동적 서비스 조합을 위한 서비스 컴포넌트 아키텍처의 확장

황윤영, 이규철
충남대학교 컴퓨터공학과

A Extension of Service Component Architecture for Dynamic Service Composition

Yun-Young Hwang, Kyu-Chul Lee

요 약

사용자의 서비스 요구시 사용자의 요구사항에 꼭 맞는 단일 서비스가 존재하지 않는다면, 여러 서비스를 조합할 필요성이 있다. 서비스 조합에 대한 연구 중 서비스 지향 아키텍처 기반의 서비스 조합 기술인 서비스 컴포넌트 아키텍처(Service Component Architecture) 표준화가 진행되고 있다. 이 표준은 시스템 설계 단계에서의 서비스 조합을 목표로 하고 있어, 동적 서비스 조합이 요구되는 유비쿼터스 환경에서는 활용되기 어렵다. 따라서, 본 논문에서는 서비스 컴포넌트 아키텍처를 확장하여 유비쿼터스 환경에서 동적 서비스 조합이 가능한 방법을 제시한다.

Abstract

This paper is to provide the method of dynamic service composition in ubiquitous environment. This method is extended of Service Component Architecture, which is a standard about service composition. This standard only supports service composition in system design time. It has problem we cannot it ubiquitous environment, requests dynamic service composition. To solve this problem, we suggest dynamic service composition method based on SCA.

1. 서론

사용자의 서비스 요구시 사용자의 요구사항에 꼭 맞는 단일 서비스가 존재하지 않는다면, 여러 서비스를 조합할 필요성이 있다.

웹서비스 등장 이후, 서비스 발견과 더불어 서비스 조합에 대한 연구가 활발히 진행되어 왔다. 그 중 최근에 서비스 지향 아키텍처 기반의 서비스 조합 관련하여 서비스 컴포넌트 아키텍처(Service Component Architecture) 표준화가 진행되고 있다. 그러나 서비스 컴포넌트 아키텍처는 서비스 개발자가 미리 정의해 놓은 서비스 조합 규칙에 따라 사용자가 서비스를 조합하도록 하고 있다.

유비쿼터스란 어떤 사용자가 언제 어디서든지 어떤 네트워크에 있는 어떤 디바이스의 어떤 서비스도 사용가능하도록 해야 한다. 이를 위해서는 동적 서비스 발견 및 동적 서비스 조합이 필요하다.

동적 서비스 조합이란 사용자의 요구상황에 맞도록 그 상황에서 사용 가능한 여러 서비스를 조합하는 것을 뜻한다. 예를 들어, 사용자가 "쿵푸팬더"를 PDA로 보고 있다가 차에 탑승한 뒤 그 서비스를 지속적으로 제공받길 원하는 상황이라고 가정한다. 이 때, 사용자는 PDA로 영화를 계속 상영하기 어렵다. 이 상황에서 동적 서비스 조합이 가능하다면 사용자는 네비게이션과 차량 스피커를 통해 지속적으로 영화를 관람할 수 있다.

그러나 정적 서비스 조합을 지원하는 서비스 컴포넌트 아키텍처는 유비쿼터스 환경을 지원하지 못한다. 따라서 본 논문에서는 서비스 컴포넌트 아키텍처의 단점을 보완하여 유비쿼터스 환경에 접목시킴으로써 동적 서비스 조합이 가능한 방법을 제안한다.

본 논문은 다음과 같은 구성을 가진다. 첫 번째로 서비스 컴포넌트 아키텍처를 소개하고, 두 번째로 동적 서비스 조합 방법을 제안한다. 마지막으로 결론 및 향후 연구에 대해 기술한다.

2. 서비스 컴포넌트 아키텍처

서비스 컴포넌트 아키텍처(Service Component Architecture)[1]는 서비스 지향 아키텍처 기반의 서비스 컴포넌트 조합(Composite)에 관한 표준이다.

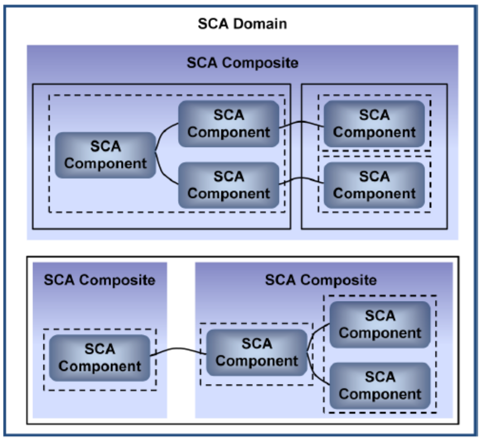

그림 1. 서비스 컴포넌트 아키텍처

서비스 컴포넌트 아키텍처는 [그림 1]과 같이 조합에 필요한 모든 요소를 컴포넌트로 정의한다.

