

SOA 기반의 U-City 서비스 통합 아키텍처

(SOA-based Integrated U-City Service Architecture)

이강표[†] 임영석[†] 안재민[†] 유진수^{**} 김형주^{***}
 (Kangpyo Lee) (Youngseok Lim) (Jaemin Ahn) (Jinsoo You) (Hyoung-Joo Kim)

요약 최근 각광 받고 있는 SOA(Service-Oriented Architecture)는 서비스 지향 아키텍처로서, 소프트웨어 개발 및 응용의 새로운 패러다임이라고 할 수 있다. 본 논문에서는 서비스 통합의 중요성이 강조되는 U-City 사업에서 SOA를 기반으로 다양한 서비스들을 효과적으로 관리, 제어할 수 있는 통합 아키텍처를 제시한다. SOA의 주요 특징이라고 할 수 있는 느슨한 결합, 표준 기반, 그리고 분산 컴퓨팅은 U-City 사업의 다양한 서비스를 통합하여 제공하기 위한 필수적인 요소들이다. 본 연구에서는 SOA의 이러한 특징들을 반영하기 위해 통합 플랫폼인 ESB(Enterprise Service Bus)를 이용하였는데, ESB는 상호 이질적인 컴포넌트와 시스템을 연결함으로써 서비스들 간의 통신을 구현하는 SOA의 핵심 모듈이라고 할 수 있다. 본 논문에서는 U-City 서비스에서 SOA의 필요성과 이를 실제로 구현할 수 있는 가상 시나리오와 구현방법에 대해 논한다. 최종적으로 U-City 통합운영센터를 위한 SOA 기반 통합 아키텍처를 제안한다.

키워드 : SOA, 서비스, 아키텍처, U-City, ESB

Abstract SOA (Service-Oriented Architecture), which has become very popular recently, is a new paradigm for software development and application. In this paper, we propose an integrated architecture which is able to effectively manage and control a variety of services for U-City projects focusing on the importance of service integration. SOA has a number of important features such as loose coupling, standard bases, and distributed computing, all of which are the essential elements for merging and providing various services in U-City projects. We exploit the ESB (Enterprise Service Bus) for reflecting those features, which is a core module linking mutually heterogeneous components so that the communication of services can be implemented. In this paper, we discuss the necessity of SOA in U-City services and a possible scenario and method for the implementation. Finally, we propose an integrated architecture for the U-City Integration and Management Center.

Key words : SOA, Services, Architecture, U-City, ESB

· 본 연구는 국토해양부 첨단도시개발사업의 연구비지원(07첨단도시 A01, BK-21 정보기술 사업단, 지식 경제부 및 정보통신연구진흥원의 대학 IT연구센터 육성지원사업(NIPA-2009-C1090-0902-0031)의 연구 결과로 수행되었음

† 비회원 : 서울대학교 컴퓨터공학부
 kplee@idb.snu.ac.kr
 yslim@idb.snu.ac.kr
 jmahn@idb.snu.ac.kr

****** 비회원 : 연세대학교 공간정보서비스연구소 교수
 goodtree@yonsei.ac.kr

******* 종신회원 : 서울대학교 컴퓨터공학부 교수
 hjk@snu.ac.kr

논문접수 : 2009년 11월 2일
 심사완료 : 2009년 12월 30일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 레터 제16권 제3호(2010.3)

1. 서론

SOA, 즉 서비스 지향 아키텍처(Service-Oriented Architecture)[1-5]의 태동은 비즈니스 분야의 필요에 의하여 시작되었다. SOA는 IT 분야가 빠르게 변화하는 비즈니스 환경에 효율적으로 적응하기 위한 방법론으로서, 1996년 가트너 그룹[6]에서 최초로 제안되었다. SOA는 개념적인 수준의 모델로 출발하였으나, 지금은 웹 서비스(Web Services)의 등장과 함께 WSDL(Web Service Definition Language)이라는 XML(eXtensible Markup Language)에 기반한 명세를 이용하면서 그 실체를 설명할 수 있게 되었고, 구현 가능한 수준의 하나의 아키텍처로서 확고히 자리잡고 있는 추세이다.

