에 나오는 속성 값을 사용하며, Append 규칙으로 지정되면 프로파일 내에 나타나는 모든 속성 값들을 Bag 집합 유형과 같은 형태로 사용한다.

2.2.2 W-HTTP 프로토콜

앞에서 설명한 바와 같이 CC/PP 기술은 일련의 단말기 특성 정보를 클라이언트가 서버에게 스스로 전송한다. 그러나 기존의 HTTP 프로토콜에서는 요청 헤더에 단말기 특성 정보를 추가하기 위한 필드가 정의되어 있지 않다. 따라서 W3C에서는 CC/PP 프로파일의 HTTP 프로토콜을 통한 전송을 위해 HTTP-ex라고 알려져 있는 HTTP Extension Framework에 기반을 둔 CC/PP exchange protocol 표준을 제정하고 있다[8].

이와는 별도로 WAP 포럼에서는 CC/PP 프로파일 전송을 위해 W-HTTP(Wireless Profiled HTTP) 프로토콜을 제안하고 있는데, 이는 HTTP-ex에 기반을 둔 CC/PP 프로토콜과 기능적으로 일치할 뿐만 아니라 HTTP 프로토콜에 적합한 형태로서 현재 CC/PP 프로파일을 지원하는 휴대폰들은 이 방식을 채택하여 서버로 정보를 전송하고 있다. 그림 2는 W-HTTP 프로토콜로 표현한 요청 헤더의 예이다.

```

GET /ccpp/html/ HTTP/1.1
Host: localhost
x-wap-profile: "http://localhost:8080/ccpp/profiles/test09defaults.rdf",
  "1-Rb0sq/mUUF075vkjRyiHw="
x-wap-profile-diff: 1:<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#"
  <rdf:Description rdf:ID="MyDeviceProfile">
    <prf:component>
      <rdf:Description rdf:ID="HardwarePlatform">
        <rdf:type
  rdf:resource="http://www.wapforum.org/profiles/UAPROF/ccppschem-
  20010426#HardwarePlatform"/>
        <prf:BitsPerPixel>16</prf:BitsPerPixel>
      </rdf:Description>
    </prf:component>
  </rdf:Description>
</rdf:RDF>
    
```

그림 2. W-HTTP 요청 헤더의 예
Fig. 2. An Example of W-HTTP Request Header

W-HTTP 프로토콜의 요청 헤더에는 일반적인 HTTP 요청 헤더에 두 가지의 필드가 추가로 들어간다. x-wap-profile 필드와 x-wap-profile-diff 필드가 그것인데, 먼저 x-wap-profile은 해당 클라이언트 단말기 특성 프로파일의 위치를 URI로 기술된다.

그림 2에서는 "http://localhost:8080/ccpp/profiles/test09defaults.rdf" 라는 위치에 있는 프로파일을 자신의 프로파일로 지정하고 있다. URI 다음에 오는 1-Rb0sq/nuUFQU75vAjKyiHw== 라는 문자열은 profile-diff 다이제스트(digest)로서, 처음에 나오는 1-은 profile-diff 순서 번호이고, 나머지 부분은 해당 번호의 profile-diff를 MD5 알고리즘과 Base64 알고리즘을 이용하여 생성한 것이다. 다음으로 x-wap-profile-diff 필드를 보면, 첫 부분에 x-wap-profile에서 지정되었던 순서번호 1이 나오는 것을 볼 수 있고, 뒤이어 참조된 프로파일에 추가되어야 할 단말기 특성이 XML로 기술된다.

요청 헤더에 기술된 프로파일은 미리 정의된 프로파일 분석 규칙에 따라 분석된다. 먼저 x-wap-profile로 지정된 프로파일이 분석된다. 만일 프로파일이 외부의 다른 프로파일을 참조하고 있다면 해당 프로파일 또한 같이 분석된다. 이후 profile-diff의 값은 어휘에 정의된 각 속성의 분석 규칙에 따라 프로파일과 합성되어 해당 단말기에 대한 최종 프로파일로 해석된다. 이때 같은 이름의 속성이 여러 개 존재한다면, 속성이 갖는 분석 규칙에 따라 해당 값을 선택 또는 합병하게 된다. 그림 2의 경우 먼저 test09defaults.rdf 프로파일이 분석된 결과에 profile-diff에서의 BitsPerPixel 항목을 추가하는 형식이 된다. profile-diff에서 사용된 어휘인 ccppschemata-20010426에 의하면 BitsPerPixel 속성의 분석 규칙은 Override로 지정되어 있기 때문에, test09defaults.rdf에 이미 BitsPerPixel 값이 지정되어 있더라도 이 값은 무시되고 profile-diff에서 지정된 BitsPerPixel 값인 16이 사용된다.