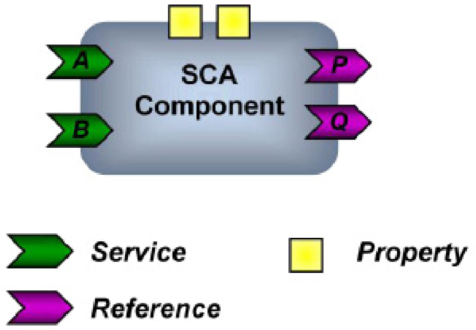

그림 2. 서비스 컴포넌트 아키텍처의 컴포넌트

여러 컴포넌트로 조합된 서비스 컴포넌트 아키텍처 조합(SCA Composite)는 또다시 다른 조합의 컴포넌트로 사용될 수 있다.

서비스 컴포넌트 아키텍처에서 정의하는 컴포넌트는 [그림 2]과 같은 모습을 가진다.

컴포넌트에 접근하기 위해 필요한 서비스(Service)와 다른 컴포넌트와의 연결을 위한 참조(Reference), 컴포넌트의 특성을 기술하기 위한 프로퍼티(Property)가 있다. 서비스는 컴포넌트에 접근하는 클라이언트가 어떤 환경을 요구하냐에 따라 Java 혹은 WSDL(Web Services Description Language) 등으로 다양하게 기술 될 수 있다. 참조는 [그림 3]에서 보여지듯이 다른 컴포넌트의 서비스와 연결 된다. 참조는 자신과 연결될 컴포넌트가 외부에 있든 내부에 있든 상관하지 않고 참조될 컴포넌트의 논리적 위치를 지정한다. 이처럼 참조와 서비스간의 연결을 포함하여 서비스와 서비스 간의 연결, 참조와 참조간의 연결을 모두 와이어링(Wiring)이라고 한다.

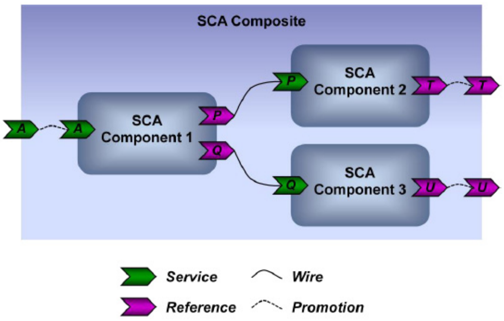

그림 3. 컴포넌트간의 연결

서비스 컴포넌트 아키텍처에서 제공하는 와이어링에는 다음과 같이 3가지가 있다.

- Component.Reference->Component.Service
- Composite.Reference->Component.Reference
- Composite.Service->Component.Service

Component.Reference와 Component.Service 간의 와이어링은 [그림 3]의 P(참조)->P(서비스)와 Q(참조)->Q(서비스)와 같은 연결을 의미한다. 이 와이어링의 경우에는 3가지 상황이 발생할 수 있는데, [그림 3]의 SCA Component1과 SCA Component2 모두가 조합된 컴포넌트가 아닐 경우와, SCA Component1 또는 SCA Component2가 조합된 컴포넌트 일 경우이다. 이 중, 서비스 컴포넌트 아키텍처는 뒤의 2가지 경우를 지원하지 못한다.

[그림 3]의 T(참조)->T(참조)와 U(참조)->U(참조)는Composite.Reference와 Component.Reference 간의 와이어링을 보여주고 있다. Composite.Service와 Component.Service 간의 연결은 [그림 3]의 A(서비스)->A(서비스)와 같은 경우이다. 이 두 가지 경우는 컴포넌트의 조합을 다시 하나의 컴포넌트로 정의함으로써, 다른 조합에 활용될 수 있게 하는데 필요하다.

이와 같은 구성을 가지는 서비스 컴포넌트 아키텍처는 서비스 지향 아키텍처를 기반으로 여러 다양한 플랫폼 및 구동 환경의 서비스를 조합하는데 효과적인 방안을 제공한다. 또한,

기업간의 서로 다른 서비스를 조합하여 협업이 가능케 하는 상황에서는 효율적이다. 그러나 서비스 조합 규칙을 사용자가 아닌 개발자가 미리 정의해 놓음으로서, 런타임시에 새로운 서비스의 조합이 불가능하다. 이는 동적인 서비스 조합을 요구하는 유비쿼터스 환경에는 적합하지 못하다.