서비스(service)는 서비스 지향 아키텍처를 구성하는 핵심적인 개념이다. 여기서 서비스란 외부에서 표준화된

절차에 의하여 호출하는 단위로서, 재사용이 가능하면서 플랫폼에 독립적인 인터페이스를 갖고 있는 하나의 독립적인 컴포넌트(component)라고 볼 수 있다. 또한 동적으로 검색될 수 있으며, 자신의 상태를 스스로 유지하는 성질을 갖고 있다. 서비스끼리는 느슨한 결합(loose coupling)으로 연결되어 있고, 각각의 서비스는 각자 의미 있는 단위의 기능을 수행한다. 사실 서비스를 바탕으로 한 SOA의 이러한 기본 개념들은 SOA에서 최초로 제안된 것은 아니다. DCOM(Distributed Component Object Model)이나 CORBA(Common Object Request Broker)와 같은 고전적인 분산 아키텍처 기술에서도 이와 유사한 개념들이 이미 제안되었다. 그러나 이러한 과거의 시도들은 이를 뒷받침해주어야 할 기술과 표준화 작업의 부재로 현재의 SOA와 같은 혁명을 이루지는 못하였다.

U-City 서비스에는 다양한 서비스 표준화를 통해 제안되는 서비스들뿐만 아니라, 사용자의 새로운 요구에 맞추어 추가되는 서비스들이 끊임없이 등장한다. 하지만 현재의 서비스들과 새로운 서비스들은 상당 부분 중복된 기능을 포함하고 있다. 예를 들면 U-Device로부터의 정보를 받아들이거나 받아들이는 정보를 서비스 인터페이스로 보내기 위한 컴포넌트, 서비스 보안, 관리, 인증 컴포넌트들이 이에 해당한다. 따라서 이렇게 중복되는 기능들은 하나의 컴포넌트를 이용해서 다양한 여러 서비스들에게 ESB(Enterprise Service Bus)라는 채널을 통해 제공될 필요가 있다. 이를 만족시키는 플랫폼을 구성하기 위해서는 개별 서비스 단위의 컴포넌트들을 이용해서 전체 시스템을 설계할 수 있는 SOA 아키텍처가 필요하다.

그 동안의 크고 작은 U-City 사업들의 경우, SOA 기반의 통합 플랫폼을 적용하여 구축하지 않았거나 SOA 기반의 아키텍처에 대한 이해가 충분히 이루어지지 못한 관계로 SOA의 장점을 충분히 활용하지 못 해온 것이 사실이다. SOA의 재사용성과 유연성을 고려해보았을 때 U-City 서비스 통합 아키텍처 구축에 있어서 SOA는 U-City 사업의 여러 가지 요구사항들을 충족시켜줄 수 있는 최적의 아키텍처라 할 수 있다. 이에 본 연구에서는 2008년 11월부터 2009년 6월까지 참여한 국토해양부 주관 U-Eco City 사업[7] 내의 U-City 통합 운영센터 제공 서비스 표준체계 연구개발 경험을 바탕으로 SOA 기반의 U-City 서비스 통합 아키텍처를 제안한다. 참고로 본 연구에서 제안하는 통합운영센터의 SOA 기반 통합 아키텍처는 연세대학교의 주관 하에 KT, LG CNS, SK C&C, 대우정보시스템, 티맥스소프트 이상 5개 참여업체의 공동작업으로 산출된 결과물임을 밝힌다.

2. 관련연구

SOA에 대한 연구는 1996년의 첫 제안 이후 현재까지 국내외의 많은 학계 및 업체들에 의해 활발히 진행되어왔다. [8]에서는 SOA에 대한 접근방법들과 제안된 기술들, 그리고 연구 주제들에 대해 전반적이고 개괄적인 소개를 하고 있다. SOA에 대한 최근 연구에서는 SOA의 구현에 필수적인 웹 서비스를 이용하여 기존에 존재하는 다양한 분야의 어플리케이션에서 SOA가 응용될 수 있는 가능성을 제시하고 있다. [9]에서는 상호 이질적인 플랫폼이나 프로토콜에 디바이스 서비스를 유연하게 제공하기 위하여 서비스 변환 게이트웨이 엔진(Service Convergence Gateway Engine)과 웹 서비스, 소프트웨어 서비스와 어플리케이션 컴포넌트들 간의 연동을 위해 ESB를 사용하여 서비스를 통합하고 관리하는 구조를 제안하고 있다. [10]에서는 유비쿼터스 서비스 조합 시나리오에서 ESB를 활용해서 두 개 이상의 서비스 컴포넌트들 사이의 데이터 흐름을 효율적으로 제어하기 위한 시나리오 디자인 지원 시스템을 제안하고 있다.

