

상황인식을 지원하는 미들웨어를 위한 메타서버 모델†

Meta-server Model for Middleware Supporting for Context Awareness

이서정*, 황병연**, 윤용익***

Seo-Jeong Lee, Byung-Yeon Hwang, Yong-Ik Yoon

요약 이동통신 환경의 분산 응용에서는 사용자나 단말기의 위치 이동을 파악하여 서비스를 지속할 수 있는 모바일 기술이 요구된다. 이 문제를 해결하기 위해 상황 인식 미들웨어를 이용해 모바일 네트워크 상에서 사용자 혹은 사용자 디바이스의 CPU속도, 메모리 그리고 가용배터리의 변화를 적절히 지원하기 위한 연구가 많이 진행되고 있다. 여기에는 사용자와 디바이스에 대한 프로파일, 상황의 메타데이터가 필수적이다. 본 논문에서는 상황인식의 동적 반영을 지원하는 미들웨어 서비스에 필요한 메타서버를 모델링하고 구축했다. 메타데이터는 프로파일, 정책, 사용자 선호도로 구성되며, 메타 서버는 미들웨어의 요청에 따라 서비스 가능한가를 판별 후, 결과를 돌려준다. 멀티미디어 스트림을 제공하는 이동통신 환경에서 사용자나 통신 단말기의 상황이 변동되는 시나리오에 적용해 보았다.

Abstract An increasing number of distributed applications will be achieved with mobile technology. These applications face temporary loss of network connectivity when they move. They need to discover other hosts in an ad-hoc manner, and they are likely to have scarce resources including CPU speed, memory and battery power. Software engineers building mobile applications need to use a suitable middleware that resolves these problems and offers appropriate support for developing mobile applications. In this paper, we describe the meta-server building for middleware that addresses reflective context awareness and present usability with demonstration. Metadata is consist of user configuration, device configuration, user context, device context and dynamic image metadata. When middleware send a saving or retrieval request to meta-server, it returns messages to middleware after the verification of the request. This meta-server has the application for multimedia stream services with context awareness.

주요어 : 모바일 컴퓨팅, 상황인식, 동적반영, 메타데이터, 형상데이터, 상황데이터

Keywords : Mobile Computing, Context Awareness, Reflection, Metadata, Configuration data, Context data

1. 서론

최근 정보통신서비스의 글로벌화와 기술혁신에 따

라 각광받고 있는 유비쿼터스 컴퓨팅(ubiquitous computing)은 컴퓨팅, 통신, 접속방식, 제공하는 콘텐츠 및 사람이 컴퓨터의 존재를 인지하지 않도록 조용

† 본 논문은 과학기술부 한국과학재단 지정 한국항공대학교 부설 인터넷정보검색연구센터의 지원에 의함

* 숙명여자대학교 정보과학부 멀티미디어과학과 연구원

sjlee815@sookmyung.ac.kr

** 가톨릭대학교 컴퓨터정보공학부 교수

byhwang@catholic.ac.kr

*** 숙명여자대학교 정보과학부 멀티미디어과학과 부교수

yiyeon@sookmyung.ac.kr

히 처리하는 특성 (5C : Computing, Communication, Connectivity, Contents, Calm)을 이용하여 5Any (Anytime, Anywhere, Any network, Any device, Any service)를 지향하고 있다. 이러한 사용자의 유비쿼터스 서비스에 대한 요구를 만족시키기 위한 접근으로 상황인식(context awareness) 컴퓨팅 기술이 거론되고 있다.

상황인식 컴퓨팅은 컴퓨터간의 의사소통의 전달이 인간 세계와 같은 능력을 갖도록 한다는 동기와 목표를 갖고 있다. 최근 상황인식 컴퓨팅은 "사용자의 작업과 관련 있는 적절한 정보 또는 서비스를 사용자에게 제공하는 과정에서 '상황(context)'을 사용하는 경우"로 정의한다[1]. 일반적 상황 정보는 다음과 같으며, 상세한 내역은 시스템에 따라 구체화 된다.

- 사용자 상황
- 물리적 환경 상황
- 컴퓨팅 시스템 상황
- 사용자-컴퓨터 상호작용 이력
- 기타

이동통신 환경에서 사용자의 위치 정보는 상황인식 서비스 제공을 위해 가장 중요하고 필요한 상황 정보이다. 현재 GPS나 이동통신망 기반 위치인식 시스템을 이용하여 공공 안전 서비스, 위치 추적서비스, 항법 서비스, 정보 제공 서비스 등과 같은 다양한 위치기반 서비스가 제공된다[2]. 차세대 이동통신은 위치인식 시스템이 커버하지 못하는 실내에서 정밀한 위치 인식이 가능하게 될 것이다 [3].

상황인식 시스템을 구현상의 특징은 개발 대상 도메인(domain)과 특정 응용에 따라 세분화 되고 종속적이지만 일반적 특징은 다음 세 가지이다[4].

