Journal of Korea Multimedia Society Vol. 11, No. 1, January 2008(pp. 106-116)

컨텍스트에 기반한 웹 서비스 이동

김연석[†], 이경호[††]

요 약

불안정한 연결성 및 빈번한 컨텍스트 변화 등은 모바일 환경에서 웹 서비스를 제공하는데 많은 문제점을 야기시킨다. 이를 해결하기 위해서는 사용자에게 지속적으로 서비스 제공이 가능한 디바이스로의 웹 서비스 이동이 필수적이다. 그러나 모바일 디바이스에서의 웹 서비스 이동, 복제, 그리고 호스팅에 관한 연구는 아직 초기단계에 불과하다. 본 논문에서는 모바일 환경에서의 웹 서비스 이동을 위하여 경량의 웹 서비스 프레임웍과 함께 효과적인 이동정책 수립기법을 제안한다. 제안된 방법은 WS-Policy를 기반으로 다양한 컨텍스트 정보를 기술할 수 있는 스키마를 정의하고, 이를 사용하여 이동정책을 수립한다. 특히, 제안된 스키마는 서비스 제공자의 요구사항을 자유롭게 표현하기 위하여 확장 가능하도록 설계되었다. 이동정책의 유효성을 검증하기 위하여, 제안된 방법을 모바일 웹 서비스 프레임웍에 적용한 결과, 웹 서비스 이동을 통한 연속적인 서비스의 제공이 가능함을 보였다.

Web Service Migration based on Contexts

Yeon-Seok Kim[†], Kyong-Ho Lee[††]

ABSTRACT

It is difficult to provide Web services on a mobile environment because of the intermittent disconnection of a wireless network and the frequent change of context information. To resolve this, Web services must be migrate to devices, which can provide services to users seamlessly. However, researches on hosting, migrating, and replicating Web services on mobile devices are still in an early stage. This paper presents a lightweight Web service framework and an efficient method for establishing a policy of migrating Web services. Based on the WS-Policy standard, the proposed method defines a policy schema, which provides constructs for describing various kinds of contextual information. Specifically, the schema is designed to be extensible enough for different service providers to use different vocabularies to describe their policies. To show the validity, the proposed method for establishing a migration policy has been applied to our mobile Web services framework, resulting in a seamless provisioning of services.

Key words: Mobile Web Service(모바일 웹 서비스), Migration Policy(이동 정책), Context(컨텍스트)

1. 서 론

웹 서비스(Web Services)[1]는 인터넷과 같이 공개적인 네트워크 및 관련 표준기술을 통해 기존의 애플리케이션을 운영체제 및 프로그래밍 언어에 상관없이 상호운용이 가능하도록 해주는 표준화된 소프트웨어 기술이다. 이러한 웹 서비스는 다양한 디바이스가 혼재되어 있는 유비쿼터스 환경에서 사용자가 필요로 하는 서비스의 제공을 가능하게 함으로서 유비쿼터스 환경을 위한 필수적인 소프트웨어 기술

※ 교신저자(Corresponding Author) : 이경호, 주소 : 서울시 서대문구 신촌동 134 (120-749), 전화 : 02)2123-3878, FAX : 02)365-2579, E-mail : khlee@cs.yonsei.ac.kr
접수일 : 2007년 3월 19일, 완료일 : 2007년 11월 2일
[†] 준회원, 연세대학교 컴퓨터과학과
(E-mail : yskim@icl.yonsei.ac.kr)
[††] 종신회원, 연세대학교 컴퓨터과학과
※ 이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임. (KRF-2005-041-D00690)

로 인식되고 있다. 한편, 기존 웹 서비스의 대부분은 데스크탑 서버 및 유선 환경을 고려하여 개발되었기 때문에 무선 환경에 적용하기에는 문제가 많다. 그러나 머지않아 모바일 디바이스는 단순한 클라이언트 디바이스가 아닌 하나의 독립적 서비스를 제공하는 강력한 서버 형태로 변모되어 갈 것이다. 즉, 모바일 디바이스는 사용자가 원하는 서비스를 제공하는 서비스 제공자의 역할을 수행할 것이다[2-4].

모바일 디바이스를 사용하여 웹 서비스를 제공할 경우 네트워크의 불안정한 연결성 및 빈번한 컨텍스트 변화 등은 지속적인 서비스 제공을 어렵게 만든다. 예를 들어, 서비스 제공자가 서비스를 더 이상 제공할 수 없는 위치로 이동할 경우 사용자는 더 이상 서비스를 받을 수 없다. 그러나 해당 서비스를 적절한 디바이스로 이동시킨다면 사용자는 한결같은(seamless) 서비스를 제공받을 수 있을 것이다. 즉, 모바일 환경에서 사용자에게 한결같은 서비스를 제공하기 위해서는 웹 서비스의 이동이 필수적이다.

기존에 컨텍스트 기반의 웹 서비스 이동에 관한 연구가 다양하게 진행되었다. 그러나 기존 연구의 대부분은 모바일 환경을 대상으로 하지 않거나 사용자의 개입이 필요하다는 단점을 가진다. 뿐만 아니라 모바일 환경에서의 웹 서비스 이동, 복제, 그리고 호스팅에 관한 연구는 아직 초기단계에 불과하다.

