

융합 서비스의 진화를 위한 프로파일 기반 동적 서비스 바인딩 기술

Profile-Based Dynamic Service Binding for Evolution of Converged Services

김길환(Kilhwan Kim)*, 금창섭(Changsup Keum)**, 배현주(Hyun Joo Bae)***

초 록

융합 서비스를 빠른 시간 안에 시장에 제공하기 위해서는 외부 서비스와 자신의 핵심 역량을 융합하는 능력이 필요하며, 정보시스템 아키텍처는 이러한 서비스 융합을 효과적으로 지원해야 한다. 서비스 지향 아키텍처는 이러한 서비스 융합을 성공적으로 지원할 수 있는 정보시스템 아키텍처이다. 그러나 기존의 서비스 지향 아키텍처를 이용한 융합 서비스의 구현은 개별 서비스 진화가 융합 서비스의 진화로 반영되지 않는 단점이 있다. 본 논문에서는 서비스 지향 아키텍처의 맥락하에서 개별 서비스의 진화에 따른 융합 서비스의 자체적 진화를 지원하기 위한 프로파일 기반 동적 서비스 바인딩 기법의 도입을 제안하였다. 이를 위해 프로파일 기반 동적 서비스 바인딩 기술의 필요성을 제시하기 위해 스마트 운동 기구와 관련된 융합 서비스 시나리오를 제시하고 해당 기술이 구현해야 할 요구 사항을 도출하였다. 그리고 도출된 요구 사항을 만족시킬 수 있는 프로파일 기반 동적 서비스 바인딩 기술의 개념 및 구조를 제안하였다. 마지막으로 OSGi, Apache Camel, Apache CXF 등을 이용하여 프로파일 기반 동적 서비스 바인딩 기술의 프로토타입을 구현하여 제안된 기술에 대한 간략한 평가를 수행하였다.

ABSTRACT

To reduce time to market for converged services, the service provider needs to combine external services and internal processes together, and IT architecture for converged services has to support it. Service Oriented Architecture (SOA) enables the development of a converged service by combining external services and internal processes effectively. However, it is difficult for the existing SOA methods to support the evolution of a converged service, even though the service which consists of the converged service changes over time. In this paper, we argue that a method for profile-based dynamic service binding is required to support evolution of converged services. In order to identify requirements for the proposed method, a business scenario with a smart athletic equipment is given. We then present the concept and architecture of the method for profile-based dynamic service binding to meet the identified requirements. We also demonstrate a prototype implementation for evaluating the proposed method.

키워드 : 융합 서비스, 동적 서비스 바인딩, 서비스 프로파일, 서비스 지향 아키텍처
Converged Services, Dynamic Service Binding, Service Profile, SOA

* Corresponding Author, Department of Management Engineering, Sangmyung University
(E-mail : khkim@smu.ac.kr)

** Communications Internet Research Laboratory, ETRI

*** Communications Internet Research Laboratory, ETRI

2013년 01월 11일 접수, 2013년 03월 12일 심사완료 후 2013년 03월 29일 게재확정.

1. 서 론

최근 모든 산업에서 융합(convergence)을 통한 새로운 사업 기회를 창출하려는 노력이 가속화되고 있고, 서비스 산업 분야도 예외가 아니어서 기존의 서비스를 융합하거나 IT 서비스와 기존 서비스를 융합하려는 시도가 지속되고 있다[11, 13, 14]. 예로 통신 서비스 분야에서 초고속 인터넷, 전화, 방송 서비스가 통합되어 가고 있으며, 통신 서비스와 다양한 IT 서비스 간의 융합도 이루어지기도 한다[2, 6].

융합 서비스 사업자에게는 고객의 본원적 욕구를 파악하여 기존 서비스와 차별화되는 새로운 융합 서비스를 디자인하는 능력과 함께, 새로운 융합 서비스를 빠른 시간 내에 시장에 제공할 수 있는 융합 서비스 구현 능력이 필요하다. 따라서 융합 서비스 제공자는 모든 것을 자신이 구현하기 보다는 외부 서비스 제공자의 개별 서비스를 자신의 핵심 역량과 융합하여 새로운 융합 서비스로 신속히 제공하는 것이 이로우울 수 있다.