- 사용자에게 정보와 서비스 제공(presentation)
- 사용자를 위한 서비스의 자동 실행(execution)
- 이후 검색을 위한 상황 정보의 표시(tagging)

이동통신 환경에서 사용자의 위치 정보는 상황인식 서비스 제공을 위해 가장 중요하고 필요한 상황 정보이다. 현재 GPS나 이동통신망 기반 위치인식 시스템을 이용하여 공공 안전 서비스, 위치 추적서비스, 항법 서비스, 정보 제공 서비스 등과 같은 다양한 위치기반 서비스가 제공된다 또한, 위치인식 시스템이 커버하지

못하는 실내에서 정밀한 위치 인식을 하기위한 연구도 진행되어 언제 어디서나 위치인식과 네트워크이 가능한 차세대 이동 통신 환경에서는 지금보다 훨씬 다양한 위치기반 서비스가 제공될 수 있다 [2,3].

이동통신 환경에서 상황인식 서비스는 최근 들어 서비스의 속도와 품질을 높이기 위해 미들웨어로 구축이 대부분이다. 여기에 서비스를 제공하기 위해 필요한 정보는 메타데이터의 형태로 관리한다. 메타데이터는 기본적으로 사용자, 통신 장비, 그리고 주변 환경의 정보가 되며, 이를 가지고 어떤 서비스를 어떻게 제공할지를 미들웨어에서 결정 후, 서비스를 수행하는 방식으로 운영한다.

메타데이터의 형태와 종류 그리고 관리는 응용 서비스에 따라 다르지만 몇 가지 고려사항을 정리하면 다음과 같다.

첫째, 디바이스의 정보 및 위치정보, 서비스를 제공받기 위한 사용자의 정보 및 위치정보 등을 데이터베이스로 구축하고, 저장된 데이터를 관리 및 제공할 수 있는 시스템의 구축이다[5].

또한, 저장된 데이터는 변경사항에 대해 적응성을 가져야 한다. 멀티미디어 표현의 변경, 사용자 인터페이스의 변경 그리고 사용자의 프로파일(profile)의 변경이 적용의 대상이 된다[6]. 표현의 변경은 공간·시간적 변경에 따라 최적 형태로 크기나 압축 정도의 변환과 네트워크 QoS(Quality of Service)와 사용자 프로파일에 따른 변환을 말한다. 사용자 인터페이스 적용 모듈은 웹 또는 WAP(Wireless Application Protocol) 브라우저, SMS(Short Message Service) 외에도 음성기반, 자바기반의 그래픽 사용자 인터페이스에서 사용될 수 있으며, 프로토콜 핸들러, UI 어댑터, 내부 서비스 그리고 세션 관리로 구성되어 있다[7,8]. 사용자 프로파일의 적용은 명시적으로 선언된 데이터의 사용뿐만 아니라 사용자의 행동으로부터 수집된 추론 데이터의 사용에 대해서도 사용자의 선호도를 모델링하기 위해 적용된다.

한편, 모바일 클라이언트 응용은 클라이언트의 현재 위치에서 가능한 응용 서비스를 찾아 상호 운용해야 한다. 이 서비스는 RMI(Remote Method Invocation)나 publish-subscribe의 미들웨어 형태로 개발되고

UPnP(Ugly Plug n Pray)와 SLP(Service Location Protocol)와 같은 서비스 프로토콜을 사용하므로, 모바일 클라이언트의 적응성(adaptation)과 반영성(Reflection)의 지원은 미들웨어 수준에서 가능해야한다[9].

또한, 반영성이 보장된 모바일 응용의 고품질 서비스를 위해서는 메타데이터의 일관성(consistency)이 만족되어야 한다. 미들웨어는 프로파일을 저장하기 전, 정보의 일관성을 검증해야한다. 이를 위해, 파서(parser)는 XML 인스턴스를 검증한다. XML 스키마는 매우 자세하고 복잡한 문법의 명세를 지원하므로 파서의 검증 결과는 아주 정확할 수 있다[10].

본 논문에서는 이동통신 환경에서 상황 인식 서비스의 만족도를 높이기 위한 메타데이터의 설계를 제안하고 미들웨어와의 연동을 구현한 메타관리자를 설명한다. 미들웨어는 사용자에게 서비스를 제공하기 시작하여 위치를 이동하거나, 통신장비를 PDA에서 휴대폰으로 바꾸었을 경우 변경된 상황정보를 센서로부터 수집하여 적절한 서비스로 변경·제공하는 간단한 시나리오를 생각할 수 있다. 메타데이터는 미들웨어가 제공하는 서비스를 지원하기 위한 기본 정보(configuration), 변경된 상황을 위해 적절한 서비스를 판단하기 위한 상황정보(context), 정해진 정책(policy), 사용자의 선호도(preference)로 구성되며, 미들웨어의 요청이 발생하는 경우, 저장(save) 또는 검색(retrieve)의 기능을 SQL 쿼리로 수행한다. 미들웨어와의 통신은 XML메시지로 이루어진다.

본 논문의 구성은 다음과 같다. 2장에서 본 연구와 관련된 연구를 설명하고, 3장에서 본 논문에서 제안하는 메타데이터의 설계를 설명한다. 4장은 구현결과를 설명한다.