Hao 등 [5] 은 실시간 어플리케이션의 성능 향상을 위하여 웹 서비스 이동 및 실행을 지원하는 weblet 인프라와 비용모델을 제안한다. 이 방법은 이동 가능한 경량의 웹 서비스를 정의하고 웹 서비스의 실행을 지원하는 weblet 인프라를 구축한다. 특히, 웹 서비스의 타겟 호스트를 결정하기 위하여 유전자 알고리즘(Genetic Algorithm)을 적용한다. 그러나 이동시점의 결정을 위해서 사용자의 개입을 필요로 한다. Pratistha 등 [6,7] 은 웹 서비스 이동을 지원하는 프레임웍인 Fluid와 함께 이동 호스트의 결정을 위한 비용모델 모듈(Cost Model Module)을 제안한다. 이 방법은 다양한 비용모델 모듈에서 서비스 제공자가 이동에 필요한 모듈을 선택하면, 이후 웹 서비스 이동이 결정되었을 때 선정된 비용모델 모듈에 의거, 가장 적절한 호스트로 서비스를 이동시킨다. 그러나 제안된 방법은 모바일 환경을 고려하지 않으며, 서비스 제공자가 직접 비용모델 모듈을 선택해야만 하는 단점이 있다. Hemmati 등 [8] 은 서비스 코드와 실행 상태의 이동을 지원하는 서비스 이동 프레임웍을 제안한다. 제안된 프레임웍은 컨텍스트 매니저에 의해 수집된 컨텍스트 정보를 기반으로 적절한 호스트로의 이동을 결정한다. 특히, 서비스 이동의 효율은 소스 호스트에서 서비스가 실행된 정도에 따라, 그리고 목적 호스트에서 서비스를 재실행 했을 때의 효율에 따라 그 정도가 달라진다. 그러나 이 방법은 웹 서비스를 대상으로 하지 않는다. Riva 등 [9] 은 애드 혹 네트워크에서의 동적인 컨텍스트 변화를 지원하는 모바일 서비스 이동 프레임웍을 제안한다. 이 방법은 서비스 요청자의 컨텍스트를 모니터링 하여 모바일 서비스를 적절한 노드로 이동시킨 다음 서비스를 제공한다. 그러나 제안된 프레임웍은 사용자의 요청에 의한 컨텍스트의 모니터링이 필요하다는 단점을 갖는다. Corradini 등 [10] 은 모바일 코드의 이동을 위한 프레임웍을 제안한다.

본 논문에서는 서비스의 효과적인 제공을 위하여 웹 서비스 이동 및 복제를 지원하는 모바일 웹 서비스 프레임웍과 함께 언제, 어디로 이동할 것인지를 결정하기 위한 이동정책 수립기법을 제안한다. 제안된 프레임웍은 모바일 환경을 위하여 경량화된 구조를 가지며, 6개의 매니저, 즉, 디렉토리 매니저, 실행 매니저, SOAP 매니저, 이동 매니저, 컨텍스트 매니저 그리고 출판 및 검색 매니저로 구성된다. 한편, 제안된 방법은 이동정책 수립을 위하여 스키마를 정의한다. 제안된 스키마는 서비스 제공자의 요구사항을 자유롭게 표현하기 위하여 UAProf을 포함한 CC/PP 프로파일 등을 적용할 수 있도록 확장 가능하게 설계되었다. 뿐만 아니라 이동시점 및 타겟 호스트의 결정을 위하여 컨텍스트에 따른 우선순위를 표현할 수 있다.

제안된 방법은 스키마를 기반으로 서비스 제공자로부터 필요한 정보를 입력받아 이동정책을 수립한다. 이후, 모바일 웹 서비스 프레임웍은 이동정책을 기반으로 컨텍스트를 모니터링하고, 그 결과에 따라 웹 서비스를 이동시킨 후 서비스를 재실행한다. 이동정책의 유효성을 검증하기 위하여 모바일 웹 서비스 프레임웍에 제안된 방법을 적용한 결과, 웹 서비스 이동을 통한 연속적인 서비스를 제공이 가능함을 보였다.

본 논문의 구성은 다음과 같다. 2절에서는 웹 서비스 프레임웍을 6개의 모듈로 구분하여 설명하고, 이

에 기반한 웹 서비스 호출 및 응답 절차에 대하여 간략히 기술한다. 3절에서는 본 논문에서 제안한 스키마 및 이동정책 생성을 위한 사용자 인터페이스에 대하여 자세히 기술하고, 4절에서는 제안된 이동정책 생성기법의 유효성 검증을 통하여 모바일 환경에서의 지속적인 서비스 제공이 가능함을 보인다. 마지막으로 5절에서는 결론 및 향후 연구방향을 기술한다.

2. 모바일 웹 서비스 프레임웍

본 절에서는 웹 서비스 이동을 지원하는 모바일 웹 서비스 프레임웍에 대하여 간략히 기술한다. 제안된 프레임웍은 그림 1과 같이 6개의 매니저로 구성된다.

2.1 프레임웍의 구성

제안된 모바일 웹 서비스 프레임웍은 다음과 같이 6개의 매니저, 즉, 디렉토리 매니저, 실행 매니저, SOAP 매니저, 이동 매니저, 컨텍스트 매니저 그리고 출판 및 검색 매니저로 구성된다.

- 디렉토리 매니저

디렉토리 매니저는 모바일 디바이스에 존재하는 웹 서비스를 관리한다. 모바일 디바이스는 그 특성상 다양한 제약조건 즉, 프로세스, 메모리 용량, 배터리 지속시간 등을 갖기 때문에 리소스를 효율적으로 사용해야만 한다. 이를 위하여 디렉토리 매니저는 웹 서비스를 액티브(Active) 모드와 디액티브(Deactive) 모드로 구분한 후 액티브 모드의 웹 서비스만을 사용하도록 설정한다. 또한 애드-혹 환경에서의 웹 서비스 제공을 위하여 WSDL 문서를 저장하고 서비스 요청 시 이를 제공하는 역할을 한다.