따라서, 본 논문에서는 서비스 컴포넌트 아키텍처를 확장하여 런타임 시에도 새로운 서비스 조합이 가능하도록 하는 방안을 제안한다.

3. 동적 서비스 조합

2장에서 살펴보았듯이 서비스 컴포넌트 아키텍처에서는 3가지의 와이어링을 제공한다. 이 중, 컴포넌트간의 연결을 담당하는 Component.Reference->Component.Service 와이어링 중 조합된 컴포넌트간의 연결은 지원하지 못하고 있었다. 동적인 서비스 조합을 지원하기 위해서는 이 와이어링을 제공해야 한다. 이를 위해, 본 논문에서는 대체(Replace)라는 방법을 활용하기로 하였다.

대체란 기존의 컴포넌트를 새로운 컴포넌트로 대체하는 방법이다. 이 때, 기존 컴포넌트의 서비스, 프로퍼티, 참조는 변경하지 않고 내부 구성만 변경하여 대체하는 방법이다.

본 논문에서 제안하는 방법을 쉽게 설명하기 위해, 다음과 같은 시나리오를 가정한다.

시나리오: 사용자 A는 아침 6시에 *시계 알람이 울리면* 부엌에서는 *커피포트와 토스트기가 동작*되고, *사용자가 시계 알람을 멈추면*, *TV가 켜지고 사용자가 지정된 채널이 보여*지길 원한다.

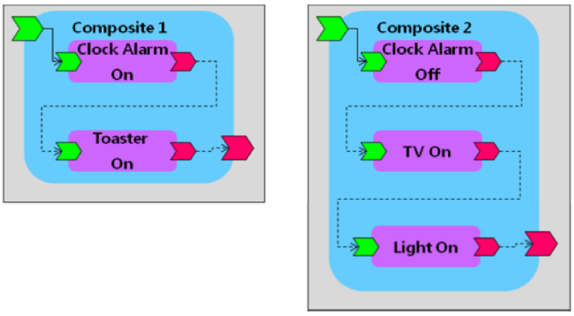

그림 4. 미리 조합된 서비스

그러나 서비스 컴포넌트 아키텍처로 이미 정의된 서비스 조합은 [그림 4]와 같다. 시계 알람이 울리면 토스터기가 동작되는 조합된 서비스 하나와 시계 알람이 꺼지면 TV가 켜지고 전등이 켜지는 또 다른 조합된 서비스가 존재한다. 이 때, 사용자 A의 요구를 만족시켜 주려면 다음과 같은 오퍼레이션이 필요하다.

1. Composite1에 시계 알람 서비스와 토스트기계 서비스 사이에 커피포트 **서비스 삽입**
2. Composite2에 TV 서비스 다음에 전등 **서비스 삭제**
3. Composite1과 Composite2의 **조합**

이 오페레이션을 수행하기 위해 본 논문에서는 [그림 4]를 [그림 5]와 같이 각각의 서비스를 조합된 서비스로 정의한다.

이렇게 서비스 컴포넌트 아키텍처를 이용하여 서비스를 정의한 시나리오를 수행하기 위해 필요한 오퍼레이션을 런타임시에 다음과 같이 수행할 수 있다. 우선 첫번째 오퍼레이션을 수행하기 위해, Composite 1-1를 복사한 뒤, 시계 서비스 뒤에 커피 서비스를 연결하고, 기존의 Composite 1-1을 대체한다.