이와 같이 CC/PP 기술은 단말기 특성을 효과적으로 표현할 수 있는 기술임을 알 수 있다. 그러나 지원하는 단말기, 즉 자신의 프로파일을 가지고 이를 요청 헤더를 통해 서버에 전달하는 단말기는 현재 그리 많지가 않다. CC/PP를 지원하는 휴대폰들이 출시되고는 있으나 기존에 생산된 휴대폰들 중 대다수가 이를 지원하지 않고 있으며, PDA와 PC의 경우 CC/PP를 지원하지 않고 있다. 따라서 서버가 프로파일의 분석을 통해 단말기의 특성에 최적화된 콘텐츠를 제공할 수 있다고 하더라도 기존의 단말기를

효과적으로 지원하기 위한 방법이 필요하다.

2.3. PandA

PandA(Portable and Adaptable CC/PP Proxy and Agent)는 다양한 설정 하에서의 CC/PP 성능을 측정하기 위한 원형(prototype) 시스템으로 개발되었다[11].

PandA 브라우저는 화면 크기, 사용 가능한 색상 수, 표현 가능한 이미지 포맷, 기본 프로파일과 표현 가능 마크업 언어를 설정함으로써 클라이언트의 특성을 변경할 수 있으며, 클라이언트의 특성을 CC/PP 프로파일로 표현하기 위하여 자체 어휘를 정의하였다. 표 2는 PandA 어휘에 정의된 항목들이다.

표 2. PandA 어휘
Table 2. The Vocabulary of PandA

컴포넌트	속성	설명
HardwarePlatform	Model	모델 이름 혹은 번호
	ScreenSize	표시 장치 크기
	ScreenDepth	색상 수
	Speaker	음향 지원
SoftwarePlatform	HTMLVersion	HTML 버전
UserPreference	ImageResize	영상 크기
	Image	영상 포맷

PandA는 CC/PP 프로파일을 지원하도록 자체 제작된 PandA 브라우저의 요청만을 처리할 수 있는 등 제약적인 환경을 갖고는 있으나 해당 클라이언트의 특성 프로파일에 기초한 콘텐츠 변환을 통하여 클라이언트 특성에 최적화된 콘텐츠를 제공할 수 있는 방법을 보여준다.

III. CC/PP 프로파일 생성 및 처리

그림 3은 프로파일 생성 및 처리를 위해 본 연구에서 제안하는 CC/PP 프로파일 생성 흐름도이다. 단말기 특성 정보 요청이 들어오면, 먼저 클라이언트 단말기의 요청 헤더를 통해 해당 단말기가 CC/PP 지원 단말기인지 검사한다. 단말기가 CC/PP 프로파일을 지원하면 시스템은 헤더에 기술되어 있는 프로파일 정보를 얻는다.

CC/PP 프로파일을 지원하지 않는 비지원 단말기인 경우, 시스템은 요청 헤더를 파싱하여 유효 정보들을 추출

한다. 그리고 이 정보를 기반으로 CC/PP 프로파일을 생성한다.

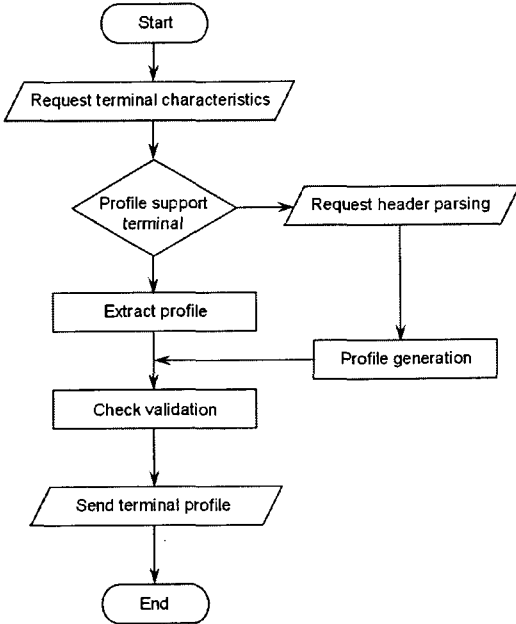


그림 3. CC/PP 프로파일 생성 흐름도

Fig. 3. The flow diagram of CC/PP Profiling System

3.1. 어휘 정의

본 연구에서는 프로파일 생성에 앞서 해당 프로파일이 따라야 할 어휘를 정의하였다. 생성되는 프로파일이 사용하는 어휘는 WAP 포럼에서 정의한 UAPProf 어휘를 기반으로 하였으며, 해당 어휘 중 PandA에서 콘텐츠 변환을 위해 사용한 용어들을 위주로 일반적인 이미지와 콘텐츠를 표현하기 위해 필요한 속성들을 선별하여 표 3과 같이 정의하였다.

