

# 유비쿼터스 환경을 위한 웹 서비스 기술\*

신동훈\* · 이경호\*\*

## 1. 서 론

최근 들어 정보기술의 패러다임이 XML 및 웹 서비스에 기반한 서비스 지향 컴퓨팅(Service-Oriented Computing) 환경으로 변화하고 있다. 또한 무선 네트워크 및 모바일 컴퓨팅 기술의 발전에 힘입어 언제 어디서든 디바이스에 상관없이 원하는 서비스에 접근할 수 있는 유비쿼터스 컴퓨팅 환경이 구축되고 있다. 유비쿼터스 컴퓨팅 환경은 사용자가 컴퓨터나 네트워크를 의식하지 않고 장소에 상관없이 자유롭게 네트워크에 접속하여 컴퓨팅 자원을 활용할 수 있는 환경을 말한다. 1988년 마크 와이저는 '유비쿼터스 컴퓨팅'이라는 용어를 처음 사용하면서 유비쿼터스 컴퓨팅이 메인프레임과 PC에 이어 제3의 정보혁명을 이끌 것이라고 주장하였다[1]. 현재 유비쿼터스 환경의 구축 및 유비쿼터스 환경에서의 서비스 제공과 관련하여 전 세계적으로 많은 연구와 개발[2-5]이 활발히 진행되고 있다.

유비쿼터스 컴퓨팅 환경에서는 그림 1과 같이 기존에 디바이스로 인식되지 않았던 다양한 형태의 디바이스와 다수의 애플리케이션들이 네트워크를 통해 유기적으로 협업하여 사용자에게 다양

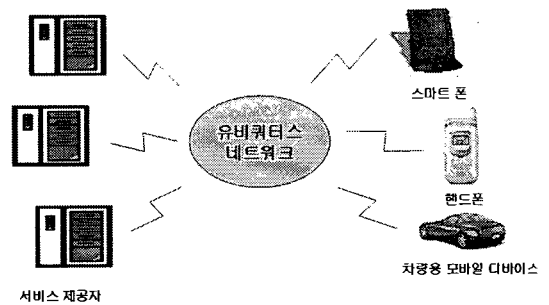


그림 1. 유비쿼터스 컴퓨팅 환경의 개념도

한 서비스를 제공한다. 이러한 이질적이고 자율적인 환경에서 디바이스 혹은 애플리케이션간의 효율적인 상호작용을 위해서는 상호운용을 위한 표준이 반드시 필요하다[6]. 이를 위해서 XML에 기반한 웹 서비스(Web services) 기술이 주목받고 있다. 웹 서비스는 웹을 통하여 호출이 가능하고 플랫폼에 독립적이기 때문에 이질적이며 개방된 환경에서 표준 인터페이스로서의 역할을 효과적으로 수행할 것으로 기대된다.

웹 서비스는 웹 상에서 호출하여 사용할 수 있는 소프트웨어 컴포넌트로서 XML, WSDL[7], SOAP[8], 그리고 UDDI[9] 등의 표준을 통하여 서비스의 출판, 검색 및 호출을 가능하게 하는 기술이다. 일반적으로 그림 2와 같이 서비스 제공자는 WSDL을 사용하여 제공하고자 하는 서비스를 기술하고, 이를 UDDI 레지스트리에 등록한다. 서비스 요청자는 레지스트리에서 원하는 서비스를

\* 연세대학교 컴퓨터과학과 박사과정  
 \*\* 연세대학교 컴퓨터과학과 조교수  
 ※ 이 연구는 정보통신부(정보통신연구진흥원)에서 지원하는 2005년도 IT기초기술연구지원사업의 연구결과임.

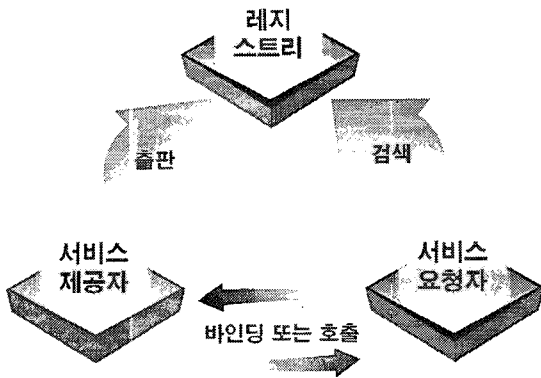


그림 2. 웹 서비스 개념도

검색한 후 SOAP을 통해 해당 서비스를 호출한다.

웹 서비스는 현재 웹처럼 이질적이며 자율적인 환경에서 개발 및 운영되는 소프트웨어의 상호운용 표준으로서 B2B 분야의 시스템 통합을 위해 많이 사용되고 있다. 그러나 데스크 탑 및 유선 환경만을 고려하여 개발되었기 때문에 기본적으로 무선 모바일 환경을 지원하는 유비쿼터스 환경에 그대로 적용하기는 어렵다. 무선 환경은 기존 유선 환경과 달리 네트워크 연결성이 보장되지 않고 상대적으로 작은 대역폭을 갖으며, 모바일 디바이스 역시 데스크탑과 비교하여 제한적인 리소스를 가진다. 또한 언제 어디서나 접근 가능한 유비쿼터스 네트워크의 장점을 살리기 위해서는 서비스 제공시에 사용자와 디바이스의 위치나 상태 혹은 선호도와 같은 컨텍스트 정보가 고려되어야 한다.

특히 유비쿼터스 환경에서는 사용자가 원하는 서비스를 정확히 찾기 위해서 시맨틱 웹 서비스 탐색 기술이 필요하다. 또한 고수준의 서비스 제공을 위해 복합 서비스를 자동으로 구성하는 기술이 필요하며, 컨텍스트의 변화에 따라 디바이스간에 서비스를 이동시킬 수 있어야 한다. 더불어 제한된 화면을 갖는 모바일 디바이스를 위해 멀티모달(multi-modal) 인터페이스를 지원하여야 한다.

본 논문에서는 이와같이 유비쿼터스 컴퓨팅 환경을 위한 웹 서비스 기술에 대해 논의하고자 한다. 기존의 웹 서비스 기술과 이들의 한계점을 극복하기 위한 연구들을 간략히 기술하고 유비쿼터스 컴퓨팅 환경에서 새롭게 고려되어야 할 이슈와 이와 관련한 연구를 소개한다. 또한 유비쿼터스 컴퓨팅 환경에서 사용자들에게 편리하고 질 높은 서비스를 제공하기 위하여 해결되어야 할 향후 과제에 대해 논의한다.