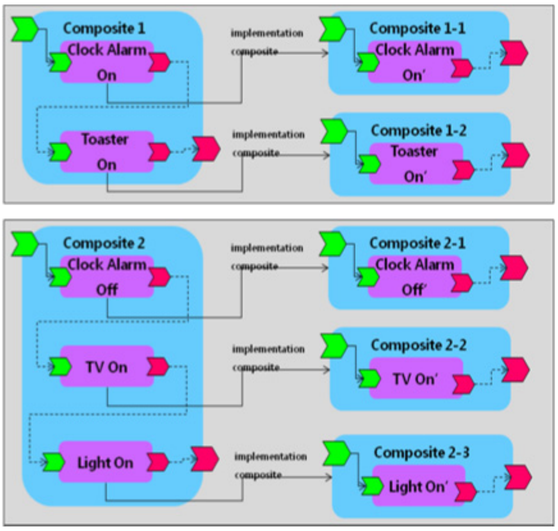

그림 5. 동적 서비스 조합을 위한 서비스 정의

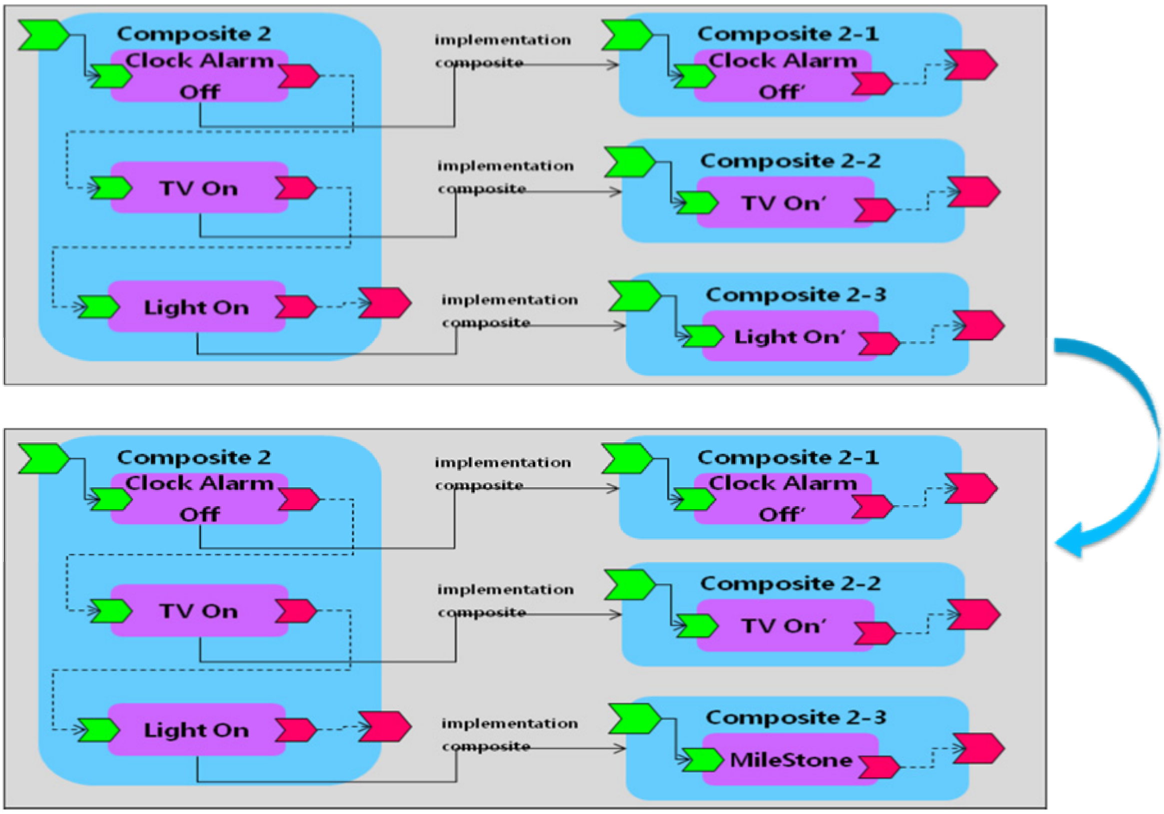

그림 6. 서비스 삭제 방법

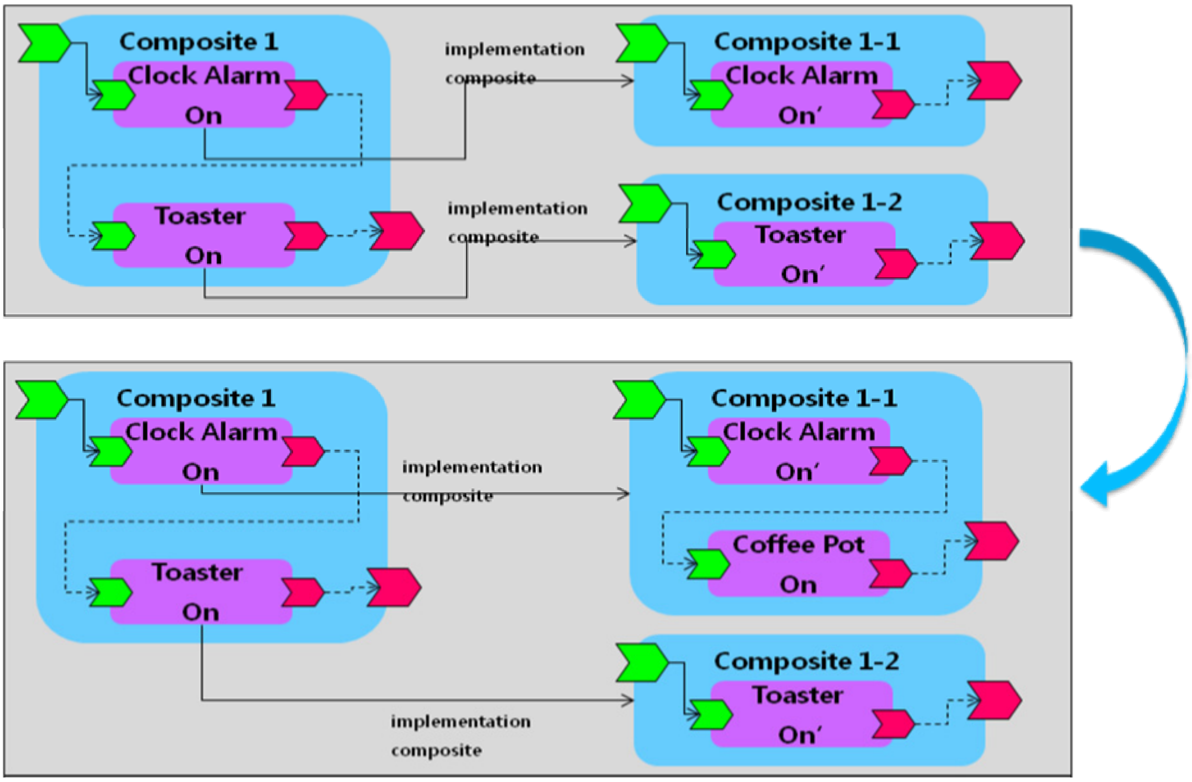

그림 7. 서비스 삽입 방법

서비스를 삭제하는 방법은 기존의 서비스 조합의 연결 사이에 영향을 주지 않도록 해야 하기 때문에, 마일스톤 서비스를 생성하여 대체하도록 한다. 마일스톤 서비스란 기존의 연결을 유지하면서 서비스의 입력을 단순히 출력하는 서비스이다([**오류! 참조 원본을 찾을 수 없습니다.**]). 세 번째 오퍼레이션은 첫 번째 오퍼레이션 방법을 그대로 적용하면 된다.

4. 결론 및 향후연구

지금까지 서비스 컴포넌트 아키텍처를 이용한 동적 서비스 조합 방법을 제안하였다. 서비스 컴포넌트 아키텍처는 개발자가 서비스 조합 규칙을 정의함으로써 런타임시에 사용자의 요구에 맞는 서비스 조합을 제공해 줄 수 없다는 문제점을 가졌다.

본 논문에서는 이러한 문제점을 해결하고자, 대체라는 방법을 정의하고, 기존 서비스 조합에 새로운 서비스를 삽입, 삭제하는 방법을 제안하였다.

그러나 아직까지 본 논문의 방법은 여러 조합된 서비스들의 새로운 조합 및 삭제, 수정 등은 충분히 반영되지 못함으로써, 앞으로 이에 대한 연구가 지속적으로 진행되어야 한다.

ACKNOWLEDGEMENT

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업(IITA-2005-C01090-0502-0016)의 연구결과로 수행되었음

참고문헌