어휘의 다양한 속성 중 콘텐츠 변환을 위해 생성하여야 할 최소한의 속성들은 이미지 지원 여부를 나타내는 ImageCapable과 색상 정보를 위한 ColorCapable, BitsPerPixel, 화면에 표시할 콘텐츠의 크기 결정을 위한 ScreenSize 및 표현 형식을 위한 TablesCapable, FramesCapable과 단말기가 지원하는 MIME 형식을 나타내는 CcppAccept가 있다. 이는 단말기에 서비스되는 웹 콘텐츠의 변환을 위해 필요한 최소한의 속성들로서 PandA의 어휘를 기반으로 하였다.

표 3. 본 연구에서 정의한 어휘
Table 3. The proposed vocabulary

컴포넌트	속성	ResolutionRule	설명
HardwarePlatform	ColorCapable	Locked	색상 지원
	BitsPerPixel	Locked	생상 수
	ImageCapable	Locked	영상 지원
	SoundOutputCapable	Locked	음향 지원
	ScreenSize	Locked	표시 장치 크기
BrowserUA	Model	Locked	모델 명/번호
	HtmlVersion	Locked	HTML 버전
	JavaScriptEnabled	Locked	javascript 지원
	JavaScriptVersion	Locked	javascript 버전
	TablesCapable	Locked	표 지원
	FramesCapable	Locked	프레임 지원
	BrowserName	Locked	브라우저 이름
SoftwarePlatform	BrowserVersion	Locked	브라우저 버전
	CcppAccept	Append	MIME 타입
	CcppAccept-Encoding	Append	인코딩
	CcppAccept-Language	Append	지원 언어
WapCharacteristics	CcppAccept-Charset	Append	문자 셋
	WmlVersion	Locked	WML 버전
	WmlScriptVersion	Locked	WMLScript 버전

3.2. 요청 헤더에서의 정보 추출

프로파일을 지원하는 W-HTTP 요청 헤더인 경우 W-HTTP 프로토콜 규격에 따라 해당 요청 헤더를 파싱하여 프로파일 추출을 위한 정보를 획득한다. W-HTTP 프로토콜은 CC/PP 프로파일을 위하여 두 가지의 필드를 사용한다. x-wap-profile 필드는 URI를 기술하여 해당 단말기의 프로파일을 지정하며, x-wap-profile-diff는 URI로 표현된 해당 단말기의 프로파일에 추가되는 추가 특성을 RDF 구문으로 표현한다. 따라서 요청 헤더에서 x-wap-profile로 지정된 URI와 x-wap-profile-diff인 RDF 구문을 추출함으로써 요청 헤더의 정보 추출을 완료한다.

프로파일을 지원하지 않는 HTTP 표준 요청 헤더의 경우 요청 헤더의 User-Agent 필드와 Accept 요청 헤더 필드에서 정보를 추출한다. 이 중 User-Agent 항목에서는 에이전트 명과 버전, 플랫폼을 추출하며, Accept 필드에서는 지원하는 MIME 타입, 언어, 문자 셋, 인코딩을 추출한다. 다음은 마이크로소프트 인터넷 익스플로러의 요청 헤더의 예이다.

```
GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap,
image/jpeg, image/pjpeg,
application/x-shockwave-flash,
application/vnd.ms-excel,
application/vnd.ms-powerpoint,
application/msword, */*
Accept-Language: ko
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1)
Host: localhost
Connection: Keep-Alive
```

Accept, Accept-Language, Accept-Encoding 필드는 SoftwarePlatform 컴포넌트의 속성인 CcppAccept, CcppAccept-Language, CcppAccept-Encoding으로 매핑된다. 이때 Accept 필드의 값 중 모든 MIME 타입에 적용되는 */*는 그 특성상 특성 정보에서 배제된다.