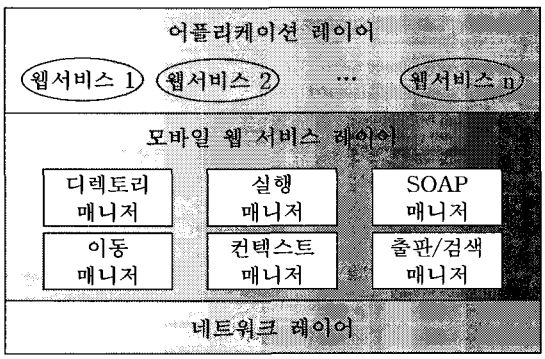


그림 1. 모바일 웹 서비스 프레임웍

- 실행 매니저

모바일 환경에서의 웹 서비스 실행은 유선 환경에서의 그것과 유사하다. 우선, 서비스 제공자가 사용자로부터 요청 메시지를 받으면 실행 매니저는 디렉토리 매니저에 있는 WSDL 문서를 검색한 후 이에 기반하여 사용자의 입력을 요구한다. 만약, 사용자가 적절한 입력을 제공하면 실행 매니저는 요청된 웹 서비스의 인스턴스를 생성한 후 해당 함수를 호출함으로써 서비스를 실행한다.

- SOAP 매니저

SOAP 매니저는 서비스 제공자와 사용자간의 통신을 위하여 SOAP 메시지의 생성 및 파싱을 담당한다. 기존에 모바일 환경을 위한 다양한 SOAP 툴킷이 개발되어 왔다. WSDL 파서와 stub/skeleton 컴파일러를 포함한 C++용 gSOAP을 비롯하여, J2ME 디바이스를 위한 오픈소스 API인 kSOAP 등이 그 예이다. 그러나 kSOAP은 WSDL 문서로부터 사용자 stub을 생성할 수 없다는 단점을 가진다. 한편, 제안된 SOAP 매니저는 편의를 위하여 윈도우 기반이며, 경량의 SOAP 툴킷인 PocketSOAP 툴킷을 사용한다. PocketSOAP의 용량은 약 680K에 불과하다.

- 이동 매니저

모바일 환경에서의 연속적인 서비스 제공을 위해서는 웹 서비스의 이동이 필수적이다. 웹 서비스의 이동은 서비스 제공자의 요청에 의하거나 혹은 디바이스의 배터리가 더 이상 지속될 수 없을 때, 서비스가 제공될 수 없는 위치로 디바이스가 이동했을 때 등 다양한 컨텍스트 변화에 의거 결정된다. 이때, 이동 매니저는 웹 서비스의 코드와 인스턴스 그리고, 디렉토리 매니저에 존재하는 WSDL 문서 등을 적절한 디바이스로 이동시킨다. 특히, 이동 매니저는 제안된 이동정책을 사용하여 이동 가능한 호스트 중 가장 비용이 낮은 호스트를 타겟 호스트로 결정한다.

- 컨텍스트 매니저

제안된 방법은 웹 서비스가 이동 할 타겟 호스트의 결정을 위하여 이웃한 호스트로부터 필요한 컨텍스트를 수집하고 관리한다. 예를 들어, 서비스 제공자가 웹 서비스의 이동을 요청하면, 컨텍스트 매니저는 네트워크를 구성하는 이웃한 호스트에 대하여 필요한 컨텍스트 정보를 요청하고 수집한다. 수집된 컨텍스트 정보는 해당 웹 서비스의 이동정책에 적용되어 타겟 호스트를 결정하는 데 사용된다. 한편, 컨텍

스트 매니저는 주기적인 컨텍스트의 갱신을 통하여 네트워크의 트래픽을 감소시킨다.

- 출판/검색 매니저

일반적인 의미의 웹 서비스에서, 웹 서비스 제공자는 해당 서비스를 설명하는 WSDL 문서를 브로커에 출판하고, 사용자는 브로커와의 통신을 통하여 서비스를 검색한다. 그러나 모바일 환경 특히, 애드-혹 환경에서의 서비스 제공자 및 사용자는 HTTP 프로토콜을 사용할 수 없다. 따라서 웹 서비스를 출판하고 검색하는 새로운 방법이 필요하다. 제안된 프레임웍은 출판 및 검색 매니저를 통하여 웹 서비스를 출판하고 검색한다. 출판 및 검색 매니저는 모바일 디바이스에 있는 액티브 모드의 웹 서비스를 출판하는데 특히, 일반적인 웹 서비스의 출판과는 달리 서비스의 이름만을 출판하기 때문에 웹 서비스는 자신의 기능을 표현할 수 있는 이름을 사용해야만 한다.

2.2 웹 서비스 호출 및 응답절차

제안된 프레임웍에 기반한 웹 서비스의 호출 및 응답절차는 그림 2와 같이 유선환경에서의 그것과 유사하다. 우선, 사용자가 자신의 디렉토리 매니저를 통하여 원하는 서비스를 검색했다면 해당 호스트에게 웹 서비스의 WSDL 문서를 요청한다. 서비스 제공자의 실행 매니저는 사용자의 요청 메시지를 받은 후 자신의 디렉토리 매니저로부터 WSDL 문서를 검색하고 다시 사용자의 입력을 위하여 해당 문서를 사용자에게 재전송하게 된다. 만약 사용자가 적절한 입력 값을 제공하여 서비스를 호출하면, 서비스 제공자의 실행 매니저는 SOAP 매니저를 통하여 SOAP 메시지를 파싱하고, 호출된 웹 서비스의 인스턴스를 생성한 후 사용자가 원하는 서비스를 실행한다. 이후 그 결과를 다시 SOAP 매니저를 통하여 SOAP 메시지로 변환한 후 사용자에게 적절한 정보를 제공하게 된다. 한편, 사용자의 SOAP 매니저는 수신된 결과를 파싱한 후 적절한 인터페이스를 통하여 사용자에게 정보를 제공한다.

