

유비쿼터스 서비스 개발을 위한 서비스의 효과적 융합기법 및 도구

안춘기*, 최재현*, 박제원* 이남용*

*송실대학교 컴퓨터학과

e-mail:neogray@gmail.com, uniker80@empal.com, jwpark5656@hotmail.com

nylee@ssu.ac.kr

A Effective Technique and Tool for Development of Converged Ubiquitous Service

Choon-ki Ahn*, Jae-hyun Choi*, Jae-Won Park*, Nam-Yong Lee*

*Dept of Computer Science and Engineering, Soong-sil University

요 약

유비쿼터스 컴퓨팅이란 언제 어디서나 사용자가 원하는 서비스를 제공받을 수 있는 컴퓨팅 환경을 말한다. 최근 네트워크와 컴퓨팅 기술이 발달하고 서비스에 대한 수요가 늘어남에 따라 유비쿼터스 컴퓨팅에 대한 관심도 증가하고 있으며 지속적인 연구가 이루어지고 있다. 그 동안 이루어진 많은 연구들을 통해 유비쿼터스 컴퓨팅 환경을 구축하기 위한 기반기술에 큰 발전이 있었다면, 최근에는 실제 유비쿼터스 환경에서 사용자에게 제공되는 서비스 콘텐츠의 효과적인 개발 및 확대에 대한 관심이 높아지고 있다. 모든 개체들이 네트워크로 연결되는 유비쿼터스 컴퓨팅 환경에서는 대부분의 사용자 콘텐츠가 네트워크를 통해 호출할 수 있는 원격 서비스의 형태로 제공된다. 또한 이러한 서비스들은 웹 서비스나 OpenAPI 등 다양한 형태로 존재하므로 이러한 원격 서비스들이 효과적으로 융합되고 재생산될 수 있어야 한다. 따라서 본 논문에서는 기존에 존재하는 형태의 원격 서비스들을 융합하여 새로운 서비스를 개발하기 위한 기법 및 도구를 제시한다. 이를 위해 본 논문에서는 유비쿼터스 컴퓨팅 환경에서 제공되는 서비스 형태를 U-Service로 정의하며, U-Service를 효과적으로 명세하고 융합할 수 있는 기법 및 도구를 제시한다.

1. 서론

유비쿼터스(Ubiquitous)는 라틴어 ‘ubique’를 어원으로 하여 “언제 어디서나 존재하는”이라는 뜻을 갖고 있는 말이며 이를 IT 분야에 접목하여 파생된 말인 유비쿼터스 컴퓨팅은 컴퓨터가 생활의 곳곳에 스며들어서 사용자가 의식하지 않고 언제 어디서나 컴퓨팅이 가능한 환경을 말한다.[1][2] 이러한 유비쿼터스 컴퓨팅 환경은 최근 근거리 통신, 네트워크, RFID, 단말기 기술 등 여러 가지 기술이 발달하고 SmartPhone 등의 개인용 단말의 보급이 확대되면서 점점 그 수요가 증가하고 있으며 그에 따라 발전속도도 가속화되고 있다. 이러한 경향 속에서 그 동안 여러 가지 연구들을 통해 많은 발전이 있었지만, 그간 연구되었던 기술들의 상당 부분이 유비쿼터스 컴퓨팅 환경을 구축하기 위한 네트워크 및 하드웨어의 기반기술에 중점을 두고 이루어졌다. 하지만 최근에는 컴퓨팅 환경에 있어 성능이나 효율성 뿐 아니라 활용성에도 무게를 두는 경향에 따라 유비쿼터스 컴퓨팅 분야에서도 사용자에게 제공되는 서비스 콘텐츠에 대한 중요성이 강조되고 있으며, 더불어 새로운 서비스를 어떻게 개발할 것인가에 대한 관심도 증가하고 있다.