한편, 우리나라에서도 U-City에 대한 관심과 필요성이 증가하고 여러 정책들이 현실화되면서, U-City 구축에 대한 연구도 활발히 진행되었다. [12]에서 U-City는 유비쿼터스 정보기술에 기반을 둔 차세대 지능화된 도시의 새로운 패러다임을 의미하며, 언제 어디서나 원하는 정보와 기능을 얻을 수 있는 친환경, 첨단, 자급자족형, 지속가능한 구조의 도시개념으로 정의하고 있다. [14]에서는 U-City 에서 필요한 서비스들의 일관된 표준을 위해 U-City 서비스 표준화의 개념을 '합리적인 U-City 서비스 표준을 정의하고 활용하기 위해 서비스 표준을 발굴, 개발, 운용 및 관리 등의 전반적인 활동'으로 정의한다. [11]에서는 이와 같은 U-City에서 필요한 U-Home, U-Work, U-Learning, U-Health, U-Transport, U-Public 등 U-City 주요 분야별로 구현 가능한 U-City 응용서비스 모델을 제시하고 있으며 [13]에서는 도시공간별 도시활동에 따라 서비스를 세분화, 모듈화하여 제시하고 있다.

3. SOA 기반의 U-City 서비스 통합 아키텍처 제안

3.1 서비스의 정의

U-City에서의 서비스는 '유비쿼터스 도시의 건설 등에 관한 법률'[15]에 따르면 유비쿼터스 도시 기반 시설 등을 통하여 행정, 교통, 복지, 환경, 방재 등 도시의 주요 기능별 정보를 수집한 후, 그 정보 또는 이를 서로 연계하여 제공하는 서비스로서 대통령령으로 정하는 개

넘적인 수준의 서비스를 의미한다.

3.2 SOA 기반 통합 아키텍처

전체 아키텍처는 크게 Device Interface, Core Service Module, Process Management Module, Enterprise Service Bus, Integrated Database, Service Interface의 6개 컴포넌트들로 구성되어 있다. 각 컴포넌트의 역할과 특징에 대해서 알아본다.

3.2.1 Device Interface

그림 1과 같이 Device Interface는 USN, RFID, CCTV 등의 U-Device들로부터 정보를 수집하고 U-Device의 보안, 인증, 제어를 담당하게 된다. 또한 ESB에서 요구하는 형식에 맞게 전처리 과정을 수행한다. 전처리 과정을 거친 정보는 ESB로 보내진다.

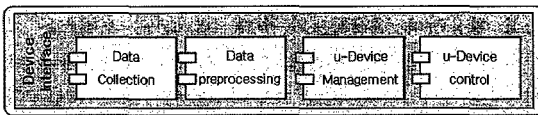


그림 1 Device Interface

3.2.2 Enterprise Service Bus(ESB)

본 아키텍처에서의 ESB는 그림 2와 같이 Information Routing, Message Transformation, Protocol Transformation의 세 가지 역할을 수행하게 된다. 첫 번째 Information Routing이란 메시지의 내용에 기반한 Routing을 지원하고 메시지를 중개함을 의미한다. 두 번째 Message Transformation은 XML 기반의 동기/비동기 SOAP 메시징을 가능하게 한다. SOAP 표준에 기반한 메시지를 교환함으로써 이기종이나 분산 환경에서의 데이터 및 애플리케이션 연동이 가능하도록 지원한다. 마지막 Protocol Transformation은 J2EE JCA를 준수하는 모든 상용 어댑터(adapters)를 지원하게 된다. SOA의 가치를 충분히 보장할 수 있도록 시스템을 구축하기 위해서는 ESB를 중심으로 한 아키텍처 구성이 필수적이며 그에 따라 본 연구에서도 이를 반영하였다.