## 2. 유비쿼터스 웹 서비스 프레임워크

유비쿼터스 환경에서 사용자는 언제, 어디서든, 디바이스에 상관없이 네트워크에 접속하여 원하는 서비스를 이용할 것이다. 또한 사용자가 의식하지 못하는 순간에도 사용자의 디바이스는 사용자의 선호도, 작업패턴 혹은 주위 상황을 고려하여 필요한 서비스를 자동으로 제공한다. 특히 고수준의 복합 서비스를 제공하기 위해서 네트워크에 연결되어 있는 다양한 디바이스 혹은 서비스간의 자율적인 협업이 이루어지게 된다. 이러한 유비쿼터스 환경은 무선 모바일 컴퓨팅 기술 및 센서 네트워크 등의 기술 발달에 힘입어 점점 현실로 다가오고 있다.

그림 3은 유비쿼터스 환경에서 가능한 응용 시나리오의 예이다. 회사원 A씨는 출근길 지하철 안에서 스마트 폰을 사용하여 휴가기간 동안 가족들과의 여행을 위한 항공권과 숙박 시설을 예약하였다. 여행 당일 버스 정류장에 도착하니 공항행 버스의 번호와 버스 도착시간에 대한 정보가 스마트 폰을 통해 자동으로 제공되었다. 또한 공항에 도착하니 항공권의 발권 안내, 발권 위치, 출국 수속 안내 등 항공기 탑승에 필요한 정보들이 스마트 폰을 통해 자동으로 제공되었다.

위 시나리오를 위해서 여행 패키지 및 여행지

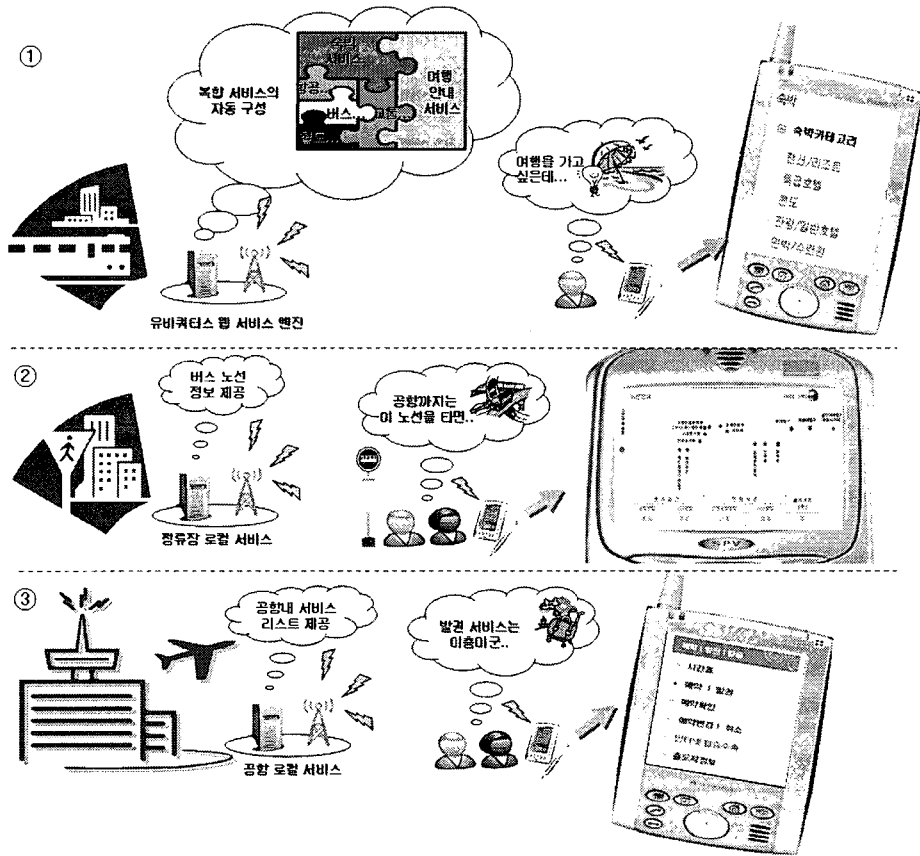


그림 3. 유비쿼터스 환경에서의 응용 시나리오의 예

에 대한 정보를 제공하고 호텔과 항공권의 예약을 지원하는 다수의 서비스들이 자동으로 조합되어 여행안내서비스라는 하나의 인터페이스를 통해 제공되었다. 또한 스마트 폰은 버스 정류장과 공항에 설치된 액세스 포인트 및 서비스 레지스트리와 상호작용을 통해 필요한 서비스를 검색하고 사용자에게 유용한 정보 및 서비스를 제공하였다.

이와 같이 다양한 디바이스와 애플리케이션들이 네트워크를 통해 연결되고 상호작용하게 될 유비쿼터스 환경에서 웹 서비스는 디바이스 및 애플리케이션 간의 훌륭한 인터페이스 역할을 할 것으로 기대된다. 하지만 위 시나리오에서와 같은 서비스를 제공하기 위해서는 기존의 웹 서비스와

다른 형태의 새로운 프레임워크 및 기술들이 요구된다. 그림 4는 이를 위한 유비쿼터스 웹 서비스 프레임워크를 보여준다.

기존의 웹 서비스 프레임워크와 달리 유비쿼터스 웹 서비스 프레임워크에서는 기본적으로 컨텍스트 정보의 수집과 처리, 서비스의 이동 및 재구성, 그리고 멀티모달 인터페이스를 지원하기 위해 개별 디바이스와 서버에 위치하는 미들웨어가 필요하다. 디바이스 측 미들웨어는 주로 컨텍스트 정보를 수집 및 관리하거나 사용자와 디바이스의 프로파일 및 IP 주소 등을 유지 및 갱신한다. 또한 다른 디바이스나 애플리케이션의 요청이 있을 경우 정보를 전송 및 공유하는 역할을 한다. 서버 측 미들