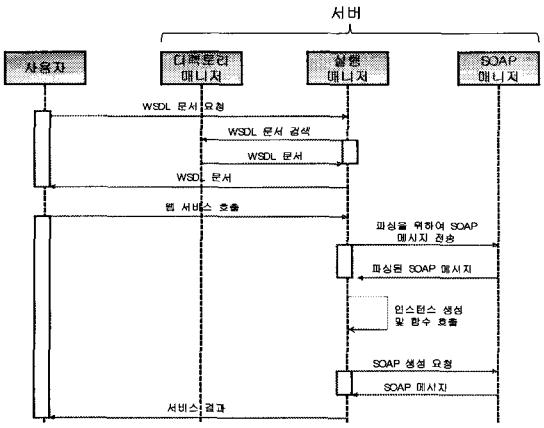


그림 2. 웹 서비스 호출 및 응답 절차

3. 이동정책 수립기법

웹 서비스의 이동을 위해서는 해당 웹 서비스가 언제, 어디로 이동해야 할 것인지를 명세한 이동정책이 필수적이다. 이를 위하여 제안된 방법은 W3C의 표준인 WS-Policy [11] 에 기반하여 이동정책 수립기법을 제안한다. WS-Policy란 웹 서비스의 WSDL 문서가 기술하지 못하는 다양한 명세 즉, 서비스 요구 사항, 선호도, 기능 등의 서술이 가능한 표준으로서, WSDL 문서에 쉽게 귀속될 수 있다는 장점을 가진다. 또한 하나의 서비스에 대하여 다양한 정책을 수립할 수 있기 때문에 컨텍스트 및 상태 변화에 따른 적절한 대응이 가능하다. 제안된 방법은 그림 3과 같이 웹 서비스의 이동정책 수립을 위한 스키마를 정의하며, 웹 서비스 제공자로부터 최소한의 정보를 입력받아 이동정책을 수립한다.

3.1 제안된 스키마 및 사용자 인터페이스

제안된 스키마는 크게 이동 및 복제를 위한 "Migration_Policy"와 "Replication_Policy" 엘리먼트로 구성된다. 이는 이동 및 복제정책을 구분함으로써 상황에 따른 적절한 대응이 가능하게 하기 위함이다. 한편, "Migration_Policy"과 "Replication_olicy" 엘리먼트는 각각 언제, 어디로를 나타내는 "When"과 "Where" 엘리먼트를 자식 엘리먼트로 갖는다. "When"과 "Where" 엘리먼트는 이동시점과 타겟 호스트를 결정하기 위하여 필요한 하드웨어 및 소프트웨어, 그리고 네트워크 정보 등을 자식으로 갖는 엘리먼트들이다.

제안된 방법은 컨텍스트 변화에 따른 단계별 이동정책 수립을 위하여 WS-Policy의 "ExactlyOne" 및 "All" 엘리먼트를 사용한다. "ExactlyOne" 및 "All" 엘리먼트는 각각 정책집합(policy collection)과 확언

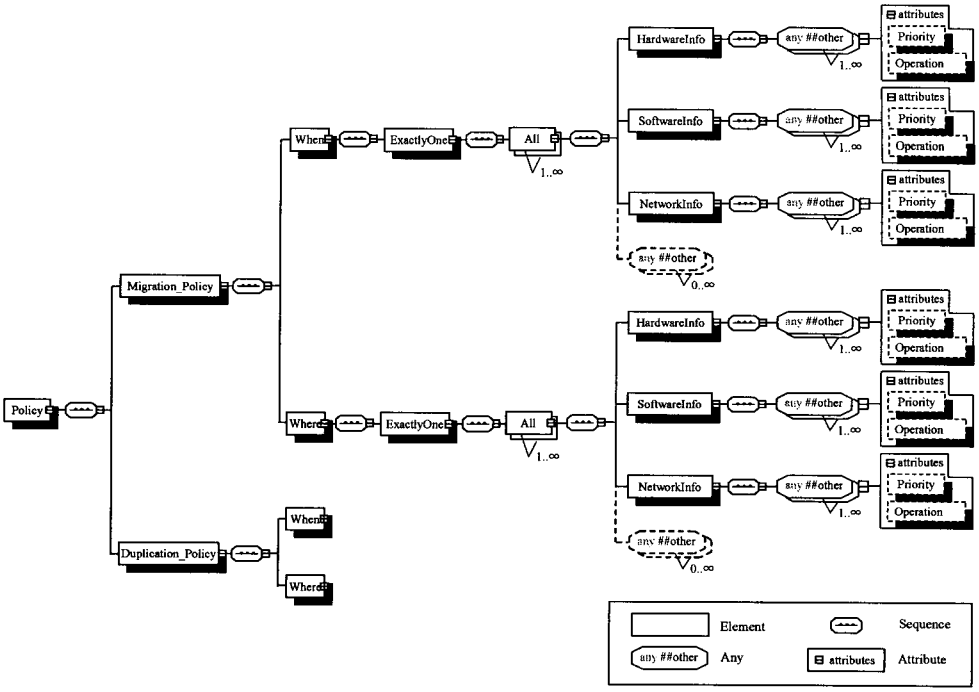
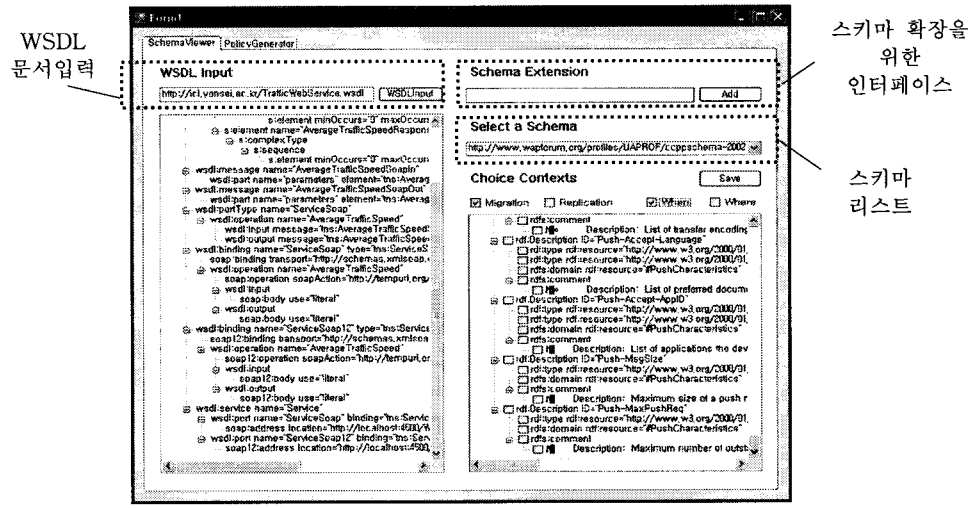