표 4. 요청 헤더 필드 및 대응 속성
Table 4. Request header fields and attributes

요청 헤더 필드	내용	컴포넌트 속성
Accept	MIME 타입	CcppAccept
Accept-Language	지원 언어	CcppAccept-Language
Accept-Charset	문자 셋	CcppAccept-Charset
Accept-Encoding	인코딩 방법	CcppAccept-Encoding
User-Agent	사용자 에이전트 이름	BrowserName
	에이전트 버전	BrowserVersion
	플랫폼	Model

User-Agent 필드에서는 MSIE 6.0과 Windows NT 5.1 값을 추출한다. MSIE 6.0은 BrowserUA의 BrowserName과 BrowserVersion으로 매핑되며, Windows NT 5.1은 HardwarePlatform의 Model로 설정한다. 표 4는 해당 정보가 어떤 속성 요소들로 변환 가능한 지를 보여준다.

3.3. CC/PP 프로파일의 생성

요청 헤더의 분석이 완료된 후 분석된 값들을 기반으로 CC/PP 프로파일을 생성한다. 생성 과정은 크게 두 단

계로 구성되는데, 먼저 요청 헤더에서 추출된 정보를 토대로 추가적인 속성들을 생성하고, 이어 요청 헤더에서 추출된 정보 중 브라우저 이름과 버전을 통해 매핑되는 에이전트 프로파일을 추출하여 병합함으로써 단말기에 대한 프로파일 생성을 완료한다.

3.3.1 추가 정보 추출

요청 헤더에서 추출된 정보는 그 정보 그대로도 단말기의 특성을 표현하지만, 분석을 통해 추가적인 특성 정보를 추출할 수 있다. 추가 정보 생성을 위해서는 요청 헤더에서 추출된 모든 정보를 대상으로 생성 규칙이 적용되는지를 검사하여 해당하는 정보에 따른 추가 정보를 생성한다.

표 5 영상 포맷별 추가 생성 정보
Table 5. Additional coloring information

영상 포맷	색상	ColorCapable
wbmp	black, white	No
bmp	b&w, gray, color	No
jpg	color	Yes
png	color	Yes
nbmp	black, white	No
gif	gray, color	No

User-Agent 필드에 의한 정보 추출 외에 Accept 필드에서 제공되는 지원 MIME 타입 또한 추가 정보의 추출에 이용된다. MIME 타입 내의 주요 정보로는 먼저 영상 포맷을 통한 해당 단말기의 색상 지원 여부와 지원 마크업 언어를 통한 프레임 및 테이블의 지원 여부를 알 수 있다. 표 5는 영상 포맷별 색상 지원 여부에 관한 정보이다.

추가 정보 생성 규칙을 통해 생성된 정보는 기존 헤더 정보에 추가되어 프로파일로 생성된다. 생성된 프로파일은 다음 절에서 설명하는 에이전트 매핑 규칙을 통한 프로파일과 결합되어 해당 단말기의 최종 프로파일로 생성된다.

3.3.2 하드웨어 정보 추출

사용자 에이전트 매핑 규칙은 요청 헤더에 대한 추가 정보 생성 규칙을 통하여 알아내기 힘든 단말기 하드웨어 특성을 기술하기 위해 사용한다. 이는 추가 정보 생성 규칙의 적용 대상인 요청 헤더의 정보 자체가 해당 단말기

에서 사용되는 브라우저 위주의 특성을 주로 기술하고 있기 때문에 단말기 자체가 갖는 하드웨어 특성을 생성하기 어렵다는 단점을 보완하기 위한 것이다.

매핑은 User-Agent 필드 상에서 나타나는 브라우저 이름과 모델명 각각에 대해 매핑되는 프로파일이 있는지 검사하고, 매핑 프로파일이 있을 때 해당 프로파일을 추출하여 사용한다. 여기서 브라우저 이름과 모델명을 구분하여 매핑시키는 이유는 해당 플랫폼이 갖는 특성과 그 플랫폼에서 동작하는 브라우저가 갖는 특성이 서로 독립적이기 때문이다. 이를 위해 브라우저 명에 매핑되는 프로파일인 경우 추가 정보 생성 규칙에서 생성 가능한 정보 이외의 BrowserUA 컴포넌트 위주의 특성을 기술하고, 모델명에 매핑되는 프로파일은 해당 단말기의 하드웨어적 특성인 HardwarePlatform 컴포넌트 위주로 특성을 기술한다. 다음은 마이크로소프트 인터넷 익스플로러 버전 6에 해당하는 프로파일 msie_6.rdf의 일부이다. 이 경우 인터넷 익스플로러는 Html 4.0과 JavaScript 1.3 버전을 지원한다.

```
<rdf:Description rdf:ID="BrowserUA">
  <rdf:type rdf:resource=
    "http://www.wapforum.org/profiles/UAPROF/ccppschem-20010430#BrowserUA"/>
  <prf:BrowserName>Microsoft</prf:BrowserName>
  <prf:BrowserVersion>6.0</prf:BrowserVersion>
  <prf:HtmlVersion>4.0</prf:HtmlVersion>
  <prf:JavaScriptEnabled>Yes</prf:JavaScriptEnabled>
  <prf:JavaScriptVersion>1.3</prf:JavaScriptVersion>
</rdf:Description>
```