2. 관련 연구

이 장에서는 모바일 컴퓨팅 환경의 미들웨어를 설계하기 위한 반영성과 메타데이터의 표현에 관한 연구로 메타데이터의 시멘틱 구조에 관한 연구와 CARMEN 아키텍처를 설명한다. 위치기반 상황인식 서비스는 모바일 컴퓨팅 환경에서 수시로 이동하는

단말기와 사용자의 위치를 인식하여 다양한 위치 인식 응용에 활용할 수 있도록 하는 것으로, MS의 UPnP(Universal Plug and Play), Sun의 Jini, Oracle의 9iASWE(Oracle9i Advanced Search Wireless Enabling), CMU의 Coda, UCL의 XMIDDLE(Information Sharing Middleware for Mobile Environment) 등이 있다[11].

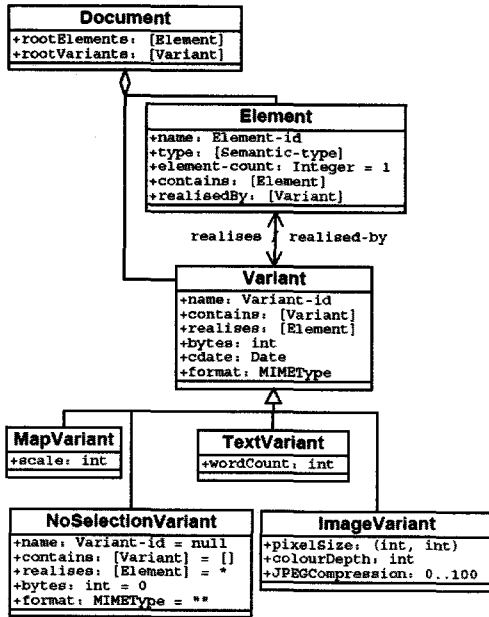
이 중, XMIDDLE은 LIME모델의 데이터 구조 부문을 보완하여 데이터의 반복(replication)과 일치(reconciliation)에 중점을 둔 모델이다[10]. 하지만, XMIDDLE의 개념은 정적인 데이터와 동적인 데이터를 분리하여 Application Profile과 Context로 표현하는 간단한 구조의 개념이지만, Reflective API는 미들웨어 코어나 서비스로 전환하는 메커니즘을 제공하지 못한다.

이 절에서는 실제 모델을 구현한 연구에서 메타데이터를 다루는 방법과 장단점을 살펴본다.

2.1 메타데이터의 시멘틱 구조

상황반영의 모바일 응용을 구현하기 위한 메타데이터는 그 양이 많고 복잡한 경우 오히려 모바일 컴퓨팅의 성능을 저하시키는 요인이 될 수 있음에 착안하여, [12]에서는 메타데이터를 구조화하여 시멘틱에 따라 엘리먼트(element)로 구분하는 구조를 제안했다. 엘리먼트는 구조표현의 다양성을 제공하는 배리언트(variant)중의 하나 현실화(realize)된다. <그림 1>은 이러한 개념을 설명하는 클래스 모델을 나타낸다.

하나의 배리언트는 같이 표현해야하는 데이터를 포함한 배리언트에 종속되어 있으며, 루트 배리언트는 상황에 따라 적절한 하위 배리언트를 포함하여 엘리먼트가 된다. 예를 들어, 사용자가 핸드폰에서 PDA로 서비스 요청을 변경했을 경우, 또는 지도, 이미지 또는 텍스트의 선택사양 중 이미지를 선택했을 경우 등 실제 실행될 때는 요청한 배리언트만 사용되므로 효율적으로 메타데이터를 관리할 수 있는 환경을 제공한다. 예를 들어, <그림 1>은 Map, Text, Image 그리고 NoSelection의 네가지 배리언트를 갖는 구조를 보여준다. 이 중 하나만 현실화(realize)된다.



<그림 1> 배리언트를 갖는 메타데이터 모델

이 방법의 응용으로는 웹사이트의 구조를 표현한 것 과 지도서버(Map server)의 구축이 있는데 웹사이트의 경우, 구조를 유지하기 위한 오버헤드가 상대적으로 더 크고, 지도서버와 같이 많은 양의 데이터와 구조를 유지해야하는 경우 더 효과적이다. 즉, 이 방법은 메타데이터 구조의 유연성을 지원하는 장점을 가졌고 구조가 복잡하거나 변동되기 쉬운 환경에서 활용할 경우 매우 효과적이다. 하지만, 배리언트 관리 구조의 복잡도에 따라 발생하는 응답지연은 오버헤드로 작용하며, 사용자에게 보여져야하는 상황정보를 어떻게 다룰 것인가에 초점이 맞추어져 다른 상황정보의 인식에 대한 고려는 부족하다.