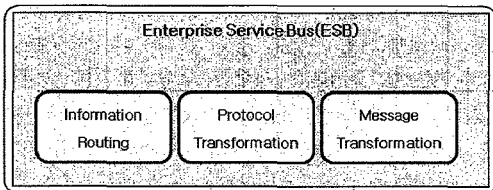


그림 2 Enterprise Service Bus(ESB)

3.2.3 WSDL

WSDL은 Microsoft 사의 SDL(Service Description Language)과 SCL(SOAP Contract Language), IBM

의 NASSL(Network Accessible Services Specification Language)을 결합하여 개발된 것으로 SOAP과 마찬가지로 W3C의 공개표준으로 제출되어 있다. ESB에서 지원하는 WSDL은 웹서비스의 구체적 내용이 기술되어 있는데, 서비스 제공 장소, 서비스 메시지 포맷, 프로토콜 등이 XML 형태로 기술되어 있어 웹 서비스를 제공하는 쪽에서 많은 서비스 이용자에게 손쉽게 웹 서비스에 관한 내용을 배포할 수 있고, 표준 인터페이스이므로 기술된 내용에 대한 이해가 쉬운 것이 장점이다.

3.2.4 통합 데이터베이스(Integrated Database)

그림 3과 같이 U-City에서 사용되는 다양한 서비스들은 분산된 형태의 정보관리가 아닌 하나의 통합 데이터베이스에서 정보가 관리된다. 다양한 종류의 디바이스에서 입력되는 정보뿐만 아니라 Process Management Module에서 사용되는 워크플로우(workflow) 정보, U-City 시스템의 운영관리에 관한 정보 등이 통합 데이터베이스에서 관리된다.

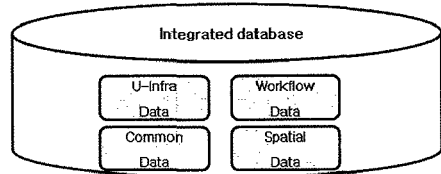


그림 3 Integrated Database

3.2.5 Core Service Module

그림 4와 같이 Core Service Module에서는 U-City에서 사용되는 다양한 U-서비스들에서 공통적으로 필요한 핵심 서비스들을 포함하고 있다. 핵심 서비스는 메시지, 보안/인증, 운영관리/Admin, 서비스 지원, 공간정보 서비스의 크게 5가지의 형태로 분류할 수 있다.

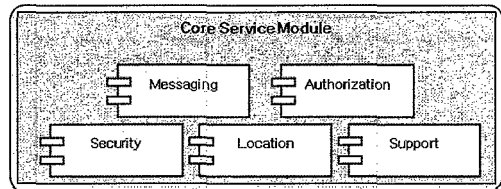


그림 4 Core Service Module

3.2.6 Process Management Module

그림 5와 같이 Process Management Module에서는 각각의 U-City 서비스들의 자료 입력부터 실제 서비스 되기까지의 전체과정에 대한 워크플로우의 정의, 관리, 프로세스 실행, 상황인식, 상황처리 등의 기능을 수행한다.

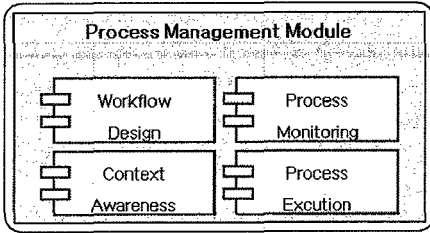


그림 5 Process Management Module

3.2.7 Service Interface

그림 6과 같이 Service Interface Module은 ESB를 통해 전달 받은 정보를 U-서비스를 이용하는 사용자들에게 다양한 매체를 통해 제공하는 역할을 수행한다. 앞서 밝힌 바와 같이 ESB를 통해 통신하므로 일반적인 애플리케이션뿐만 아니라 레거시 시스템에서도 쉽게 정보전달이 가능하다.