3.3.3. 프로파일 합성

단말기의 최종 프로파일을 생성하기 위하여 요청 헤더에서 추출된 정보와 추가 정보 추출, 하드웨어 정보를 합성한다. 프로파일의 합성은 요청 헤더에서 추출된 정보와 추가 정보를 토대로 생성한 프로파일을 기본으로 하여, 여기에 하드웨어 정보를 합하여 최종 프로파일을 생성한다. 프로파일의 합성 시 같은 이름의 속성이 중복되어 나타날 경우는 어휘에 정의된 분석 규칙에 따라 해당 속성

의 값을 결정한다.

IV. 구현 및 평가

이번 장에서는 CC/PP 프로파일 생성 방법을 구현한 프로파일 생성 시스템의 전체 구조와 이를 구성하는 주요 구성부분 및 동작 방식에 대해서 설명하고, 각 단말기 별로 프로파일 생성에 대해 테스트하고 결과를 분석한다.

4.1. CC/PP 프로파일 생성 시스템 구성도

프로파일 생성 시스템의 전체 구조는 그림 4와 같이 네 부분으로 구성된다. 프로파일 매니저(Profile Manager)는 프로파일에 관한 정보 요청을 받고 생성된 프로파일을 반환하는 역할을 담당한다. 이때 프로파일 검증기(Profile Validator)를 이용하여 프로파일의 유효성 여부를 검사한다.

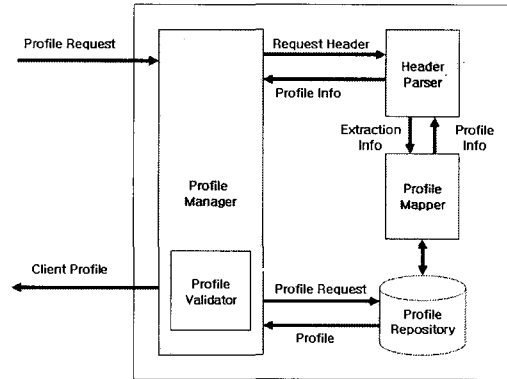


그림 4. CC/PP 프로파일 생성 시스템 구성도
Fig. 4. The architecture of profiling system

헤더 분석기(Header Parser)는 요청 헤더의 분석을 통해 요청 단말기의 프로파일 지원 여부를 판단한다. W-HTTP 프로토콜의 경우 프로파일 정보를 추출하여 반환하며, HTTP 프로토콜의 경우 HTTP 요청 헤더를 분석하여 프로파일 생성에 필요한 정보를 추출한다. 프로파일 매퍼(Profile Mapper)는 헤더 분석기에 의해 추출된 정보를 기반으로 추가 생성 정보 및 하드웨어 정보 추출을 통하여 적절한 단말기 프로파일을 생성하며, 프로파일 저장소(Profile Repository)는 프로파일 생성을 위한 기본 정보들의 저장 및 관리를 담당한다.

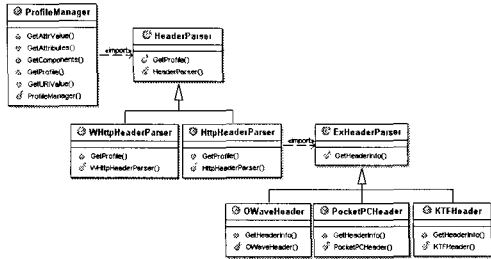


그림 5. ProfileManager/HeaderParser 클래스
Fig. 5. The class diagram for ProfileManager/HeaderParser

그림 5는 프로파일 매니저와 헤더 분석기의 클래스 다이어그램이다. 헤더 분석기 모듈은 프로파일 매니저로부터 전달받은 요청 헤더를 분석하여 CC/PP를 지원하는 W-HTTP 프로토콜인 경우 WHttpHeaderParser로 헤더를 파싱하고, CC/PP를 지원하지 않는 일반 HTTP 헤더인 경우에는 HttpHeaderParser를 이용하여 헤더를 파싱한다. WHttpHeaderParser는 W-HTTP 프로토콜을 따를 때는 해당 헤더의 프로파일 URI를 추출하여 프로파일 매퍼에게 전달하여 프로파일을 추출할 수 있도록 한다. HttpHeaderParser는 표준 HTTP 헤더를 파싱한다.