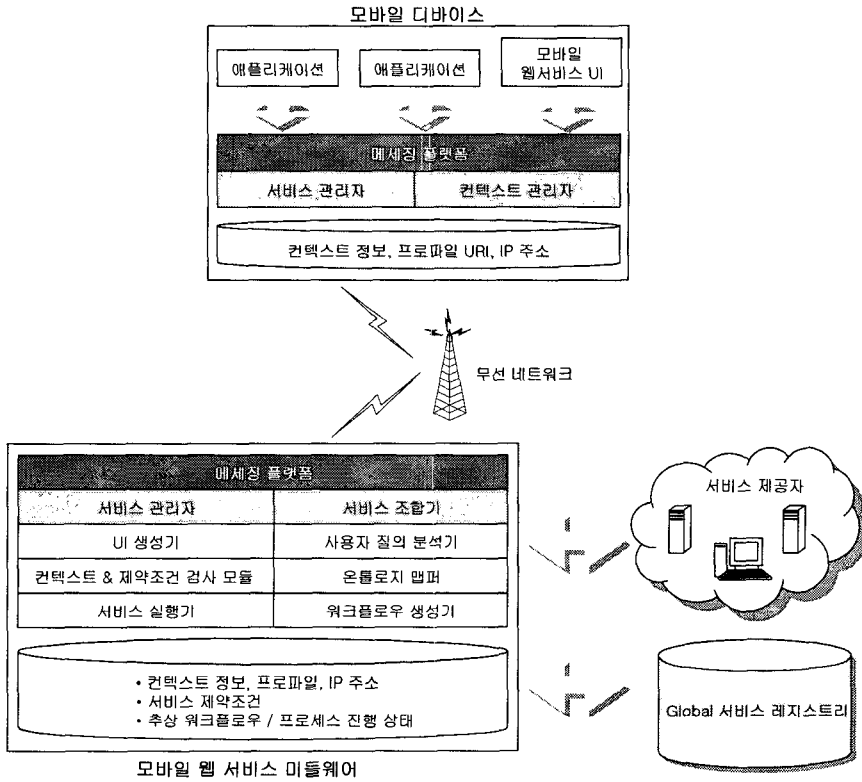


그림 4. 유비쿼터스 웹 서비스 프레임워크의 예

웨어는 개별 디바이스로부터 제공되는 컨텍스트 정보를 바탕으로 각 디바이스가 요구하는 기능을 수행할 수 있는 서비스를 검색하여 제공한다. 이때 개별 디바이스의 하드웨어적 성능과 컨텍스트를 고려하여 멀티모달 인터페이스를 자동으로 생성하여 제공한다. 또한 요구사항을 만족시키는 단일 서비스가 없을 경우 다수의 서비스를 조합하여 복합 서비스를 자동으로 구성하고 서비스를 요청한 사용자나 디바이스의 컨텍스트가 변하는 경우에 따라 서비스를 재구성 혹은 이동시킨다.

이러한 요구에 부응하기에 기존의 웹 서비스는 기술적 한계를 갖는다. 유비쿼터스 컴퓨팅 환경의 장점을 극대화하고 사용자의 편의성을 향상시키기 위해서는 사용자와 디바이스의 컨텍스트에 따라 적합한 서비스를 제공할 수 있는 기능과 컨텍스트 변화에 반응하여 서비스를 동적으로 재구성하거나 이동시킬 수 있는 기능이 필요하다. 또한 모바일 디바이스의 상대적으로 작은 화면과 제한적인 입출력 환경을 고려하여 디바이스에 적합한 멀티모달 인터페이스를 제공해 줄 수 있는 기능이 필수적이다.

### 3. 유비쿼터스 웹 서비스 기술 동향

유비쿼터스 컴퓨팅 환경에서 사용자는 좀 더 편리하고 지능적인 서비스를 요구하게 될 것이다.

#### 3.1 시맨틱 웹 서비스 탐색

현재 웹 서비스 레지스트리 표준인 UDDI는 키워드 기반의 단순 탐색 방법을 제공한다. 탐색에

사용되는 키워드에 따라 너무 많은 탐색 결과를 반환하거나, 원치 않는 서비스들이 탐색 결과에 포함된다. 이러한 문제점을 극복하기 위해 시맨틱 웹 기술을 이용하여 보다 정교한 수준의 탐색을 지원하는 시맨틱 웹 서비스 탐색 방법[10-14]이 제안되었다. 시맨틱 웹 서비스 탐색 방법은 서비스의 기능 및 입출력 정보 등을 온톨로지를 이용하여 기술한 후, 이를 기반으로 탐색하기 때문에 기존 키워드 기반 탐색 방법에 비해 정확성이 높다. 유비쿼터스 환경에서는 사람의 직접적인 개입이 없어도 디바이스 혹은 애플리케이션이 자율적인 판단에 의해 컨텍스트에 따라 사용자에게 유용한 서비스를 제공할 수 있어야 한다. 이를 위해서 요구사항을 만족하는 서비스를 정확히 탐색하는 것이 무엇보다 중요하며 시맨틱 웹 서비스 탐색이 이러한 요구를 만족시킬 수 있다.

그림 5는 시맨틱 웹 서비스 탐색 과정의 예를 보여준다. 서비스 제공자는 도메인 온톨로지를 이용하여 자신의 서비스를 의미적인 정보를 포함하여 기술하고 이를 레지스트리에 출판한다. 서비스 요청자는 먼저 도메인 정보를 획득한 후, 해당 도메인의 온톨로지를 이용하여 요구 사항을 기술하고 이를 기반으로 적합한 서비스를 탐색한다.

시맨틱 웹 서비스 탐색은 서비스와 사용자 요

구사항에 매핑된 온톨로지 개념 간의 매칭을 통하여 적절한 서비스를 추출한다. 이를 위해 다양한 매칭 방법들[10,13,14]이 연구되었으며, 탐색 효율을 향상시키기 위해 서비스 출판 시에 서비스의 매칭 정도를 사전에 저장하거나[12], 탐색 시 네임스페이스, 텍스트, 입출력 타입, 제약조건 필터를 통해 적절한 서비스를 필터링하는 방법 [11]이 제안되었다.

한편 유비쿼터스 환경에서 인접 디바이스간의 웹 서비스를 탐색 및 저장하는 디렉토리 기법을 개발하여야 한다. 특히 웹 서비스의 수가 급증함에 따라 분산 레지스트리에 기반한 웹 서비스의 탐색 및 관리 기법에 대한 연구가 요구된다.