유비쿼터스 컴퓨팅 환경에서 제공되는 사용자 콘텐츠들

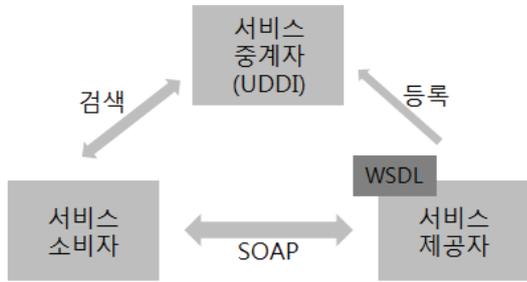
은 대부분 각 개체간의 네트워크 연결에 기반한 원격 서비스 형태로 제공되기 때문에 사용자 콘텐츠의 효과적인 개발 및 확대를 위해서는 이러한 서비스들이 효과적으로 융합되고 재생산될 수 있어야 한다. 이러한 서비스들은 한 가지 형태가 아니라 웹서비스, OpenAPI 등의 다양한 형태로 존재하기 때문에 이들을 효과적으로 명세하고 융합할 수 있는 기법 및 도구가 필요하다. 따라서 본 논문에서는 유비쿼터스 환경에서 사용자에게 제공될 수 있는 서비스를 U-Service로 정의하며, 다양한 서비스들을 융합하여 U-Service를 개발할 수 있는 도구를 제시한다.

2. 유비쿼터스 환경에서 사용 가능한 서비스

2.1 웹 서비스(Web Service)

W3C에서는 웹서비스를 “네트워크를 통해 기계와 기계간의 상호작용을 지원하기 위해 설계된 소프트웨어 시스템”으로 정의하고 있다.[3] 웹서비스는 분산 환경에서 서로 다른 플랫폼 혹은 프레임워크간의 상호작용을 위한 방법을 제공한다. 따라서 웹서비스는 프로그램 언어에 독립적이며, 클라이언트와 서버가 서로 다른 환경에서 구현되더라도 사용이 가능하다. 이를 위해 웹서비스는 명세언어인

WSDL(Web Service Description Language)을 이용하여 해당 웹서비스에 대한 명세를 제공하며, HTTP를 통해 전달되는 SOAP(Simple Object Access Protocol)을 이용하여 메시지를 주고받는다.



(그림 1) 웹 서비스 모델

2.2 OpenAPI

OpenAPI는 API 제공자가 자신들이 제공하는 서비스나 데이터를 외부에서도 이용할 수 있도록 개방해 놓은 프로그래밍 인터페이스이며, 사용자들은 이러한 OpenAPI를 통해 새로운 서비스를 만들어낼 수 있다. 하지만 OpenAPI는 보통 자신을 명세하기 위한 구조화된 문서를 갖지 않으며 REST(REpresentational State Transfer), SOAP 등 다양한 통신 프로토콜을 통해 서비스된다.

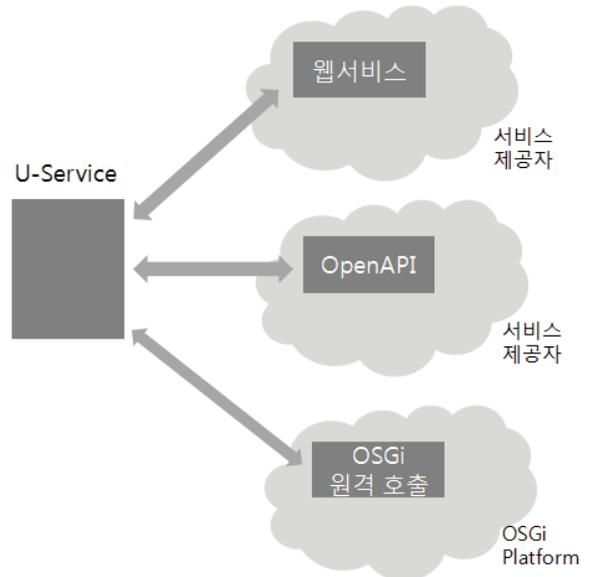
2.3 OSGi

OSGi는 서로 다른 환경의 다양한 장치들을 효과적으로 통합하고 미들웨어간의 효과적인 상호작용을 위해 표준화된 API를 제공하는 개방형 서비스 플랫폼 규격이다.[4] 현재까지 제공되어 있는 표준 스펙에서는 OSGi 프레임워크 내부에 정의된 서비스들간의 동적 재구성이나 협업만을 지원할 뿐 프레임워크 외부의 서비스들과 연계는 지원하지 않고 있다. 하지만 이를 가능하게 하기 위한 방법들이 제시되고 있으며,[5] 이러한 연구 결과를 통해 OSGi Framework상의 장치들에 대한 서비스를 원격으로 이용할 수 있다.