파싱된 요청 헤더 정보는 헤더 분석기에서 프로파일 매퍼에게 전달되어 CC/PP 프로파일 생성을 위해 사용된다. 그림 6은 프로파일 매퍼 및 프로파일 생성을 위해 사용되는 클래스에 대한 다이어그램이다. 프로파일 매퍼에게 전달된 헤더 정보는 MakeExtInfo 메소드에 의해 추가 정보를 추출하고 MakeAcceptRequest 메소드에 의해 필요 정보가 추출되어 MakeHeaderProfile 메소드에 의해 프로파일로 변환된다. 이어 MakeMappingProfile 메소드에 의해 추출한 사용자 에이전트 매핑 프로파일과 결합되어 프로파일로 생성되며, 프로파일 매니저에서 유효성 검사를 마친 후 최종 CC/PP 프로파일로 사용된다.

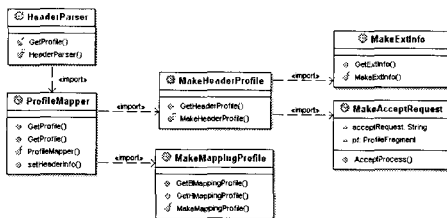


그림 6. ProfileMapper 및 관련 클래스들
Fig. 6. ProfileMapper and related classes

4.2. 프로파일 생성 실험 결과

프로파일 생성 결과를 분석하기 위하여 프로파일 생성 시스템에서 추출 및 생성된 프로파일은 과상 과정을 거쳐 컴포넌트 별로 정렬하여 출력하였다. 먼저 CC/PP 지원 단말기의 요청에 대하여 요청 헤더에 기술된 프로파일을 추출하여 처리하는지에 대해 테스트하였다. CC/PP 지원 단말기에 대한 테스트는 Openwave SDK 6.2.2 에뮬레이터를 이용하였다. 그림 7은 Openwave 에뮬레이터의 W-HTTP 요청 헤더의 예이다.

```
host = localhost:8080
x-up-udp-session-headers = User-Agent, Accept-Charset, Accept-Language,
user-agent = OPWV-SDK/62 UP.Browser/6.2.2.1.208 (GUI) MMP/2.0
accept-charset = ks_c_5601
accept-language = ko
x-wap-profile = "http://developer.openwave.com/uaprof/OPWV3DK62.xml"
accept-encoding = deflate, gzip
x-up-proxy-enable-trust = 1
x-up-devcap-cc = 1
accept = application/smil, application/vnd.phoncom.mmc+xml, application/
connection = close
```