### 3.2 복합 웹 서비스의 자동 구성

유비쿼터스 환경에서 주변 환경과 사용자의 선호 등에 따라 적절한 서비스를 선택 및 제공하는 연구들이 진행 중이다[15-18]. 사용자의 상황에 맞는 적절한 서비스를 제공하기 위해서는 컨텍스트 정보가 고려되어야만 한다. 컨텍스트는 사용자의 위치, 선호, 디바이스의 특징, 네트워크 대역폭 등 사용자가 서비스를 이용하고자 하는 시점에서 고려되어야 할 비기능적 특성을 총칭한다. 이러한 컨텍스트 정보는 일반적으로 온톨로지를 사용하

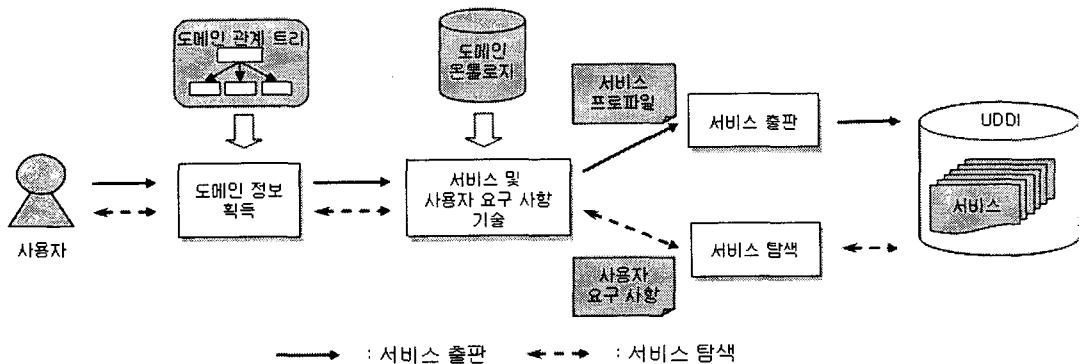


그림 5. 시맨틱 웹 서비스 탐색 과정의 예

여 기술되며 디바이스나 애플리케이션이 사용자의 상황을 인지하고 적합한 서비스를 탐색 및 구성하는데 사용된다. 그림 6은 컨텍스트에 따라 서로 다른 서비스를 조합하여 복합 웹 서비스를 동적으로 구성하는 예를 보여준다.

사용자의 요구가 점차 복잡하고 다양해짐에 따라 단일 서비스로 사용자의 요구 사항을 만족시키는 것은 점점 더 어려워지고 있다. 이에 따라 여러 개의 단일 서비스를 조합하여 사용자의 요구 사항을 만족시키는 복합 웹 서비스 구성이 중요한 이슈가 되었다. 그림 7은 복합 웹 서비스 구성 과정

의 예이다. 사용자가 OWL[19] 이나 OWL-S [20] 와 같은 외부 명세 언어를 사용하여 요구사항을 기술하면 번역기는 이를 STRIPS 또는 Linear Logic과 같은 내부 명세언어로 변환한다. 워크플로우 생성기는 내부 명세언어로 변환된 사용자 요구 사항을 기반으로 추론 과정을 통해 복합 웹 서비스에 대한 추상 워크플로우를 생성한다. 서비스 선택기는 추상 워크플로우에 실제 웹 서비스들을 바인딩하여 BPEL[21] 등의 실행 가능한 복합 서비스를 구성한다.

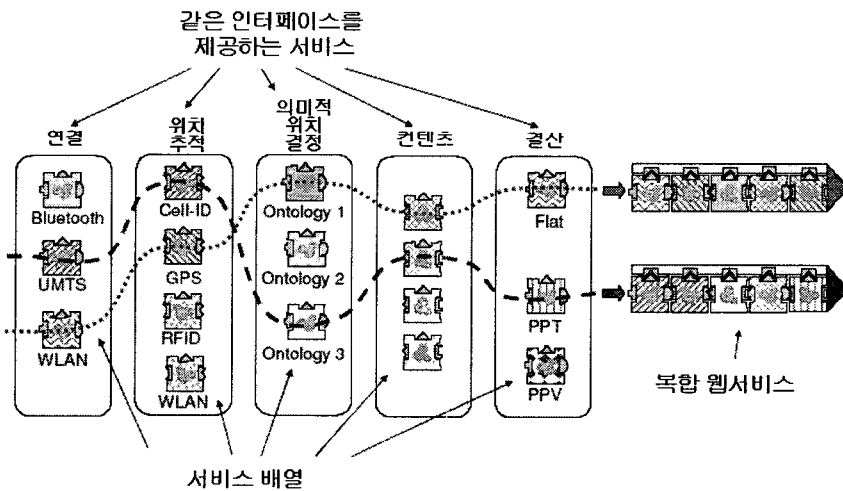


그림 6. 웹 서비스의 동적인 구성의 예

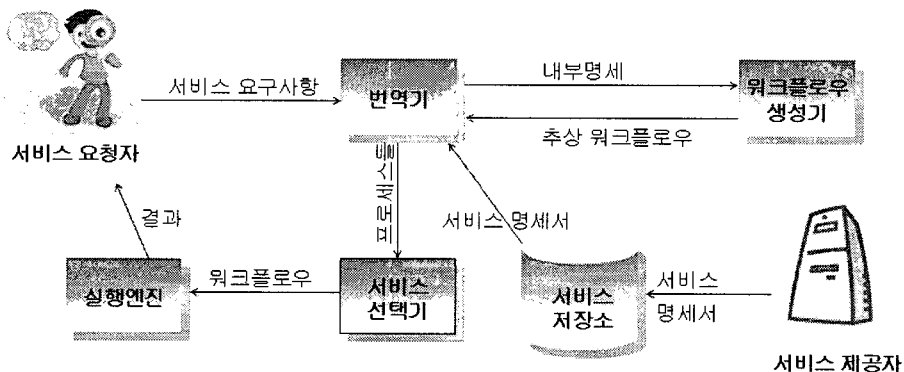


그림 7. 복합 웹 서비스 구성 과정의 예

복합 웹 서비스 구성에 관한 연구는 추상 워크플로우를 생성하는 방법에 따라 크게 두 가지로 나누어질 수 있다. 워크플로우 디자이너가 추상 워크플로우를 수동으로 디자인하는 방법[22-23]과 플래닝(planning) 기법을 적용하여 추상 워크플로우를 자동으로 구성하는 방법[24-26]이다. 전자의 경우 루프, 조건부 분기 등을 사용하여 복잡한 워크플로우를 구성할 수 있지만 워크플로우를 디자인하기 위해서는 주어진 작업과 워크플로우 기술 방법 등에 대해 전문적인 지식이 필요하다. 후자의 경우 기존 인공지능 분야에서 연구되어 온 다양한 플래닝 기법들을 적용하여 사용자의 요구사항 기술로부터 추상 워크플로우를 자동으로 구성한다. 사용자는 단지 요구사항 만들 기술 하면 되므로 전문적인 지식이 필요치 않다. 그러나 루프나 조건부 분기 등으로 표현되어야 하는 복잡한 워크플로우는 생성하기 어렵고 적용되는 플래닝 기법에 따라 도메인 온톨로지 이외의 도메인 지식이 요구되기도 한다.