3. U-Service

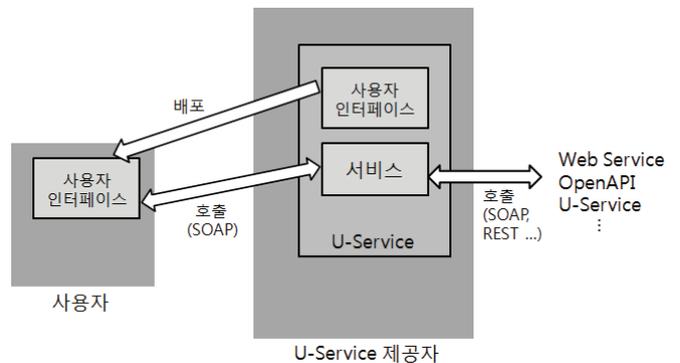
본 논문에서 제시하는 U-Service는 유비쿼터스 컴퓨팅 환경에서 사용자에게 제공되기 위해 만들어진 서비스로서, 웹서비스, OpenAPI, OSGi 등 원격 호출이 가능한 형태의 기존의 서비스들을 융합하여 만들어진다.

또한 U-Service는 웹 서비스의 개념을 기본으로 하여 확장한 형태이다. U-Service는 웹 서비스 형태로 되어있어 실제로 서비스 프로세스를 처리하는 부분과 해당 서비스를 이용하기 위한 사용자 인터페이스로 구성되며 또한 자신이 제공하는 서비스에 대한 명세도 포함한다. 이는 웹 서비스의 WSDL을 확장한 형태로서 WSDL에 기술되는



(그림 2) 서비스 융합

내용 이외에도 U-Service에 대한 설명과 배포를 위한 정보 등이 기술된다. 개발이 완료된 U-Service는 패키징된 형태로 U-Service 제공자를 통해 배포된다. 이렇게 U-Service가 사용자에게 배포될 때, 사용자는 서비스 전체 코드를 제공받는 것이 아니라 해당 서비스의 인터페이스만을 제공받는다. 그리고 이 인터페이스는 SOAP 메시지를 이용하여 서비스를 호출하고, 서비스는 자신이 참조한 웹서비스나 OpenAPI를 호출한다.



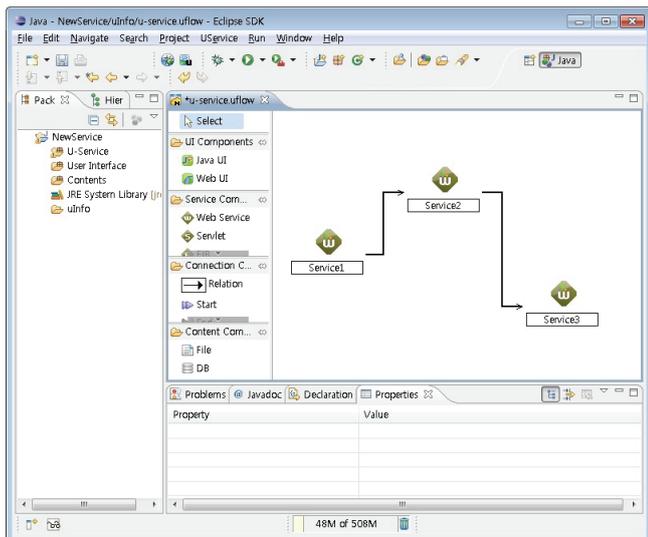
(그림 3) U-Service의 구성과 배포

이러한 구조는 유비쿼터스 컴퓨팅 환경의 특징과 관련이 있다. 유비쿼터스 컴퓨팅 환경을 특성상 대부분의 서비스들이 원격 호출을 통해 이루어지게 되는데, 이 때 참조한 모든 호출이 유효해야만 U-Service가 정상적으로 동작할 수 있다. 하지만 유비쿼터스 환경에서는 새로운 서비스가 생성되거나, 기존에 정상적으로 제공되던 서비스가 더 이상 유효하지 않은 상태가 되는 변화가 일어날 수 있다. 이런 경우 U-Service가 정상적으로 동작하지 않을 수 있는데, 특히 호출된 서비스의 수가 많을수록 문제가 발생할 확률은 높아진다. 이렇게 문제가 발생하면 U-Service의

수정이 필요한데, 만약 U-Service 전체를 사용자에게 배포하는 구조라면 수정을 위해 많은 비용이 발생하게 된다. 따라서 인터페이스와 서비스를 분리하는 형태를 취함으로써 서비스의 수정을 용이하게 할 수 있으며 사용자 입장에서 보다 높은 가용성을 기대할 수 있다. 또한, 기존에 개발된 U-Service를 활용하여 새로운 U-Service를 만들거나, 개발된 하나의 U-Service를 다양한 환경에서 사용하기 위해서도 (그림 3)과 같이 인터페이스와 서비스를 분리하는 구조가 유리하다.