그림 3. 웹 서비스 이동정책 설정을 위한 스키마

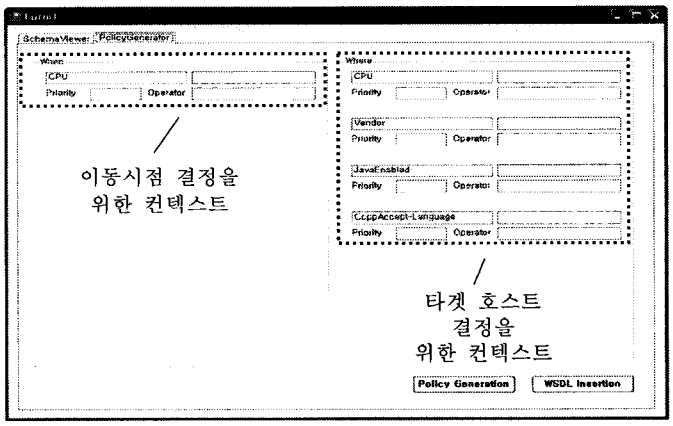
집합(assertion collection)을 표현하기 위한 엘리먼트들로서 동일한 컨텍스트 일지라도 그 값에 따라 적절한 이동정책을 수립할 수 있게 한다. 또한 제안된 스키마는 UAProf을 포함한 CC/PP 프로파일 등 기존의 컨텍스트 모델을 적용하여 확장 가능하도록 설계되었다. 즉, 스키마의 확장을 통하여 서비스 제공자는 그 요구사항을 이동정책에 자유롭게 표현할 수 있다. 한편, 제안된 방법은 웹 서비스의 이동시점 및 타겟 호스트의 결정을 위하여 서비스 제공자로부터 컨텍스트의 우선순위 및 조건 연산자의 입력을 요청한다. 예를 들어, 디바이스의 성능저하로 인하여 특정 지역의 교통정보를 제공하는 웹 서비스가 이동되어야 한다고 가정할 때, 서비스 제공자는 해당 지역을 첫 번째 우선순위로, 성능을 다음 우선순위로 설정할 수 있다. 이때, 해당 컨텍스트의 "Priority" 속성을 사용한다. 또한 메모리, CPU 등과 같이 그 사용 정도에 따라 이동이 결정되어야 할 경우, 제안된 방법은 해당 컨텍스트의 "operation" 속성을 사용, 조건 연산자(Conditional operator) 즉, 크다, 작다, 크거나 같다, 작거나 같다, 같다 등을 표현함으로서 컨텍스트의 상태에 따른 이동을 결정한다.

제안된 방법은 이동정책 수립을 위하여 그림 4와 같은 사용자 인터페이스를 제공한다. 사용자 인터페이스는 해당 웹 서비스의 WSDL 문서, 제안된 스키마, 그리고 서비스 제공자의 이동정책 기술을 위하여 필요한 외부 스키마를 사용하여 정책수립에 필요한 최소한의 정보를 입력받아 이동정책을 수립한 후 WSDL 문서에 이를 귀속시킨다. 그림 4(a)는 WSDL 문서 및 외부 스키마 입력을 위한 인터페이스이며, 그림 4(b)는 그림 4(a)에서 설정한 컨텍스트를 기반으로 서비스 제공자로부터 필요한 정보를 입력받기 위한 사용자 인터페이스이다. 특히, 제안된 방법은 생성된 이동정책을 해당 WSDL 문서에 적절히 삽입함으로써 웹 서비스 표준에 기반한 웹 서비스 이동이 가능하게 한다.

그림 5는 제안된 인터페이스를 통하여 생성된 이동정책의 예로서, 제안된 스키마와 WAP 포럼의 UAProf를 기반으로 한다. 생성된 이동정책은 크게 이동시점의 결정을 위한 "When"과 타겟 호스트의 결정을 위한 "Where" 엘리먼트로 구성된다. 두 엘리먼트는 각각 서비스 제공자가 선택한 컨텍스트를 자식 엘리먼트로 갖는다. 예를 들어, 그림 5의 이동정책은 현재 서비스를 제공하고 있는 디바이스의 CPU 사용률이 70% 이상일 경우 이동하되, 타겟 호스트로



(a) WSDL 및 스키마 입력을 위한 인터페이스



(b) 이동정책 설정을 위한 인터페이스

그림 4. 사용자 인터페이스

는 자바의 사용이 가능하며, CPU 사용률이 50% 미만이고, 영어를 사용하며, HP 제품의 디바이스로 이동하라는 의미를 내포한다.