2.2 CARMEN 아키텍처

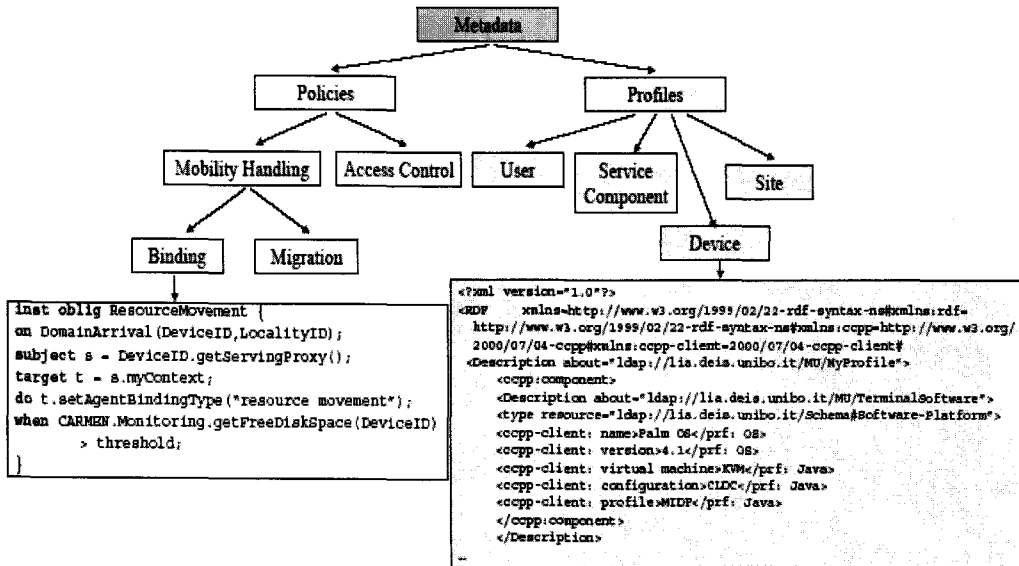
CARMEN(Context-Aware Resource Management Environment) 무선 인터넷 환경에서 상황 변화에 따라 서비스가 자동으로 변환되는 것을 목표로 고안한

상황인식 자원관리를 위한 미들웨어다[13] 전형적으로 메타데이터를 이용하여 프로파일(profile)과 정책(policy)를 고려한다는 점에서 메타매니저의 독립된 역할을 고려한 연구이다.

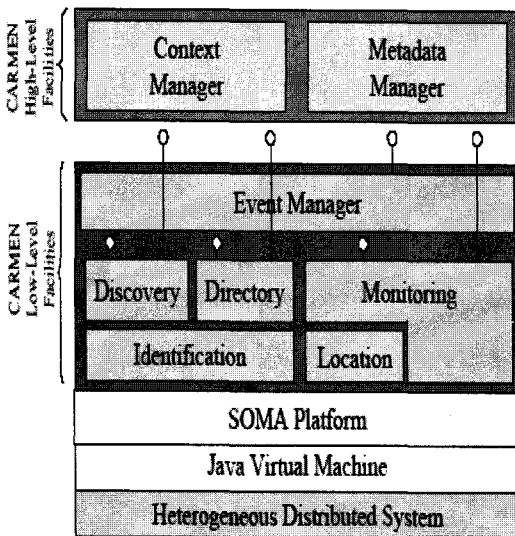
<그림 2>는 CARMEN에서 사용하는 메타데이터의 개념과 간단한 적용이다. 사용자(user), 서비스 컴포넌트(Service component), 디바이스(Device), 사이트(Site)를 포함하는 프로파일(profile)과 액세스 컨트롤(Access control)과 모빌리티핸들링(Mobility Handling)을 위한 바인딩(binding)과 이식(Migration)으로 구성되어 있다.

메타데이터 매니저(MM:Metadata Manager)는 메타데이터를 이용해 프로파일(profile) 정보를 명세하고 변경하는 작업과 액세스 컨트롤과 모빌리티 핸들링을 위한 정책을 지원한다. <그림 3>은 CARMEN의 아키텍처를 보여준다. MM과 컨텍스트 매니저(CM: Context Manager)는 상위수준(high-level) 미들웨어를 구성한다. MM은 CARMEN에서 지원되는 프로파일과 정책 정보에 대해 명세, 수정, 정확성 검사, 설치 및 평가를 지원한다. CM은 CARMEN클라이언트의 상황을 동적으로 결정하여, 상황에 포함되어있는 자원에 접근, 클라이언트의 요구를 지원한다. CARMEN의 하위수준(low-level) 미들웨어는 무선 환경의 클라이언트에 서비스하기 위해 고려해야하는 일반적인 기능들로 구성된다. 이 중 이벤트 매니저(EM:Event Manager)는 모니터링(Monitoring), 디스커버리(Discovery), 그리고 디렉토리(Directory) 등을 통해 수집된 이벤트를 연관된 정책에 바인딩하는 기능을 제공한다. 예를 들어, 현재 특정위치의 서비스 구역에 IEEE 802.11b를 사용하는 무선 단말기를 모니터링하여 EM에게 전달한다. EM은 수집 정보를 분류하여 적절한 프로파일과 정책을 판단할 수 있는 MM과 CM으로 이벤트를 전달한다.

이 방법은 메타데이터를 이용한 상황인식 메커니즘을 제공하기 위해 프로파일과 정책을 활용하는 전형적인 연구로써 모바일 뉴스 서비스에서 좋은 결과를 보였다. 메타데이터매니저는 미들웨어에 포함되어있는 구조로 서비스를 제공할 클라이언트나 관리할 메타데이터의 로드(load)가 많아지는 경우, QoS(Quality of Service)면에서 문제가 발생한다.