4. 매쉬업 개념을 이용한 U-Service 개발 도구

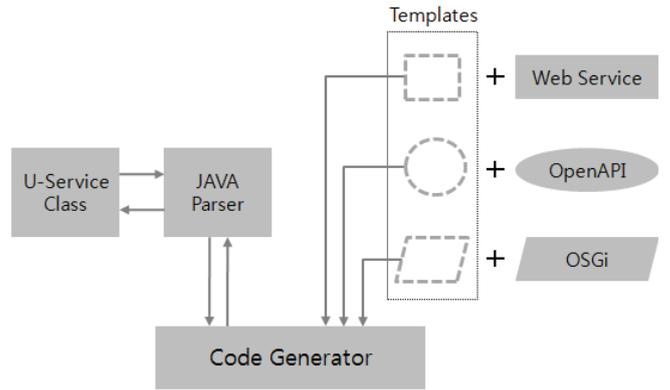
(그림 4)는 매쉬업 개념을 이용하여 본 논문에서 제시한 U-Service를 개발할 수 있는 개발 도구를 보이고 있다. 개발자는 사용하고자 하는 서비스 컴포넌트들을 편집 화면에 올려놓고 해당 컴포넌트를 실제 사용 가능한 서비스와 1:1로 매핑시킨다. 이후 각각의 서비스 컴포넌트들을 선으로 연결함으로써 순차적인 하나의 서비스 흐름을 만들게 되는데 이 과정에서 연결하는 서비스들간의 입력과 출력이 상이하므로 개발자는 각각의 연결들마다 팝업창을 통해 컴포넌트간의 입력과 출력을 대응시킨다. 이 때 개발 도구는 개발자가 설정하는 내용과 미리 정의해 놓은 템플릿을 결합하여 실행 가능한 코드를 만들고, 이를 통해 새로운 서비스를 생성한다. 따라서 U-Service에는 웹 서비스를 호출하는 SOAP 메시지를 생성하기 위한 코드, REST 형식을 지원하는 OpenAPI를 활용하기 위한 코드, OSGi 프레임워크상의 원격 호출을 위한 코드 등이 생성된다. 또한 개발자는 U-Service의 코드를 직접 편집함으로써 팝업창을 통해 설정하지 못한 부분을 편집할 수 있다.



(그림 4) U-Service 개발도구

5. 결론

본 논문을 통해 유비쿼터스 컴퓨팅 환경에서 사용자에게 제공 가능한 서비스 형태인 U-Service와, 기존의 서비



(그림 5) U-Service의 코드 생성

스들을 융합하여 U-Service를 개발하기 위한 도구를 제시하였다. 유비쿼터스 컴퓨팅이 실현 단계에 접어들고, 그에 따른 수요가 증가하고 있는 상황에서 기존의 서비스 콘텐츠들을 활용해서 새로운 서비스 콘텐츠를 개발하고 재생산하는 일은 꼭 필요한 일이며, 매쉬업을 활용한 개발도구는 서비스 개발자로 하여금 개발 작업을 보다 명시적이고 직관적으로 할 수 있게 해 준다. 이는 생산성 향상과도 관련된 문제이다.

향후에는 개발도구에 조건별 분기, 필터 등 개발도구 레벨에서 적용 가능한 컴포넌트를 추가하여 개발도구의 자동화 범위를 넓히기 위한 연구가 진행될 것이며, 또한 현재 각 제공자마다 각기 다른 형태로 제공되고 있는 OpenAPI를 표준화할 수 있는 명세에 대한 연구가 진행될 것이다.

참고문헌

- [1] Mark Weiser, "The Computer for the 21st century", 1991
- [2] 한국정보통신기술협회(TTA), 정보통신용어사전, <http://www.tta.or.kr>
- [3] W3C Web Services WG, "Web Services Architecture," W3C Working Group Note 11 February 2004.
- [4] The OSGi Alliance, "OSGi Service Platform Core Specification Release4, June, 2009
- [5] Choi Jae-hyun, Park Jae-won, Lee Nam-yong, "Java Remote Service Invocation for OSGi Framework to Enhance Inter-Operations of Services on OSGi", 2009