3.2 이동시점 및 타겟 호스트의 결정

특정 웹 서비스에 대한 이동정책이 수립되면 제안된 프레임워크의 이동 매니저는 수집된 컨텍스트 정보를 사용하여 이동시점을 결정한다. 즉, 이동 매니저는 생성된 이동정책으로부터 이동시점 결정을 위한 컨텍스트를 추출하고, 디바이스의 해당 컨텍스트를 모니터링 하여 이동정책을 만족하는 컨텍스트의 상태변화 시 이동을 결정하게 된다.

이동시점이 결정되면 제안된 방법은 이웃한 호스트들의 컨텍스트를 수집하고 타겟 호스트로서의 최소조건을 만족하는 후보 호스트를 추출한 후 타겟 호스트를 결정한다. 특히, 타겟 호스트의 결정은 서비스 제공자에 의해 정의된 컨텍스트의 우선순위에 기반하며, 그 값이 동일할 경우 다음 순위의 컨텍스트를 사용한다. 예를 들어, 표 1과 같이 이동정책을 만족하며, 각각 다른 컨텍스트 값을 가진 3개의 디바이스가 있다고 가정할 때, 제안된 방법은 우선순위가 가장 높은 컨텍스트인 CPU 사용률을 검사한 후 메모리양과 대역폭을 차례로 검사한다. 이때, 제안된 방법은 현재 컨텍스트의 값과 이동정책에 정의된 값의 차가 가장 큰 호스트를 타겟 호스트로 설정한다. 따라서 3개의 디바이스 중 우선순위가 가장 높은 CPU

```
<mig:Policy xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
            xmlns:mig="http://icl.yonsei.ac.kr/Policy/MigrationSchema.xsd"
            xmlns:uaf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20020710#">
  <mig:Migration_Policy>
    <mig:When>
      <wsp:ExactlyOne>
        <wsp:All>
          <mig:HardwareInfo>
            <uaf:cpu operation="morethan">70%</uaf:cpu>
          </mig:HardwareInfo>
          <mig:SoftwareInfo />
          <mig:NetworkInfo />
        </wsp:All>
      </wsp:ExactlyOne>
    </mig:When>
    <mig:Where>
      <wsp:ExactlyOne>
        <wsp:All>
          <mig:HardwareInfo>
            <uaf:vendor priority="4">HP</uaf:vendor>
            <uaf:cpu priority="2" operation="lessthan">50%</uaf:cpu>
          </mig:HardwareInfo>
          <mig:SoftwareInfo>
            <uaf:CcppAccept-Language priority="3">en</uaf:CcppAccept-Language>
            <uaf:JavaEnabled priority="1">yes</uaf:JavaEnabled>
          </mig:SoftwareInfo>
          <mig:NetworkInfo />
        </wsp:All>
      </wsp:ExactlyOne>
    </mig:Where>
  <Replication_Policy />
</mig:Policy>
```

그림 5. 생성된 이동정책의 예

표 1. 타겟 호스트 설정을 위한 컨텍스트의 예

	디바이스 A	디바이스 B	디바이스 C
CPU 사용률이 50%이하, 우선순위 = 1	40%	30%	30%
사용가능한 메모리 70% 이상, 우선순위 = 2	80%	70%	70%
대역폭 100MHz 이상, 우선순위 = 3	150MHz	130MHz	140MHz

사용률에 의해 'A 디바이스'가 제거되고, 마지막 대역폭에 의해 그 차가 더 큰 'C 디바이스'를 타겟 호스트로 결정하게 된다.

4. 실험 결과

본 절에서는 디바이스간의 웹 서비스 이동을 지원하는 모바일 웹 서비스 프레임웍에 제안된 이동정책을 적용함으로서 이동정책 수립기법의 유효성을 검증한다.

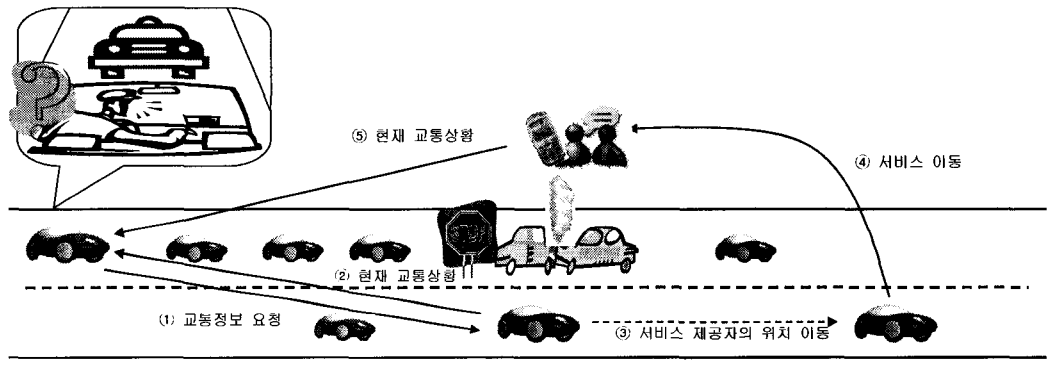
4.1 실험 환경 및 시나리오

본 논문에서는 이동정책에 기반한 웹 서비스 이동의 유효성 검증을 위하여 다음과 같은 시나리오에 제안된 방법을 적용하였다. 교통사고에 의해 지체되는 고속도로에서 서비스 사용자는 사고지점의 교통상황을 알기 원한다. 서비스 사용자는 교통정보를 제공하는 웹 서비스를 찾아 사고지점으로 이동시켜 서비스를 제공받는다. 그러나 서비스를 제공하던 디바이스를 가진 자동차가 사고지점을 벗어나 더 이상

서비스를 제공할 수 없을 때 해당 웹 서비스는 자동으로 사고지점에 있는 디바이스로 이동한다(그림 6 참조). 결과적으로 서비스 사용자는 웹 서비스의 이동과 상관없이 연속적인 서비스를 제공받는다. 제안된 방법은 표 2와 같이 1대의 소스 호스트와 2대의 타겟 후보 호스트로 구성된 802.11G 환경에서 이동정책의 유효성을 검증하였다.