<그림 2> CARMEN 메타데이터

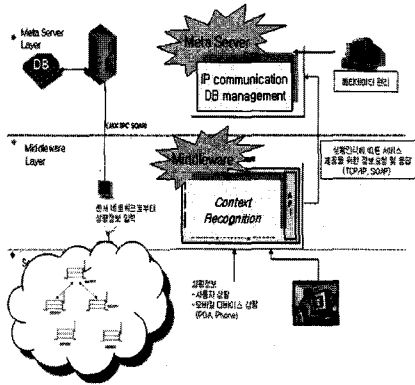


<그림 3> CARMEN 아키텍처

3. 상황 인식을 지원하는 메타서버 모델링

본 논문에서 제안하는 상황인식을 지원하는 메타서버는 미들웨어가 센서를 통해 입력된 사용자나 PDA와 휴대폰과 같은 이동통신 단말기의 변동된 상황을 인식하여 적절한 처리를 하기 위해 필요한 정보를 제공하는 기능을 갖는다. 또한, 유비쿼터스 컴퓨팅의 특성을 고려하고 미들웨어의 재형상(reconfiguration)을 통해 보다 유연한 모바일 응용 서비스를 제공하는 미들웨어의 반영성 부분을 처리하는 역할을 한다.

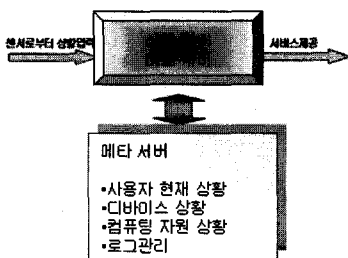
기존 메타 매니지 또는 메타데이터 매니저는 미들웨어 내부에 포함되어 있어 제한적인 상황에서는 성능이 좋을 수 있지만 사용자와 이동통신 환경이 확장되는데 대한 적응성이 떨어진다는 점을 고려하여 본 연구에서는 메타데이터 관리를 위한 서버를 별도로 구성했다. <그림 4>는 제안하는 미들웨어가 적용될 상황인식서비스의 전체 아키텍처를 보여준다.



<그림 4> 상황인식 서비스 아키텍처

미들웨어는 센서 네트워크로부터 상황정보를 입력 받아 TCP/IP와 SOAP을 통해 XML형태로 메타 서버에 정보의 입력과 검색 요청을 한다. 메타 서버는 전송된 메시지를 파싱 후, 미들웨어의 요청을 처리 후, 결과를 같은 방식으로 미들웨어에게 돌려준다.

메타서버에서 다룰 상황 정보와 관리 정보는 <그림 5>와 같이 사용자의 현재 상태, 물리적인 환경의 상태, 컴퓨팅 자원의 상태, 기존 상황정보 제공 내역을 포함한다. 즉, 미들웨어는 사용자, 물리적 환경 그리고 컴퓨팅 자원의 상태를 상황정보를 입력 또는 검색의 요청을 메타서버에게 한다. 메타서버는 프로파일과 정책정보 그리고 사용자의 선호도 정보를 미들웨어에게 돌려주는 역할을 한다.



<그림 5> 메타서버와 상황정보

3.1 메타데이터 스키마

본 연구의 메타데이터의 스키마는 모바일 환경에서

사용자가 자신의 디바이스를 사용해 모바일 호스트의 정보를 서비스 받는 상황을 전제로 설계 했으며, 다음과 같은 상황의 변경이 가능하다.

- 사용자 형상 정보의 추가 및 변경 : 인적 상황 변경
- 사용자 위치 정보의 변경
- 디바이스 정보의 추가 및 변경
- 디바이스 변경에 대한 서비스 정보의 지속적 연결
- 서비스 정보의 변경

변경 상황 중 마지막의 서비스 정보의 변경은 네트워크의 연결이 지속되지 않아도 되므로 본 연구에서 관심을 갖는 변경은 나머지 네 가지 상황이다. 한 번에 하나의 상황만 변경되는 것이 아니라 둘 이상의 경우가 한 번에 발생할 수 있다. 이런 상황에 적절히 대응하기 위해 형상과 상황 수준으로 구별해서 설계한다.

메타데이터는 형상수준(Configuration Level)과 상황수준(Context Level)으로 구분된다. 형상수준은 사용자나 사용자 인터페이스의 형상정보로 자주 변하지 않는 상대적으로 정적 데이터라 할 수 있고, 상황수준은 상황인지에 따라 입수된 상황변경 정보를 의미한다. 인지된 정보는 미들웨어의 전략 관리(Strategic Management)와 적응 관리(Adaptation Management)에 의해 이루어진다.

현재 메타데이터의 표준은 각 분야에서 활발히 진행 중이나 아직 표준이 설립되지는 못한 상태이다. 본 연구에서는 Dublin Core의 정의[14]에 근거하여 필요한 요소들을 선정하여 메타데이터를 설계한다.

사용자 형상 정보는 사용자를 구별할 수 있는 기본적인 정보로써 <표 1>과 같이 표현할 수 있다. 사용자의 ID, 패스워드, 위치정보, 사용자고유번호(주민번호), 소속기관, 부서, 전화번호, 주소 그리고 등록일이 포함된다.