유비쿼터스 환경에서 서비스 사용자는 전문적인 지식을 가진 비즈니스 업무 종사자들에 국한되지 않는다. 다양한 지식, 연령 및 기호를 가진 평범한 사람들이 서비스 사용자가 될 수 있다. 또한 경우에 따라 사용자가 개입되지 않은 상황에서 디바이스 혹은 애플리케이션이 자율적인 판단에 의해 새로운 복합 서비스를 구성하고 사용자에게 제공할 수 있어야 한다. 이러한 환경에서는 주어진 작업에 대한 요구사항 만으로 복합 서비스를 구성할 수 있는 기능이 요구되며, 플래닝 기법에 기반한 연구들이 이러한 요구를 만족시킬 수 있다.

한편 사용자에게 있어서 서비스가 갖는 기능적 특성 이외에 서비스 가격, 지연 시간 혹은 신뢰성과 같은 비기능적 특성은 서비스를 선택하는 중요한 고려사항이 될 수 있다. 예를 들어, 서로 같은 기능을 제공하는 서비스가 있을 경우, 사용자는

비용이 저렴하고 보다 빠르며 신뢰성이 높은 서비스를 선택할 것이다. 이에 복합 웹 서비스를 구성하는 과정에서 QoS(Quality of Service) 정보를 활용하여야 한다.

복합 웹 서비스의 선택 과정은 바인딩 정보를 포함하지 않은 추상 워크플로우의 각 태스크에 웹 서비스를 할당하여 실행 가능한 워크플로우를 구성하는 과정이다. 전술한 바와 같이 QoS는 서로 다르지만 동일한 기능을 제공하는 다수의 후보 서비스들이 존재하기 때문에 복합 웹 서비스의 QoS는 선택된 웹 서비스에 따라 달라질 수 있다. 따라서 서비스 선택 과정을 통해 특정 QoS 요구사항을 만족하는 서비스나 최적 QoS를 갖는 서비스를 구성할 수 있다. 그림 8은 개념적인 복합 웹 서비스의 선택 과정을 보여준다.

복합 웹 서비스 선택에 관한 연구는 QoS의 최적화 방법에 따라 로컬 최적화와 글로벌 최적화로 나누어질 수 있다. 로컬 최적화 방법[27]은 추상 워크플로우의 개별 태스크 별로 최적의 QoS를 갖는 서비스를 선택한다. 이 방법은 선택 속도가 빠르지만 실제 복합 웹 서비스의 QoS를 고려하지 못한다. 반면 글로벌 최적화 방법[28-31]은 구성 가능한 모든 복합 웹 서비스의 QoS에 대한 최적

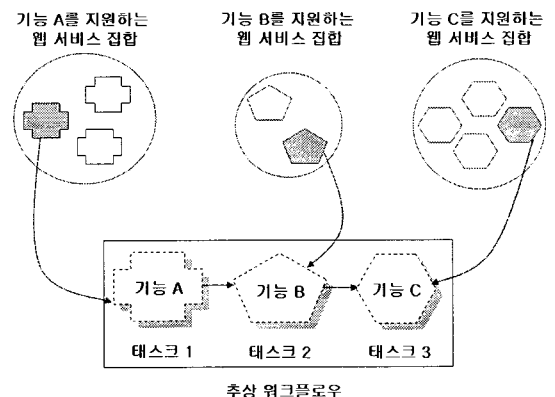


그림 8. 복합 웹 서비스의 선택 과정

화를 수행지만 선택 시간이 느리다.

일반적으로 다수의 품질에 대한 글로벌 최적화 과정은 선형시간에 해결이 불가능한 조합 최적화 문제로 볼 수 있으며, 정수 계획법[28,31] 및 유전자 알고리즘[29,30]과 같이 조합 최적화 문제를 효율적으로 해결할 수 있는 방법을 이용한 접근이 이루어지고 있다. 한편 다수의 웹 서비스가 공존하며 QoS가 동적으로 변하는 유비쿼터스 환경을 고려할 때 주어진 QoS 요구사항을 만족하는 적절한 복합 웹 서비스를 빠르게 선택할 수 있는 방법에 대한 연구가 필요하다.

### 3.2 웹 서비스의 이동 및 경량화

기존 데스크탑 및 유선환경에서 웹 서비스 제공자는 서비스를 제공하는 동안 한 자리에 고정된 채 움직이지 않았다. 그러나 유비쿼터스 환경의 기반이 되는 무선 모바일 환경에서는 모바일 디바이스가 서비스 제공자가 될 수 있다. 이로 인해 서비스 제공자의 위치가 변할 수 있으며, 서비스 제공자의 컨텍스트 역시 변할 수 있다. 이러한 환경에서 사용자에게 지속적으로 유효하고 한결같은 서비스를 제공하기 위해서는 서비스 요청자의

컨텍스트 변화 뿐만 아니라 서비스 제공자의 컨텍스트 변화에 따라 서비스를 이동하거나 재구성할 필요가 있다.

그림 9는 웹 서비스의 이동이 필요한 예를 보여 준다. 차량 A에 탑승한 사용자가 전방 1Km 앞의 교통 상황을 알아보기 위해 1Km 전방에 있는 차량 B가 제공하는 서비스를 이용한다고 가정하자. 이 상황에서 A, B는 교통 흐름에 따라 위치가 변하기 때문에 A와 B의 거리가 1Km보다 멀어지거나 가까워져서 처음 A의 탑승자가 요구했던 정보를 제공할 수 없는 경우가 발생할 수 있다. 이러한 경우 B가 제공하던 웹 서비스를 적절한 위치에 있는 다른 차량으로 이동시킴으로써 차량 A의 탑승자에게 지속적으로 서비스를 제공할 수 있다. 이때 사용자는 웹 서비스의 이동을 느끼지 못하며 웹 서비스가 이동해 가는 차량들을 모두 하나의 가상 서비스 제공자로 인식하게 된다.