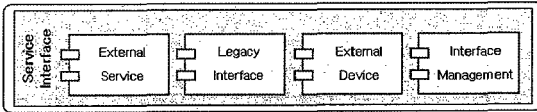


그림 6 Service Interface

3.2.8 전체 플랫폼 아키텍처

U-City 통합 플랫폼의 서비스를 효율적으로 제공하기 위해 이상에서 살펴본 컴포넌트들을 조합한 전체 통합 아키텍처는 아래 그림 7과 같다. 결론적으로 SOA 기반 U-City 통합 아키텍처는 중복되는 컴포넌트들의 사용 없이 ESB를 중심으로 다양한 프로토콜의 데이터를 WSDL를 통하여 서비스 제공자와 서비스 이용자 간의 원활한 통신이 이루어지는 구조이다.

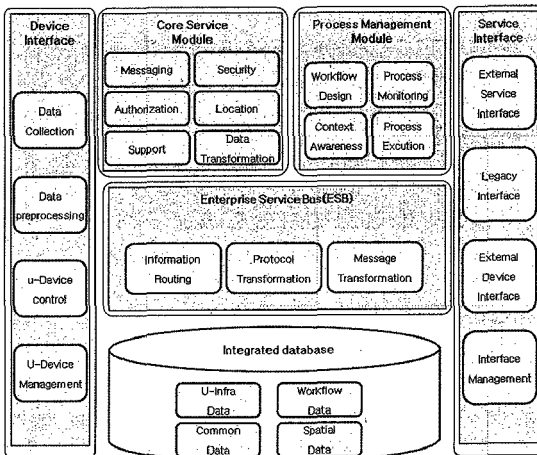


그림 7 SOA 기반 통합 아키텍처

4. 구현 예시

4.1 구현 환경

구현에 사용된 ESB는 처음 Codehaus[16]에서 시작되어 현재 Apache Project에서 진행되는 오픈 소스 기반 ESB 프로젝트인 ServiceMix[17] 버전 3.3.1을 사용하였다. 이 ServiceMix, 레거시 시스템과 통합 시스템 모두 Java 환경에서 동작할 수 있도록 JDK 1.6 버전을 사용하였다.

4.2 구현 시나리오

U-City의 다양한 서비스들의 유연한 개발과 적용을 보이기 위해 다음과 같은 시나리오를 구현하였다. 기존 레거시 시스템으로서 교통관리 서비스와 병원관리 서비스가 존재한다고 가정할 때, 통합운영센터의 인명피해에 대한 재난관리 서비스를 이 2개의 레거시 시스템 서비스를 활용하여 구성하고자 한다. 재난관리 서비스는 재난통제 시스템에서 수집한 도시 전체의 재난상황에 대한 정보 중 인명에 대해 재난이 발생할 경우 재난상황 내용을 병원관리 서비스로 전송하게 되고, 병원관리 서비스에서는 해당 사고 위치에서 가장 가까운 병원을 찾아서 재난관리 서비스에 알려준다. 이 정보를 받은 재난관리 서비스에서는 교통관리 서비스에 현재 사고 발생 위치와 해당 병원정보를 전송하게 된다. 이 정보를 받은 교통관리 서비스에서는 사고 발생위치와 해당 병원까지의 가장 빠른 경로를 탐색하여 재난관리 서비스로 전송하게 된다.

4.3 구현

ServiceMix를 이용하여 다음과 같이 서로 연동되는 3개의 서비스를 구현하였다. 교통 관리 서비스(Traffic)와 병원 관리 서비스(Emergency)는 기존에 존재하고 있다고 가정한 레거시 시스템이다. 재난관리 서비스(Disaster)는 기존의 레거시 시스템을 사용하여 새롭게 추가하고자 하는 통합 시스템이다.