<표 1> 사용자 형상 메타데이터

userID
password
location

SSnumber
occupation
division
phoneNumber
address
issueDate

사용자 디바이스 형상 정보는 <표 2>와 같이 표현한다. contributor는 디바이스에 접근할 수 있는 사용자를 의미하고, IP주소, 고정IP 여부, coverage는 사용권한의 수준, creator와 date는 디바이스 제작자 및 제조일, description은 제품설명, hardAddr은 MAC주소가 될 수 있으며, identifier는 제품고유번호, Language는 디바이스에서 사용하는 언어, resolution은 디바이스의 해상도 그리고 outputPower는 디바이스의 출력력을 의미한다.

<표 2> 사용자 디바이스 메타데이터

contributor
coverage
IPaddress
IPfixed
creator
date
description
hardAddr
identifier
language
publisher
resolution
outputPower

사용자 상황정보와 사용자 디바이스 상황정보는 <표 3>으로 구성된다. Context는 변경하려는 정보가 사용자의 상황인지 디바이스의 상황인지를 표시하고, Mode는 추가, 삭제 및 변경으로 구별된다. Factor는 형상정보에 포함된 메타데이터의 속성이 되며, 그에 대한 값을 Value에 준다. Mode가 삭제일 경우 이 값은 null을 갖는다.

<표 3> 상황정보 메타데이터

Context	Mode	Factor	Value
---------	------	--------	-------

디바이스 변경에 대한 서비스 정보의 지속적 연결

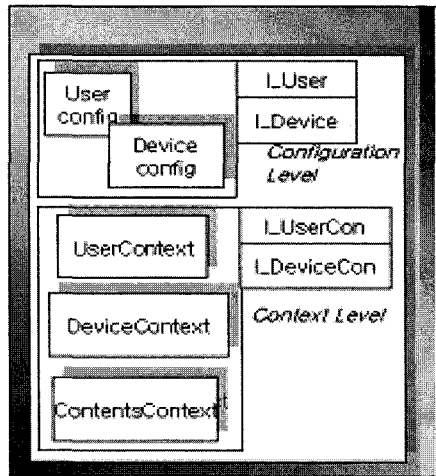
은 사용자가 다른 디바이스를 사용하는 경우 서비스가 지속적으로 이루어지는 것을 의미한다. 사용자 디바이스에서 다루는 콘텐츠의 종류는 텍스트, 이미지 그리고 동영상이다. 이에 따라 지속적 연결을 위해 유지해야하는 정보는 달라지는데 이 중 동영상의 경우 동영상 파일의 DTD에서 지속적 연결을 위한 정보를 <표 4>와 같이 추출한다.

<표 4> 지속적 연결을 위한 동영상 정보

totalFrames
currentFrame
currnetSector
totalTime

totalFrames는 동영상의 전체 프레임 수, currentFrame은 현재 프레임, currentSector는 현재 프레임이 포함된 섹터 그리고 totalTime은 전체 상영 시간을 의미한다.

3.2 메타서버 구조



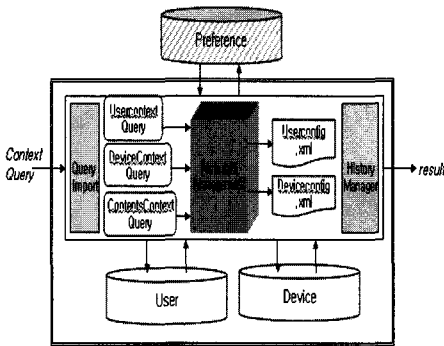
<그림 6> 메타서버의 데이터구조

<그림 6>은 메타서버의 관리구조를 나타낸다. UserConfig는 사용자의 형상정보를 가지며, 적응관리와 선호도 서비스를 위해 미들웨어에서 관리한다. DeviceConfig는 디바이스의 형상정보를 가지며, 디바

이스의 특징, 제조사 등의 속성과 네트워크 상에서의 물리적 위치를 표현한다. UserContext는 UserConfig의 속성 값이 바뀌거나, 사용자가 디바이스를 변경하는 경우 지속적인 서비스를 제공하기 위해 필요하다. DeviceContext는 DeviceConfig의 속성 값이 바뀌었을 때, 적응관리(Adaptation Management)의 정보가 된다. 예를 들어, 미들웨어에서 서비스를 제공받고 있는 어느 디바이스의 네트워크 대역폭이 현저하게 떨어졌을 때, 속성 값을 바꾸고 이에 대한 적응관리 서비스를 제공한다. ContentsContext는 디바이스의 변경이 있는 경우, 지속적으로 콘텐츠를 서비스하기 위한 상황정보를 갖는다.

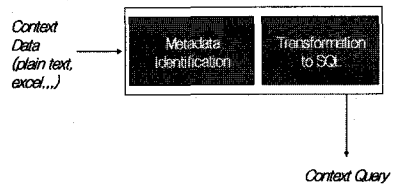
3.3 구현

본 연구에서 제안한 메타데이터 스키마의 구현 및 실험은 미들웨어 상에서 XML로 구현한다. XML은 웹 상에서 구조화된 문서를 제공하는 것이 가능하도록 설계된 표준화된 텍스트 형식으로써, 문서의 내용과 관련된 태그를 사용자가 직접 정의하여 사용한다. 체계적이고 일관적인 접근 방식을 정의하고 많은 기술을 포괄할 수 있는 기본 프레임워크를 제공한다. <그림 7>은 본 논문에서 제안한 메타 서버의 구현기능 요소를 보여주며, J2sdk1.4.2_05를 이용하였다. 데이터를 저장하기 위한 데이터베이스로는 JDBC를 이용 MySQL을 사용했다.