유비쿼터스 환경에서 웹 서비스의 이동을 가능케 하기 위해서는 애드 혹 네트워크에서의 컨텍스트 인식 및 동적인 환경 재설정[32]이 필요하다. 또한 디바이스 간의 웹 서비스 이동이 어떤 방식으로 이루어질 것인지 결정되어야만 한다. 이를 위해 클라이언트와 서버 사이에 위치한 프록시가

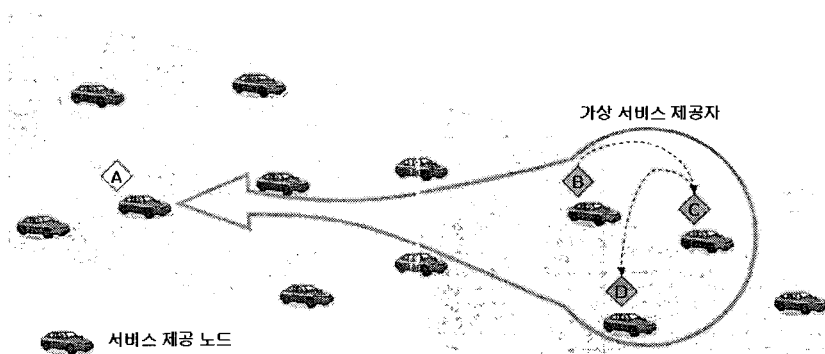


그림 9. 웹 서비스 이동의 예



웹 서비스의 이동을 담당하거나[33], P2P 환경에서 웹 서비스 이동을 지원하는 프레임워크[34] 등이 제안되었다. 또한 실행 플랫폼과 무관하게 웹 서비스를 실행시킬 수 있는 환경에 대한 연구[35]도 진행되고 있다.

한편 모바일 디바이스는 제한된 자원을 갖기 때문에 다양한 기능을 제공하는 웹 서비스를 실행하는데 한계가 있다. 따라서 그림 10과 같이 요구되는 작업에 필요하지 않은 기능들을 동적으로 삭제함으로써 웹 서비스를 경량화 할 수 있는 방

법이 요구된다. 이를 위해 마이크로 서비스 프레임워크와 이를 지원하는 콤팩트한 서버를 통해 경량의 웹 서비스를 구현할 수 있는 방법[36]이 제안되었다.

### 3.3 멀티모달 인터페이스 자동 생성

유비쿼터스 환경에서 웹 서비스 접근을 위한 사용자 인터페이스의 자동 생성이 중요한 이슈가 되고 있다. 그림 11은 모바일 디바이스를 위한 사용자 인터페이스 생성과정의 예이다. 인터페이스

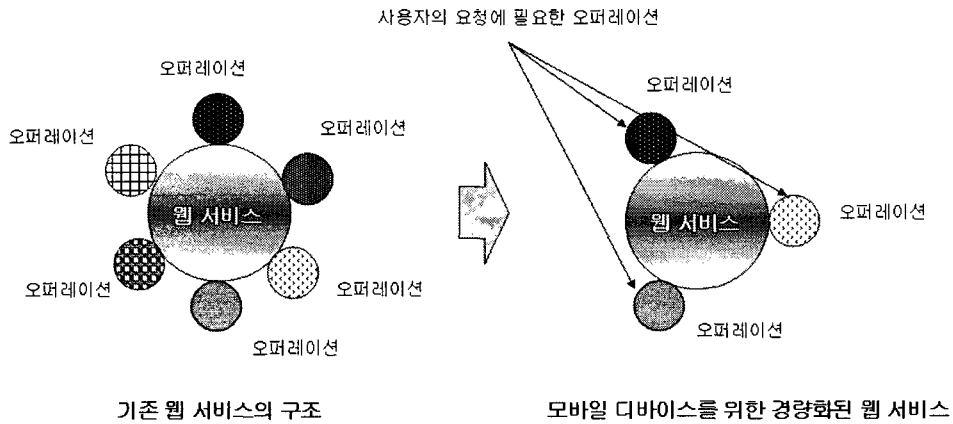


그림 10. 웹 서비스 경량화의 개념도

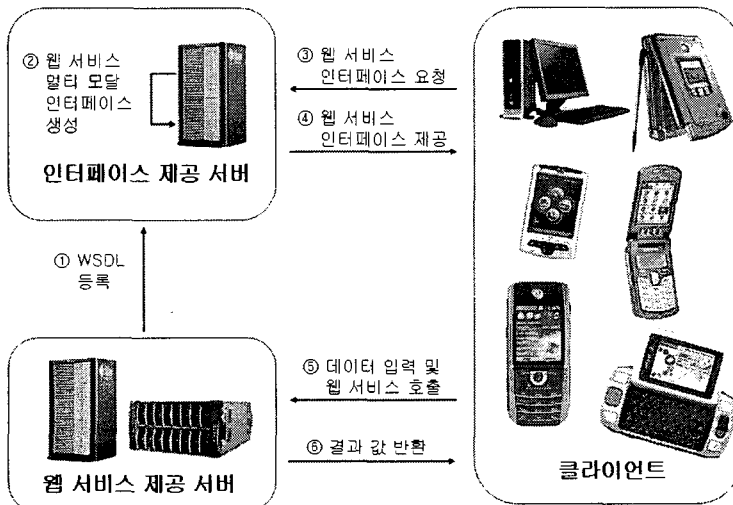


그림 11. 모바일 디바이스를 위한 사용자 인터페이스 생성 과정의 예.

생성 엔진은 웹 서비스 제공 서버로부터 WSDL 문서를 제공받아 웹 서비스 접근을 위한 인터페이스를 생성한 후, 사용자의 요청이 있을 경우 이를 제공한다. 사용자는 제공된 인터페이스를 통해 웹 서비스에 접근하게 되는데 이때 제공된 인터페이스의 작동 방식에 따라 인터페이스 제공 서버를 거쳐 웹 서비스에 접근하기도 한다.

웹 서비스의 인터페이스 생성에 대한 과제는 멀티모달 인터페이스 제공과 복합 웹 서비스에 대한 사용자 인터페이스 제공의 두 가지로 나눌 수 있다. 유비쿼터스 컴퓨팅 환경에서는 다양한 형태의 디바이스가 출현할 것이고 이에 맞추어 사용자에게 각각의 웹 서비스에 대한 적절한 인터페이스를 제공하는 것이 중요하다. 또한 복합 웹 서비스의 경우 여러 개의 단일 서비스로 구성되기 때문에 웹 서비스 시작 시점뿐만 아니라 실행 도중에도 필요한 정보를 사용자로부터 입력받을 수 있는 인터랙티브한 인터페이스의 구성이 요구된다. 이를 위해 BPEL, OWL-S 등 복합 웹 서비스를 기술하는 언어를 지원하여야 한다.