그림 7. Openwave SDK 에뮬레이터 요청 헤더
Figure 7. The header information of Openwave

요청 헤더에 따르면 http://developer.openwave.com/uaprof/OPWV3DK62.xml이라는 프로파일을 이용한다. 그림 8은 프로파일 중 하드웨어 플랫폼 요소의 모습이다.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:prf="http://www.wapforum.org/profiles/UAPROF/cppscheme-20020710#"
xmlns:mms="http://www.wapforum.org/profiles/MMS/cppscheme-20010111#">
  <rdf:Description rdf:ID="UAPROF">
    <prf:component>
      <prf:Description rdf:ID="HardwarePlatform">
        <rdf:type rdfs:resource="http://www.wapforum.org/profiles/UAPROF/cppscheme-20020710#HardwarePlatform" />
        <prf:ScreenWidth>24</prf:ScreenWidth>
        <prf:ColorCapable>Yes</prf:ColorCapable>
        <prf:CPUS>Pentium</prf:CPU>
        <prf:ImageCapable>Yes</prf:ImageCapable>
      </prf:Description>
      <prf:Keyboard>PhoneKeypad</prf:Keyboard>
      <prf:Model>OPWV-SDK/62</prf:Model>
      <prf:NumberOfSoftKeys>2</prf:NumberOfSoftKeys>
      <prf:OutputCharsets>
        <prf:PixelFormat>1x1</prf:PixelFormat>
        <prf:PrintingResolutionCharacter>prf:PrintingResolution</prf:PrintingResolution>
        <prf:ScreenSize>120x160</prf:ScreenSize>
        <prf:ScreenSizeChar>12x8</prf:ScreenSizeChar>
        <prf:StandardFontProportional>Yes</prf:StandardFontProportional>
        <prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
        <prf:TextInputCapable>Yes</prf:TextInputCapable>
        <prf:Vendor>Openwave</prf:Vendor>
        <prf:VoiceInputCapable>No</prf:VoiceInputCapable>
      </prf:Description>
    </prf:component>
    <prf:component>
      <prf:component>
        <prf:component>
          <prf:component>
            <prf:component>
              <prf:Description>
                </rdf:RDF>
```

그림 8. Openwave 에뮬레이터 CC/PP 프로파일
Fig. 8. CC/PP profile of Openwave emulator

그림 9는 요청 헤더에서 프로파일을 추출하여 파싱한 결과 중 하드웨어 플랫폼 컴포넌트 부분으로서, CC/PP 프로파일 시스템이 헤더 분석을 통하여 단말기가 사용하는 CC/PP 프로파일을 얻은 결과이다.