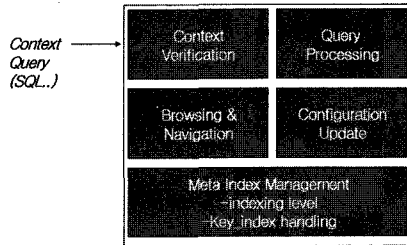
<그림 7> 메타서버의 구현 기능 요소

시스템의 구성요소를 살펴보면, 'Import Context'는 요구하는 상황정보를 받아서 이를 XML로 변환한다. 'Verify Consistency'는 일관성 검증을 통해 적절한 요구인지를 판단한다. 일관성은 예를 들어, 상황 변경 요구에 포함된 속성이 형상정보에 나타나지 않는 경우를 검사하는 기능을 포함한다. 'Manage History'는 상황정보의 내역을 관리하는 기능을 한다. 사용자, 디바이스 그리고 사용자 선호도 데이터베이스를 통해 최단시간 내에 결과를 도출한다.



<그림 8> QueryImport의 내부

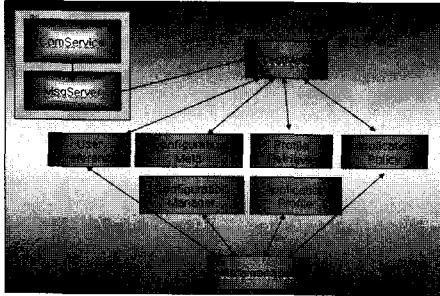
<그림 8>은 메타데이터 구현 아키텍처의 QueryImport의 내부이며, MetadataIdentification은 상황 정보에서 메타 데이터를 구별하고, TransformationToSQL은 구별된 메타데이터를 SQL로 변환한다.



<그림 9> MetaManager 내부

<그림 9>는 메타데이터 구현 아키텍처의 MetaManager에 포함되는 모듈을 보여준다. ContextVerification은 입수된 상황정보가 타당한 요구인지를 검증하는 기능을 하고, QueryProcessing은 상황검증을 마친 요구를 수행한다. Browsing&Navigation은 상황정보의 검색 기능을 수행하며, ConfigurationUpdate는 형상정보를 수정하는

기능이다. MetaIndexManagement는 메타데이터의 구조를 관리하는 기능으로, 메타데이터가 많을 경우, 인덱스(index)와 키 인덱스(key Index)를 활용하여 관리한다.



<그림 10> 메타서버 구현 클래스

<그림 10>은 메타 서버를 구축한 실제 클래스를 보여준다. 미들웨어와의 통신제어부분, 통신 수행부분 그리고 상황정보와 히스토리를 로컬 컴퓨터로 관리 위한 부분 등 세가지로 구성된다. ComService와 MsgServer는 통신제어 부분으로 TCP/IP를 통해 SOAP 메시지를 생성하고 송수신 하는 역할을 한다. TestMeta는 통신용 메인을 제어하는 클래스이고, Metamanager는 로컬관리를 위한 사용자 인터페이스를 제공한다.

<표5> 통신 시나리오의 예

Application_profile 검색 시나리오	Class	method
클라이언트 접속을 기다려 서비스 스레드를 할당한다.	MsgServer	doJob()
미들웨어에서 보낸 Message를 읽고 parseMsg를 호출한다.	ComService	Run()
메시지를 parsing하고 해당 클래스를 호출한다.	TestMeta	ParseMsg()
Application_profile 검색 결과를 xmlResult와 파일(profile.xml)로 생성한다.	Profile	Retrievefrom DB()
결과를 클라이언트로 전송한다.	MsgServer	sendMsg()
필요에 따라 클라이언트 제거 또는 통신을 중단한다.	MsgServer ComService	doJob() Run()

<표6> 관리 시나리오의 예

관리 시나리오	Class	method
Metamanager를 시작하고 [application profile] 메뉴를 선택한다.	Metamanager	Metamanager[]
Application profile manage 화면에서 원하는 기능을 수행한다.	Application Profile	
[User preference]메뉴를 선택하면 전체 preference를 보여준다.	User Preference	totalPreference[]
[Application _profile]선택하고 원하는 기능을 수행한다.	Application Profile	
[configuration]선택하고 원하는 기능을 수행한다.	Configuration Manager	
[Application Policy] 선택하면 전체 policy를 보여준다.	Application Policy	total Policy

<표 5>와 <표 6>은 미들웨어와의 통신 시나리오와 로컬 관리 시나리오의 예를 각각 보여준다.