4.2 실험 결과

시나리오를 위하여 제안된 방법은 서비스 제공자로부터 웹 서비스 이동에 필요한 정보를 입력받아 그림 7과 같이 이동정책을 생성한 후 WSDL 문서에 귀속시킨다. 한편, 소스 호스트의 컨텍스트 매니저는 이동정책의 "When" 엘리먼트를 기반으로 적절한 컨텍스트를 추출한 후 그 변화를 모니터링하고, 이동 매니저는 이를 기반으로 이동시점을 결정한다. 이동이 결정되면 이웃한 호스트로부터 "Where" 엘리먼트에 정의된 컨텍스트를 수집하여 후보 호스트를 추출하고 다시 컨텍스트의 우선순위에 기반하여 타겟 호스트를 결정한다. 그림 7은 교통정보 웹 서비스의



① 전방의 교통사고 때문에 정차해 있는 운전자가 사고지역을 지나는 차량에게 교통사고 처리상황에 대한 정보 서비스를 요청함.
② 교통사고 지점을 통과하는 차량의 모바일 디바이스에 탑재된 적절한 웹 서비스가 서비스 요청자에게 실시간으로 교통정보를 제공함.
③ 서비스 제공자가 사고 지역을 벗어나 더 이상 서비스를 제공할 수 없게 됨.
④ 해당 웹 서비스를 사고 지역에 위치한 다른 디바이스로 이동시킴.
⑤ 사고 지점에 위치한 또 다른 모바일 디바이스가 해당 웹 서비스를 설치 및 실행하여 사용자에게 계속해서 서비스를 제공함.

그림 6. 웹 서비스 이동이 요구되는 응용 시나리오의 예

표 2. 실험 환경

	소스 호스트	이웃 호스트 1	이웃 호스트 2 (타겟 호스트)
제조사	HP iPAQ hx4700	Acer n50	HP iPAQ hx4700
OS	Windows Mobile 2003		
Processor	Intel PXA270 624MHz	Intel PXA272 312MHz	Intel PXA270 624MHz
Memory	64Mb RAM 128Mb ROM	64Mb RAM 64Mb ROM	64Mb RAM 128Mb ROM
위치	사고지점	사고지점으로부터 반경 50M에 위치	사고지점으로부터 반경 50M에 위치
CPU 사용률	80%	70%	30%

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                                                    ....
                  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
                  xmlns:mig="http://icl.yonsei.ac.kr/Policy/MigrationSchema.xsd"
                  xmlns:uaf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20020710#"
                  xmlns:dis="http://InfoMapServices.cz"
                  targetNamespace="http://tempuri.org/"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

  <mig:Policy>
    <mig:Migration_Policy>
      <mig:When>
        <wsp:ExactlyOne>
          <wsp:All>
            <mig:HardwareInfo />
            <mig:SoftwareInfo />
            <mig:NetworkInfo />
            <dis:distance operation="morethan"> 100M </dis:distance>
          </wsp:All>
        </wsp:ExactlyOne>
      </mig:When>
      <mig:Where>
        <wsp:ExactlyOne>
          <wsp:All>
            <mig:HardwareInfo />
            <mig:SoftwareInfo />
            <mig:NetworkInfo />
            <dis:distance priority="1" operation="lessthan"> 100M </dis:distance>
            <mig:HardwareInfo>
              <uaf:cpu priority="2" operation="lessthan">50%</uaf:cpu>
            </mig:HardwareInfo>
            <mig:SoftwareInfo />
            <mig:NetworkInfo />
          </wsp:All>
        </wsp:ExactlyOne>
      </mig:Where>
    <Replication_Policy />
  </mig:Policy>