```
Component [ HardwarePlatform ]
  BitsPerPixel = 24
  ColorCapable = true
  OutputCharSet = [johab, euc-kr, ibm062, windows-1256, iso-8859-3,
  ScreenSizeChar = 12x8
  InputCharSet = [johab, euc-kr, ibm062, windows-1256, iso-8859-3,
  CPU = Pentium
  SoundOutputCapable = true
  ImageCapable = true
  PointingResolution = Character
  ScreenSize = 120x160
  NumberOfSoftKeys = 2
  Keyboard = PhoneKeypad
  TextInputCapable = true
  VoiceInputCapable = false
  Model = OPVV-SDK/62
  Vendor = Openwave
  PixelAspectRatio = 1x1
  StandardFontProportional = true
```

그림 9. CC/PP 프로파일 정보 획득 결과
Fig. 9. The resulting CC/PP profile information

생성된 프로파일의 검증은 위해 Panda에서 콘텐츠 변환을 위해 사용한 속성들을 기반으로 마크 업 언어, 이미지, 색상 정보 위주의 몇 가지 속성 요소를 정하여 속성들을 생성할 수 있는지를 검사하였다. 테스트에는 CC/PP 지원 단말기인 Openwave와 일반 데스크톱 용 브라우저인 마이크로소프트 인터넷 익스플로러 6.0 버전을 이용하였다.

표 6은 클라이언트에 대해 생성된 프로파일 검사 결과이다. 결과에 의하면 CC/PP 지원 단말기인 경우 해당 단말기의 프로파일을 파싱하여 모든 속성 테스트 요소에 대한 값을 추출 할 수 있었다. 마이크로소프트 인터넷 익스플로러 6.0과 같이 표준 요청 헤더만을 사용하는 경우에는, BitsPer-Pixel, ScreenSize와 같이 단말기에 종속적인 정보를 생성할 수 없었는데, 이는 해당 속성이 인터넷 익스플로러 6.0 브라우저가 동작하는 윈도우 플랫폼의 성격 상 하드웨어 종속적인 속성이면서 또한 사용자의 기호에 속하는 속성이어서 매핑 프로파일에 고정된 값을 적용하기가 어렵기 때문이었다. 이러한 경우에는 매핑 프로파일 상에 해당 속성이 가질 수 있는 기본 값을 기술함으로써 문제를 해결할 수 있을 것이다.

표 6. CC/PP 프로파일 속성 생성 결과
Table 6. The processing result for two kinds of client devices

Attribute/device	Openwave	MSIE 6.0
ColorCapable	O	O
ImageCapable	O	O
BitsPerPixel	O	X
ScreenSize	O	X
TablesCapable	O	O
FramesCapable	O	O
CcppAccept	O	O

V. 결론

본 논문에서는 유비쿼터스 환경에서 사용되는 단말기들에 대하여, W-HTTP 프로토콜로 자신의 특성 프로파일을 제공하는 CC/PP 지원 단말기뿐만 아니라, CC/PP를 지원하지 않는 HTTP 기반 클라이언트에 대해서도 단말기 특성 정보를 추출하고 분석할 수 있도록 하는 CC/PP 프로파일 생성 시스템을 설계하고 구현하였다.

이를 위하여 CC/PP 지원 단말기인 경우 W-HTTP 요청 헤더의 단말기 프로파일 정보를 이용하여 해당 단말기의 프로파일을 추출하였고, 비지원 단말기인 경우 단말기의 서비스 HTTP 요청 헤더 정보를 분석하여 분석된 요청 헤더 정보를 바탕으로 추가적인 특성 정보를 생성함으로써 CC/PP 프로파일을 생성하였다.

본 논문에서 제안한 CC/PP 프로파일 생성 방법은 단말기의 CC/PP 지원 여부 관계를 떠나 서버에서 클라이언트 단말기의 특성을 파악할 수 있도록 함으로써, 앞으로 다양한 이 기종 단말기들의 특성을 추출 및 분석하고, 이를 기반으로 해당 단말기에 적합한 서비스를 제공할 수 있는 기반 기술로 활용할 수 있을 것이다.

참고문헌

- [1] Uwe Hansmann, Lothar Merk, Martin S.Nicklous, Thomas Stober, "Ubiquitous Computing Handbook," Springer, 2001.
- [2] 김종우, 한승조, "디지털 콘텐츠 공유를 위한 개선된 Kerberos P2P 인증시스템 설계, 한국해양정보통신학회논문지, 제8권, 제7호, pp.1516-1523, 2004.
- [3] Mark Butler, Fabio Giannetti, Roger Gimson, Tony Wiley, "Device Independence and the Web," IEEE Internet Computing, pp.81-86, 2002.
- [4] Device Independence Working Group, "Delivery Context Overview for Device Independence," <http://www.w3.org/TR/2002/WD-di-dco-20021213/>, 2002.
- [5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, "Hypertext Transfer Protocol - HTTP/1.1," <http://www.w3.org/Protocols/rfc2616/rfc2616.html>, 2004.
- [6] Graham Klyne, Franklin Reynolds, Chris Woodrow, Hidetaka Ohto, Johan Hjelm, Mark H. Butler, Luu Tran,

"Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0," <http://www.w3.org/TR/CCPP-struct-vocab/>, 2004.

- [7] L. Tran, "CC/PP: Structure and Vocabularies Implementation Report," Sun Microsystems, 2003.
- [8] H. Ohto, J. Hjelm, "CC/PP exchange protocol based on HTTP Extension Framework", <http://www.w3.org/TR/NOTE-CCPPexchange>, 2005.
- [9] H. Ohto, L. Suryanarayana, J. Hjelm, "CC/PP Implementors Guide: Privacy and Protocols," <http://www.w3.org/TR/CCPP-trust/>, 2005.
- [10] R. Eisinger, M.G. Manzato, R. Goularte, "Devices descriptions for context-based content adaptation," Web Congress, LA-WEB, pp.9-15, 2005.
- [11] Q. H. Mahmoud, Wang Zhixin, "Customizing and delivering mobile services using software agents and CC/PP," Consumer Communications and Networking Conference, pp.1114-1118, 2006.

저자소개

변 영 철(Yung-Cheol Byun)



1993년 제주대학교 정보공학과 학사
 1995년 연세대학교 컴퓨터과학과 석사
 2001년 연세대학교 컴퓨터공학과 박사
 2001년 한국전자통신연구원 선임 연구원

2002년~현 제주대학교 공과대학 컴퓨터공학전공 교수
 ※ 관심분야 : 패턴인식, 시맨틱 웹, 지능형 컴퓨팅, 유비쿼터스 미들웨어

강 철 응(Chul-Ung Kang)



1993년 제주대학교 기관공학과
 1996년 일본 나가사키대학 기계시스템공학석사
 1999년 일본 나가사키대학 해양생산공학박사

1999년 일본 나가사키대학 기계시스템공학과 조수
 2001년~현 제주대학교 메카트로닉스공학과 교수
 ※ 관심분야 : 로봇공학, 인공지능, 환경인식, 신호처리, 웹페어 디자인

이 상 준(Sang-Joon Lee)



1984년 중앙대학교 컴퓨터공학과 학사
 1989년 중앙대학교 컴퓨터공학과 석사
 1992년 중앙대학교 컴퓨터공학과 박사
 1993~현 제주대학교 공과대학 컴퓨터공학전공 교수

※ 관심분야 : 디지털 콘텐츠 서비스, 지식기반 시스템, 휴리스틱 알고리즘