4. 결 론

본 연구에서는 모바일 응용 서비스의 상황 인식을 위한 메타데이터 서버를 제안했다. 메타데이터는 형상 데이터와 상황데이터로 구분되며, 형상 데이터는 사용자와 사용자 디바이스의 형상을 표현한다. 상황데이터는 사용자 또는 디바이스의 상황정보를 표현하기 위한 구조를 가지며, 동영상 콘텐츠 서비스의 연결을 위한 메타정보를 포함한다. 메타데이터 구현을 위한 아키텍처는 상황정보를 받아서 이를 XML로 변환한다. 일관성 검증을 통해 적절한 요구인지를 판단 후, 사용자, 디바이스 그리고 사용자 선호도 데이터베이스를 통해 최단시간 내에 결과를 도출하는 개념이다.

이 방법은 기존 연구에서 갖는 확장성의 문제를 극복하기 위해 서버로 구현했고, 메타데이터의 유연성을 보장하는 반면 다양한 경우의 배리언트를 취급하는 오버헤드를 줄이기 위해 상대적으로 단순화했다. 또

한, 접근속도의 향상을 위해 인덱스 구조를 사용한다. 현재 일관성 검증은 요구한 상황정보가 형상정보에 포함되어있는지를 검사하는 수준이나, 향후, 데이터 관리측면에서의 일관성 검증을 위한 규칙을 정의하는 작업이 이루어질 것이다.

참고문헌

[1] A.K.Dey and G.Abowd, "Towards an understanding of context and context_awareness," Handheld and Ubiquitous Computing(HUC), 1999

[2] 김재호, 배정숙, 김성희, "차세대 이동통신망에서 상황인식 서비스," 전자통신 동향분석 제19권 제 3호, pp.32-40, 2004

[3] Jeffrey Hightower and Gaetano Borriello, "A Survey and Taxonomy of Location Systems for Ubiquitous Computing," Technical Report UW-CSE 01-08-03, University of Washington, Aug. 2001

[4] Schilit, B., Adams, N.Want, R. Context-Aware Computing Applications. Proceedings of 1st International Workshop on Mobile Computing Systems and Applications, 1994, pp.85-90

[5] Benjamin Bappu, "Reflective Framework for Ubiquitous Mobile Computing," Ph.D Dissertation of Lancaster University, 2002

[6] Dan Chalmers, Naranker Dulay, and Morris Sloman, "Meta Data to Support Context Aware Mobile Applications," proceedings of MDM'04, 2004, pp.199-211

[7] Gerald O'Connor and Mark Roantree, "A Mobile Schema Browser for Integration Specialists," ERCIM News, No. 54, 2003

[8] Jari Korva, Johan Plomp, Petri Määttä, and Maija Metso, "Adaptation for Mobile and Fixed Terminal Devices," Proceedings of the Second International Conference on Mobile Data Management, 2001, pp.252-259

[9] Johannes Peltola, Johan Plomp, and Tapio Seppänen, "A Dictionary-Adaptive Speech Driven User Interface for a Distributed Multimedia Platform," proceedings of EUROMICRO, 1999

[10] Licia Capra, Wolfgang Emmerich, and Cecilia Mascolo, "Exploiting Reflection and Metadata to build Mobile Computing Middleware," Proceedings of Mobile Computing Middleware, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 6, No. 4, 2002, pp. 34-44

[11] Paul Grace, Gordon S. Blair, and Sam Samuel, "ReMMoC: A Reflective Middleware to Support Mobile Client Interoperability," Proceedings of International Symposium on Distributed Objects and Applications(DOA), LNCS, 2003, pp.1170-1187

[12] Metso M, Löytynoja M, Korva J, Määttä P, and Sauvola J, "Mobile Multimedia Services - Content Adaptation," 3rd International Conference on Information, Communications and Signal Processing, 2001

[13] Context-aware Middleware for Resource Management in the wireless internet, P. Bellavista, A.Corradi, R.Montanari, C.Stefanelli, IEEE trans. on Software Engineering, 2004

[14] Stuart Weibel, John Kunze, Carl Lagoze, and Misha Wolf, "Dublin Core Metadata for Resource Discovery," The Internet Society, <http://www.ietf.org/rfc/rfc2413.txt>, 1998



이서정

1989년 숙명여자대학교 전산학과 졸업
(이학사)

1991년 동대학교 대학원 석사과정 졸업
(이학석사)

1998년 동대학교 대학원 박사과정 졸업
(이학박사)

1998년 ~ 2003년 8월 동덕여자대학교 강의교수

2004년 ~ 2004년 10월 숙명여자대학교

정보과학대학 연구원

관심분야: Component Based Development,
Product-Line Engineering
Adaptive and Awareness Network



황병연

1986년 서울대학교 컴퓨터공학과
(공학사)

1989년 한국과학기술원 전산학과
(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1994년 ~ 현재 가톨릭대학교 컴퓨터정보공학부 교수

1999년 ~ 2000년 University of Minnesota
Visiting Scholar

관심분야: 지리정보시스템, XML 데이터베이스, 데
이타 마이닝, 정보검색



윤용익

1985년 한국과학기술원 전산학과
(공학석사)

1994년 한국과학기술원 전산학과
(공학박사)

1997년 ~ 현재 숙명여자대학교 교수

1985년 ~ 1997년 9월 한국전자통신연구원
책임연구원

2004년 ~ 현재 University of Colorado at
Denver Visiting Professor

관심분야: 미들웨어, 분산 멀티미디어 시스템, 지리
정보 시스템, 임베디드 시스템