  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="AverageTrafficSpeed">
        ...
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="AverageTrafficSpeedSoapIn"> ... </wsdl:message>
  <wsdl:portType name="ServiceSoap"> ... </wsdl:portType>
  <wsdl:binding name="ServiceSoap" type="tns:ServiceSoap"> ... </wsdl:binding>
  <wsdl:service name="Service"> ... </wsdl:service>
</wsdl:definitions>
```

그림 7. 교통정보를 제공하는 웹 서비스의 이동정책

이동정책이 WSDL 문서에 귀속된 예로서, 디바이스가 사고 지점에서 100M 이상의 거리를 이동했을 때, 해당 웹 서비스는 사고 지점으로부터 반경 100M 이내에 위치하고, CPU 사용량이 50% 미만인 디바이스로 이동해야 함을 의미한다.

그림 8은 위 시나리오를 실제 디바이스에 적용한

(a) 소스 호스트 (b) 후보 호스트 (c) 후보 호스트(타겟 호스트)

그림 8. 이동정책에 기반한 웹 서비스 이동의 예

결과이다. 특히, 그림 8(a)와 그림 8(c)는 각각 웹 서비스가 이동한 전/후를 나타내는 소스 호스트와 타겟 호스트의 로그정보를 나타낸다. 그림에서와 같이 소스 호스트의 위치가 현재 위치에서 100M 이상 이동하면 웹 서비스의 이동이 결정된다. 이후 이동 매니저는 미리 정의된 이동정책을 기반으로 적절한 호스트 즉, 사고지점으로부터 100M 이내에 존재하고 CPU 사용률이 50% 미만인 호스트를 찾아 후보 호스트로 설정하고 이들에 대하여 다시 우선순위를 적용, 타겟 호스트를 결정한다. 실험결과 수립된 이동정책과 프레임웍에 의거 웹 서비스가 적절한 디바이스로 이동함으로서 사용자에게 지속적인 서비스를 제공함을 보였다.

5. 결 론

무선 네트워크와 모바일 디바이스의 발달은 웹 서비스와 결합되어 모바일 웹 서비스라는 새로운 형태의 페러다임을 형성하고 있다. 그러나 불안정한 연결성 및 빈번한 컨텍스트 변화 등은 모바일 환경에서 웹 서비스를 제공하는 데 많은 문제점을 야기한다. 본 논문에서는 모바일 환경에서 웹 서비스를 효과적으로 제공하기 위하여 모바일 디바이스간의 웹 서비스 이동(migration)을 지원하는 경량의 모바일 웹 서비스 프레임웍과 함께 언제, 어디로 이동할 것인지를 결정하기 위한 웹 서비스 이동정책 수립기법을 제안하였다.

본 논문에서는 웹 서비스의 이동을 위하여 이동정책 스키마를 정의하고, 이를 기반으로 서비스 제공자로부터 최소한의 정보를 제공받아 이동정책을 생성한다. 특히, 제안된 스키마는 서비스 제공자의 요구사항을 자유롭게 표현하기 위하여 확장 가능하도록 설계되었다. 이는 스키마의 확장을 통하여 서비스 제공자의 요구사항을 이동정책에 자유롭게 표현할 수 있게 하기 위함이다. 한편, 제안된 프레임웍은 생성된 이동정책을 기반으로 컨텍스트를 모니터링 하여 웹 서비스의 이동을 자동화한다. 실험 결과, 제안된 방법으로 기술된 이동정책과 모바일 웹 서비스 프레임웍을 기반으로 웹 서비스의 이동을 통한 연속적인 서비스를 제공이 가능함을 보였다.

향후 본 연구에서는 타겟 호스트의 자동설정을 위하여 컨텍스트의 비용을 계산하는 비용모델 개발과 함께 효과적인 SOAP 통신을 위한 SOAP 메시지 최적화에 대한 연구를 진행 할 예정이다.

참 고 문 헌

[1] World Wide Web Consortium, Web Services, http://www.w3c.org/2005/ws.

[2] S. N. Sirirama, M. Jarke, and W. Prinz,

"Mobile Web Services Provisioning," *Proc. Int'l Conf. Internet and Web Application and Services/Advanced Int'l Conf. Telecommunications* (AICT/ICIW 2006), pp. 120-126, 2006.

[3] D. Schall, M. Aiello, and S. Dustdar, "Web Services on Embedded Devices," *Web Information System*, Vol. 2, No. 1, pp. 1-6, 2006.

[4] S. N. Srirama, M. Jarke, and W. Prinz, "A Mediation Framework for Mobile Web Service Provisioning," *Proc. 10th IEEE Int'l Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*, pp. 14-17, 2006.

[5] W. Hao, T. Gao, I.-L. Yen, Y. Chen, and R. Paul, "An Infrastructure for Web Services Migration for Real-Time Applications," *Proc. 2th IEEE Int'l Symposium on Service- riented System Engineering (SOSE'06)*, pp. 41-48, 2006.

[6] D. Pratistha, A. Zaslavsky, S. Cuce, and M. Dick, "Performance Based Cost Models for Improving Web Service Efficiency Through Dynamic Relocation," *Proc. 6th Int'l Conf. Electronic Commerce and Web Technologies (EC-Web)*, Vol. 3590, pp. 248-287, 2005.

[7] D. Pratistha, A. Zaslavsky, S. Cuce, and M. Dick, "A Generic Cost Model and Infrastructure for Improving Web Service Efficiency through Dynamic Relocation," *Proc. Int'l Conf. Web Services (ICWS 2005)*, pp. 381-388, 2005.

[8] H. Hemmati, A. Ranjbar, M. Niamanesh, and R. Jalili, "A Model to Support Context-Aware Service Migration in Pervasive Computing Environments," *9th World Multi-Conf. Systemics, Cybernetics and Informatics*, 2005.

[9] O. Riva, T. Nadeem, C. Borcea, and L. Iftode, "Context-Aware Migratory Services in Ad Hoc Networks," *IEEE Transaction on Mobile Computing*, Vol. 6, No. 12, 2007.

[10] F. Corradini, R. Culmone, and M. R. D. Berardini, "Code Mobility for Pervasive Computing," *Proc. 13th IEEE Int'l Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)*, pp. 431-432, 2004.

[11] Web Service Policy Framework (WS-Policy), http://www-128.ibm.com/developerworks/library/specification/ws-polfram/.

김 연 석

2003년 명지대학교 전자정보통신공학부 졸업(학사)
2005년 연세대학교 컴퓨터과학과 졸업(석사)
2005년~현재 연세대학교 컴퓨터과학과 박사과정
관심분야 : 웹문서 분석, 정보 추출 및 통합, XML, 모바일 웹 서비스

이 경 호

1995년 연세대학교 전산과학과 졸업(학사)
1997년 연세대학교 컴퓨터과학과 졸업(석사)
2001년 연세대학교 컴퓨터과학과 졸업(박사)
2001년 National Institute of Standards and Technology(NIST) 객원연구원
2002년~현재 연세대학교 컴퓨터과학과 부교수
관심분야 : 멀티미디어 문서처리, XML, 웹 서비스