현재 멀티모달 환경을 지향하는 대표적인 인터페이스로서 W3C 표준인 XForms[37]와 VoiceXML[38]이 있다. XForms는 XML 기반의 사용자 인터페이스 개발을 위한 마크업 언어로써 다양한 디바이스에 대한 인터페이스 지원을 목표로 한다. VoiceXML은 컴퓨터와의 음성을 이용한 인터랙션을 정의하는 언어로써 음성 신호를 주요한 매개체로 사용하는 디바이스에 적용된다.

모바일 디바이스에서 웹 서비스 인터페이스 생성을 위해 WSDL과 미리 정의된 GUI 명세를 사용하여 얻은 스크린 데이터에 스타일 시트를 적용하여 적절한 HTML 폼(form)을 생성하고 이를 통해 멀티모달 인터페이스를 제공하는 방법[39]이 제안되었다. 또한 WSDL로부터 XForms 및

VoiceXML에 기반한 문서를 생성하여 멀티모달 인터페이스를 제공하는 방법[40]에 대한 연구가 진행되고 있다.

#### 4. 결론 및 향후 과제

미래 유비쿼터스 환경에서는 사용자의 직접적인 개입 없이 컨텍스트 변화에 따라 적절한 서비스를 찾아 제공할 것이다. 이를 위해서 서비스는 도메인 정보, 컨텍스트, QoS 등의 다양한 정보를 효과적이며 의미 있게 기술할 수 있는 언어를 사용하여 모델링 되어야 한다. 다양한 디바이스와 애플리케이션은 이렇게 정의된 정보를 기반으로 사용자가 원하는 서비스를 제공한다. 그러나 실제 환경에서 서비스와 관련한 모든 정보를 의미적으로 완벽하게 모델링하는 것은 매우 어려운 일이다. 또한 '이러한 정보들을 어떻게 효과적으로 저장, 관리 및 수정할 것인가?' 도 해결되어야 할 문제이다.

한편 특정 컨텍스트에서 디바이스나 애플리케이션의 잘못된 판단은 사용자에게 불편을 초래하거나 심각한 상황을 야기할 수 있다. 따라서 디바이스 및 애플리케이션이 상황을 정확하게 판단하는 것이 매우 중요하다. 그러나 수집된 컨텍스트 정보를 기반으로 사용자의 상태나 의도, 그리고 서비스 제공 후 사용자의 만족도를 예측하고 이에 따라 적절한 서비스를 선택하는 것은 매우 어려운 일이다. 예를 들어, 사용자가 자동차 열쇠를 지닌 채로 자신의 차에 접근할 때 '시동을 걸어줄 것인가?', '트렁크를 열어줄 것인가?'를 판단하는 것은 어렵다. 이처럼 상황을 정확하게 판단 및 예측하고 적절한 서비스를 제공해 주는 것은 연구되어야 할 과제이다.

유비쿼터스 환경에서는 각자에게 주어진 작업을 수행하기 위하여 수많은 디바이스와 애플리케이션

이선이 서로 의사소통 및 상호작용하게 된다. 사용자가 불편함을 느끼지 않게 하기 위해서는 각각의 주어진 작업에 대해 사용자를 오래 기다리게 해서는 안 된다. 동시에 여러 디바이스나 애플리케이션이 각자의 작업을 수행하는 경우에도 사용자가 인내할 수 있는 시간 안에 결과를 반환하여야 한다. 그러나 현재 시맨틱 웹 서비스 탐색, 복합 웹 서비스 자동 구성 등은 대부분 탐색공간이 커지고 질의의 수가 많아짐에 따라 처리 속도가 지수 복잡도로 증가한다. 실제 유비쿼터스 환경에서 원활한 서비스를 제공하기 위해서 이러한 확장성 문제를 개선할 수 있는 방법이 고려되어야 한다.

웹 서비스가 유비쿼터스 환경에서 핵심 소프트웨어 기술로 자리 잡기 위해서는 전술한 바와 같이 아직 해결해야 할 과제들이 남아있다. 하지만 국내외적으로 유비쿼터스 환경에 대한 기대와 관심이 고조되면서 유비쿼터스 환경에서의 웹 서비스 기술에 대한 연구가 활발하게 이루어지고 있다. 머지않아 무선 모바일 환경을 기반으로 한 유비쿼터스 컴퓨팅 시대가 도래할 것이며 웹 서비스는 이러한 환경에서 다양하고 이질적인 디바이스와 애플리케이션 간의 인터페이스 역할을 담당하게 될 것이다. 뿐만 아니라 시맨틱 웹, 인공지능 기술 등과 접목되어 사용자에게 더욱 편리하고 질 높은 서비스를 제공할 것이다.

## 참 고 문 헌

- [1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM*, Vol. 36, No. 7, pp. 75-84, 1993.
- [2] Aware Home, Georgia Tech, <http://www.cc.gatech.edu/fce/ahri>.
- [3] Aura, CMU, <http://www.cs.cmu.edu/~aura/>.
- [4] Oxygen, MIT, <http://oxygen.lcs.mit.edu/>.
- [5] Pervasive Computing at IBM Research, <http://www.research.ibm.com/compsci/mobile/>.
- [6] M. Hendricks, B. Galbraith, R. Irani, J. Mibery, T. Modi, A. Tost, A. Toussaint, S. J. Basha, and S. Cable, "Professional Java Web Services," WROX Press, pp. 1-16, 2002.
- [7] WSDL(Web services description language), <http://www.w3.org/TR/wsdl20/>.
- [8] SOAP(Simple Object Access Protocol), <http://www.w3.org/TR/soap12/>.
- [9] UDDI(Universal Description, Discovery and Integration), <http://www.uddi.org>.
- [10] M. C. Jaeger, G. Rojec-Goldmann, C. Liebetrueth, G. Muhl, and K. Geihs, "Ranked Matching for Service Descriptions using OWL-S," *Proc. Kommunikation in Verteilten Systemen*, pp. 91-102, 2005.
- [11] T. Kawamura, T. Hasegawa, M. Paolucci, and K. Sycara, "Web Service Lookup: A Matchmaker Experiment," *IT Professional*, Vol. 7, Issue 2, pp. 36-41, 2005.
- [12] N. Srinivasan, M. Paolucci, and K. Sycara, "An Efficient Algorithm for OWL-S Based Semantic Search in UDDI," *Proc. Int'l Conf. Semantic Web Services and Web Process Composition(LNCS 3387)*, pp. 96-110, 2005.
- [13] I. Elgedawy, Z. Tari, and M. Winikoff, "Exact Functional Context Matching for Web Services," *Proc. Int'l Conf. Service Oriented Computing*, pp. 15-19, 2004.
- [14] R. Aggarwal, K. Verma, and J. Miller, W. Milnor, "Constraint Driven Web Service Composition in METEOR-S," *Proc. IEEE Int'l Conf. Services Computing*, pp. 23-30, 2004.
- [15] 한주현, 김은희, 최재영, 조위덕, "유비쿼터스 컴퓨팅 환경을 위한 웹 서비스 기반의 상황인지 워크플로우 언어," *정보과학회논문지, 시스템 및 이론* 제32권 제11-12호, 2005 12월호.
- [16] D. Preuveneers and Y. Berbers, "Automated Context-Driven Composition of Pervasive