[1] David Chappell, “Introducing SCA”, CHAPPELL & ASSOCIATES, 2007

[2] OSOA (Open Service Oriented Architecture), http://www.osoa.org

[3] Sungrim Kim, “Attention-Based Information Composition for Multicontext-Aware Recommendation in Ubiquitous Computing”, 2006

[4] Takaaki, “A Support System for Designing Ubiquitous Service Composition Scenarios”, ICC 2007

[5] Agnel Jimenez Molina, “A template-Based Mechanism for Dynamic Service Composition Based on Context Prediction in Ubicomp Applications”, 2007

저자소개

황윤영(e-mail: yyhwang@cnu.ac.kr)은 2004년 충남대학교 컴퓨터공학과 석사를 취득하고, 2004년부터 현재까지 충남대학교 대학원 컴퓨터공학과 박사과정에 재학 중이다. 관심분야는 서비스 지향 아키텍처 및 유비쿼터스 웹 서비스이다.

이규철(e-mail: kclee@cnu.ac.kr)은 1984년 서울대학교 컴퓨터공학과 학사를 취득하고, 1986년 서울대학교 컴퓨터공학과 석사를 취득하였으며, 1990년 서울대학교 컴퓨터공학과 박사를 취득하였다. 1989년부터 현재까지 충남대학교 교수로 재직중이다. 관심분야는 XML, 웹 서비스, 시맨틱 웹 서비스, 유비쿼터스 웹 서비스이다.