ServiceMix는 다양한 프로토콜의 메시지들을 라우팅 해주는데, 본 논문에서는 파일을 통해 서로 메시지를 주고받도록 구현하였다. 이를 위해 ServiceMix의 file-binding component를 사용하였으며, 이는 어떤 파일이 inbox 디렉토리에 저장될 경우 분석하여 outputbox로 필요한 정보로 옮겨주는 역할을 수행한다. 본 구현에서는 입력되는 파일의 <disaster> 태그의 filename 속성 값을 이름으로 가지는 xml 파일로 outputbox에 저장되도록 file-binding component를 구성하였다. 3개의 시스템 모두 Java로 구현하였으며 독립적인 서비스를 나타내기 위해 서로 다른 Package로 구성하였다. 실제 적용 시 각 시스템은 Java가 아닌 다른 프로그래밍 언어를 사용해도 무방하며, 본 연구에서는 편의를 위해 Java로만 구현하였다.

4.3.1 교통관리 서비스(Traffic)

traffic package로 정의하고 교통현황 수집, 빠른 길 검색, 교통흐름 제어의 3개 컴포넌트들로 구성된다. 실제 통합을 위해서는 빠른 길 검색만이 사용되므로 다른 2개는 dummy class로 작성하고 빠른 길 검색을 traffic class로 정의하였다. 빠른 길 검색 컴포넌트의 동작은 다음과 같다. ESB의 outputbox에서 traffic_input.xml이라는 파일을 발견하게 되면 파일내용 중 사고 발생 장소, 즉 <accident_place> 태그값과 <hospital> 태그값을 취하여 이들 간의 가장 빠른 경로를 탐색하여 traffic_output.xml이라는 파일로 저장하여 재난관리서비스로의 전송을 위해 ESB의 inbox에 저장한다.

4.3.2 병원관리 서비스(Emergency)

emergency package로 정의하고 인접병원 검색, 응급 환자 분석, 응급환자 입원수속의 3개 컴포넌트들로 구성된다. 실제 통합을 위해서는 인접병원 검색만이 사용되므로 다른 2개는 dummy class로 작성하고 인접병원 검색을 emergency class로 정의하였다. 인접병원 검색 컴포넌트의 동작은 다음과 같다. ESB의 outputbox에서 emergency_input.xml이라는 파일을 발견하게 되면 파일내용 중 사고 발생 장소, 즉 <accidental_place> 태그값을 취하여 그 장소와 가장 가까운 병원을 탐색하여 그 결과를 emergency_output.xml이라는 파일로 저장하여 재난관리 서비스로의 전송을 위해 ESB의 inbox에 저장한다.

4.3.3 재난관리 서비스(Disaster)

disaster package로 정의하고 재난관리 서비스를 위한 하나의 컴포넌트로 구성된다. 이를 disaster class로 정의하였다. 기본적으로 인명피해사고의 재난상황에 대한 정보가 입력된다고 가정하고 크게 3개의 분기로 구성하였다. 재난정보가 입력되었을 경우, 병원관리 서비스로부터 입력이 들어왔을 경우, 교통관리 서비스로부터 입력이 들어왔을 경우의 세 가지로 나누어 구현하였다. 재난정보가 disaster_input.txt라는 파일로 입력되었을 경우는 해당 사고가 인명피해 사고일 경우 ESB의 inbox에 해당 사고 내용을 emergency_input.xml로 저장하여 병원관리 서비스로 전송한다. 병원관리 서비스로부터 응답을 받았을 경우 즉, outputbox에서 emergency_output.xml 파일을 발견할 경우 해당 정보를 교통관리 시스템으로 전송하기 위해 traffic_input.xml이라는 파일에 정보를 저장하여 ESB의 inbox에 저장한다. 마지막으로 교통관리 서비스로부터의 응답을 받았을 경우 즉, traffic_output.xml 파일을 발견할 경우 결과를 disaster_output.txt 파일로 저장하게 된다.

U-City의 다양한 U-서비스들, 그리고 향후 추가되는 서비스들은 대부분이 인증된 U-Device들로부터 정보를 받아들여 모든 정보가 통합 데이터베이스에 저장되고, Process Management Module의 워크플로우에 따라 ESB와 다양한 인터페이스를 통해 통신하게 된다. 특히 WSDL의 사용이 가능한 ESB는 서비스들의 중복되는 기능을 하나의 공통 모듈을 사용할 수 있도록 구성하고 새롭게 추가되는 서비스들의 유연한 개발과 운영이 가능하도록 해주는 핵심 구성요소이다. 따라서 ESB를 사용한 SOA 기반의 통합 플랫폼은 U-City의 대규모 통합 시스템을 위해 개발과 운영, 유지보수 등의 시스템 전체적인 측면에서 필수적인 요소라고 할 수 있다.