- Services to Alleviate Non-Functional Concerns," *International Journal of Computing & Information Sciences*, Vol. 3, No. 2, pp. 19-28, 2005.
- [17] M. Vukovic and P. Robinson, "Adaptive, planning-based, Web service composition for context awareness," *Int'l Conf. on Pervasive Computing*, Vienna, 2004.
- [18] S. B. Mokhtar, D. Fournier, N. Georgantas, and V. Issamy, "Context-aware Service Composition in Pervasive Computing Environments," *Proc. the 2nd Int'l Workshop on Rapid Integration of Software Engineering Techniques (RISE'2005)*, September 2005.
- [19] OWL, <http://www.w3.org/TR/owl-features/>.
- [20] OWL-S, <http://www.daml.org/services/owl-s/1.1/>.
- [21] BPELAWS, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>.
- [22] R. Aggarwal, K. Verma, J. Miller, and W. Milnor, "Constraint Driven Web Service Composition in METEOR-S," *Proc. IEEE Int'l Conf. on Services Computing*, pp. 23-30, 2004.
- [23] S. Ghandeharizadeh, C. A. Knoblock, C. Papadopoulos, C. Shahabi, E. Alwagait, J. L. Ambite, M. Cai, C. Chen, P. Pol, R. Schmidt, S. Sorg, S. Thakkar, and R. Zhou. "Proteus: a system for dynamically composing and intelligently executing web services," *Proc. Int'l Conf. On Web Services(ICWS)*, 2003.
- [24] S. R. Ponnkanti and A. Fox, "SWORD: A developer toolkit for Web service composition," *Proc. 11th World Wide Web Conf.*, Honolulu, USA, 2002.
- [25] D. Wu, E. Sirin, J. Hendler, D. Nau, and B. Parsia, "Automatic Web services composition using SHOP2," *Workshop on Planning for Web Services*, Trento, Italy, 2003.
- [26] J. Rao, P. Küngas, and M. Matskin, "Logic-based Web services composition: from service description to process model," *Proc. Int'l Conf. on Web Services*, San Diego, USA, July 2004.
- [27] R. Grønmo and M. C. Jaeger, "Model- Driven Methodology for Building QoS- Optimized Web Service Compositions," *Proc. 5th Int'l Conf. on DAIS(LNCS 3543)*, pp. 68-82, 2005.
- [28] A. Gao, D. Yang, S. Tang, and M. Zhang, "QoS-Driven Web Service Composition with Inter Service Conflicts," *Proc. 8th APWeb Conf. (LNCS 3841)*, pp. 121-132, 2006.
- [29] S. Liu, Y. Liu, N. Jing, G. Tang, and Y. Tang, "A Dynamic Web Service Selection Strategy with QoS Global Optimization Based on Multi-objective Genetic Algorithm," *Proc. 4th Int'l Conf on Grid and Cooperative Computing(LNCS 3795)*, pp. 84-89, 2005.
- [30] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An Approach for QoS- aware Service Composition based on Genetic Algorithms," *Proc. Conf. Genetic and Evolutionary Computation*, pp. 1069-1075, 2005.
- [31] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-Aware Middleware for Web Services Composition," *IEEE Trans. on Software Engineering*, Vol. 30, No. 5, pp. 311-327, 2004.
- [32] O. Riva, T. Nadeem, C. Borcea, and L. Iftode, "Mobile Services: Context-Aware Service Migration in Ad Hoc Networks," *Under Submission to The Third Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2005)*.
- [33] E. Alwagait and S. Ghandeharizadeh, "DeW: A Dependable Web Services Framework," *Proc. 14th Int'l Workshop on Research Issues on Data Engineering: Web Services for ECommerce and E-Government Applications*, pp. 111-118, 2004.
- [34] T. Mifsud and P. Stanski, "Peer-2-Peer Nomadic Web Services Migration Frame-

work,” Proc. The Second AMOC, 2002.

- [35] D. Kossmann and C. Reichel, “SLL: Running My Web Services on Your WS Platforms,” Int’l Conf. on Web Services (Industry Track), pp. 962-963, 2005.
- [36] N. Nicoloudis and D. Pratiatha, “Microservices: A Lightweight Web Service Infrastructure for Mobile Devices,” Proc. The Second Int’l Conf. on Pervasive Computing, 2004.
- [37] XForms. <http://www.w3.org/MarkUp/Forms/>.
- [38] VoiceXML. <http://www.w3c.org/TR/voicexml21/>
- [39] R. Steele, K. Khankan, and T. Dillon, “Mobile Web Services Discovery and Invocation Through Auto-Generation of Abstract Multimodal Interface,” Proc. Int’l Conf. Information Technology: Coding and Computing, Vol. 02, pp. 35-41, 2005
- [40] M. Kassoff, D. Kato, W. Mohsin, “Creating GUIs for Web Services,” IEEE Internet Computing, Vol. 7, No. 5, pp. 66-73, 2003.



신 동 훈

- 2003년 2월 연세대학교 컴퓨터과학과 졸업(학사)
- 2005년 2월 연세대학교 컴퓨터과학과 졸업(석사)
- 2005년 3월~현재 연세대학교 컴퓨터과학과 박사과정
- 관심분야 : Internet Computing, Service-Oriented Computing, Multimedia Document Engineering



이 경 호

- 1995년 2월 연세대학교 전산과학과 졸업(학사)
- 1997년 2월 연세대학교 컴퓨터과학과 졸업(석사)
- 2001년 2월 연세대학교 컴퓨터과학과 졸업(박사)
- 2001년 3월 National Institute of Standard and Technology(NIST) 객원연구원
- 2002년 9월~현재 연세대학교 컴퓨터과학과 조교수
- 관심분야 : Internet Computing, Service-Oriented Computing, Multimedia Document Engineering