참 고 문 헌

- [1] T. Erl, "Service-oriented architecture: concepts, technology, and design," Prentice Hall PTR Upper Saddle River, 2005.
- [2] T. Erl, "Service-oriented architecture: a field guide to integrating XML and web services," Prentice Hall PTR Upper Saddle River, 2004.
- [3] C. Baroudi, J. Hurwitz, and R. Bloor, "Service-Oriented Architecture for Dummies," Wiley, 2006.
- [4] 장세영, 황상철, 이현정, 조문욱, "SOA 서비스 지향 아키텍처 개념에서 설계 구현까지", 에이콘, 2006.
- [5] 전병선, "SOA, What & How: A Road to SOA," 와우북스, 2008.
- [6] <http://www.gartner.com/>
- [7] 국토해양부 첨단도시개발사업 (07첨단도시 A01).
- [8] Mike P. Papazoglou and Willem-Jan van den Heuvel, "Service Oriented Architectures: Approaches, Technologies and Research Issues," 2005.
- [9] 정덕원, 민덕기, "유비쿼터스 환경에서의 서비스 통합을 위한 SCG엔진과 ESB 적용에 관한 연구", 2007.
- [10] Takaaki Moriya, "A Support System for Designing Ubiquitous Service Composition Scenarios," 2007.
- [11] 윤심, 이계원, "U-City 구축을 위한 응용서비스 모델 개발", 2005.
- [12] 이병철, 이용주, "u-City 사업모델과 u-서비스", 2007.
- [13] ㈜엘릭스, "u-City 응용서비스 모델 연구", 2005.
- [14] 김구수, 이범교, 장성봉, "국내 u-City 서비스 및 요소 기술 표준화", 2008.
- [15] 유비쿼터스 도시의 건설 등에 관한 법률, (타)일부개정 2009.06.09 법률 제9770호 시행일 2010.7.1.
- [16] Codehaus, <http://codehaus.org/>
- [17] ServiceMix, <http://servicemix.apache.org/>.

5. 결 론



이 강 표

2004년 연세대학교 컴퓨터과학과(학사)
2006년 서울대학교 컴퓨터공학부(석사)
2007년~현재 서울대학교 컴퓨터공학부
박사과정 재학중. 관심분야는 데이터베이
스, 시맨틱웹, 웹2.0, 시스템통합



임 영 석

2007년 KAIST 전산학과(학사). 2007
년~현재 서울대학교 컴퓨터공학부 석사
과정 재학중. 관심분야는 SNS, 데이터베
이스, 웹2.0



안 재 민

2008년 연세대학교 컴퓨터과학과(학사)
2008년~현재 서울대학교 컴퓨터공학부
석사과정 재학중. 관심분야는 웹2.0, 스팸
필터링, 데이터베이스



유 진 수

1991년 한양대학교 도시공학과(학사). 1996
년 Univ. of Illinois at Urbana-Champaign
(GIS 석사). 2000년 Univ. of Illinois at
Urbana-Champaign(GIS 박사). 2005
년~2007년 영국 Univ. of Nottingham
(Research Fellow). 2008년~현재 연세
대학교, U-City 융합서비스 연구개발단(연구교수). 2009년~
현재 OGC Korea(Chair). 관심분야는 GeoWeb, U-City
Service 표준화, Geosemantic Interoperability



김 형 주

1982년 서울대학교 전산학과(학사). 1985
년 Univ. of Texas at Austin(석사). 1988
년 Univ. of Texas at Austin(박사). 1988
년~1988년 Univ. of Texas at Austin
(Post-Doc). 1988~1990년 Georgia Ins-
titute of Technology(부교수). 1991년~
현재 서울대학교 컴퓨터공학부 교수. 관심분야는 데이터베
이스, XML, 시맨틱웹, 웹 2.0