외부 서비스 제공자와 서비스를 융합하기 위해서 상호 간에 개별 서비스를 개방화하고 이를 융합할 수 있는 정보시스템 구조가 뒷받침되어야 한다. 이러한 서비스 융합을 뒷받침하는 대표적인 정보시스템 구조가 웹 서비스 기반의 서비스 지향 구조(Service Oriented Architecture; 이하 SOA)라 할 수 있다. 이러한 SOA 구조를 융합 서비스 비즈니스에 활용한 예로서, 통신 서비스에서 IT 서비스와 융합을 통해 새로운 서비스를 제공하기 위해, 웹 서비스 기반의 SOA 구조인 SDP(Service Delivery Platform)를 활용하는 경우를 들 수 있다. 이러한 이유로 융합 서비스를 위한 웹 서비스 기반

SOA에 대한 다양한 연구들이 많이 수행되었다[6, 18, 23].

그런데 융합할 수 있는 개별 서비스 수의 지속적인 증가와 개별 서비스 기능의 지속적인 업데이트는 융합 서비스의 융합 상태가 고정된 것이 아니라 지속적으로 진화해 나갈 필요를 제기하고 있다. 따라서 융합 서비스를 위한 정보시스템 구조도 서비스 융합 형상의 동적인 진화를 지원할 수 있는 구조를 갖추어야 한다. 이러한 서비스 융합의 동적 진화 구조를 구현하기 위한 유력한 기술이 웹 서비스 간의 동적 바인딩 기법이다[4, 7, 9, 16, 17, 20].

융합 서비스에서의 동적 바인딩이란 융합되는 개별 서비스가 하나로 정적으로 고정되어 있는 것이 아니라 실행 시의 최적 상황에 따라 적절한 개별 서비스가 동적으로 결정되는 것을 의미한다. 예를 들어 통신 서비스와 위치/지도 서비스를 이용한 융합 서비스를 제공하고자 할 때 위치/지도 서비스를 위해 구글, 네이버, 다음 지도 서비스 등을 실행 시점(runtime)의 상황에 따라 동적으로 선택할 수 있다.

동적 바인딩 기술은, 동적 바인딩 대상이 되는 다수의 개별 서비스가 융합 서비스 디자인 시점에 이미 결정되어 있고 실행 시점에 이 중 적절한 서비스가 선택되는 구조가 있고[17, 20], 동적 바인딩의 대상이 되는 개별 서비스가 실행 시점에 동적으로 추가 및 삭제가 이루어 질 수 있는 구조도 있다[4, 16].

그런데 이 두 동적 바인딩 기술은 모두 개별 서비스와 융합 서비스 간의 인터페이스가 고정적으로 정해져 있고 융합 인터페이스를 통해 융합되는 개별 서비스의 추가, 삭제, 바인딩 만이 동적으로 이루어진다. 따라서 개별 서비스가 진화되더라도 이러한 진화가 융합

인터페이스가 재개발되기 전에는 융합 서비스에 반영되기 어렵다. 또한 대부분의 경우 개별 서비스는 융합 서비스에 전적으로 종속적이지 않으므로 융합 인터페이스에만 순응되는 기능만을 가지는 것이 아니라 특화된 기능들을 가지고 있다. 그러나 고정적인 융합 인터페이스 구조에서는 융합 방식이 융합 대상이 되는 여러 서비스의 공통 기능으로 제한될 수밖에 없다. 따라서 동일한 융합 인터페이스를 공유하는 개별 서비스는 기능적으로 거의 동일한 서비스를 제공하나 비기능적인 요소(품질 및 가격 요소 등)에서만 서로 다른 특성을 융합 서비스에 제공하게 된다. 이 결과로 융합 서비스의 이용자는 개별 서비스만의 특화된 기능은 이용할 수 없고 공통된 최소 기능만을 사용하게 된다.

따라서 융합 서비스를 위한 정보시스템 구조가 개별 서비스가 진화함에 따라 융합 서비스의 동적인 진화를 지원하기 위해서는 핵심 기능은 동일하지만 개별 서비스만의 서로 다른 특화된 기능이 융합 서비스에 반영될 수 있는 정보시스템 구조를 가져야 한다. 그리고 이러한 융합 서비스 진화 구조가 새로운 개별 서비스가 추가되거나, 기존의 서비스가 제거 및 업그레이드 됨에 따라 융합 서비스가 중단 없이 진화할 수 있어야만 융합 서비스의 동적 진화를 보장할 수 있을 것이다.

본 논문에서는 융합 서비스 사용자와 개별 서비스의 프로파일 정보를 이용한 동적 바인딩 기술을 제안한다. 이를 통해 융합 서비스가 개별 서비스의 핵심 기능 이외의 특화 기능을 반영하여 실행 시점에 동적 바인딩 할 수 있도록 한다. 이를 통해 궁극적으로는 융합 서비스 자체가 중단 없이 지속적으로 진화할 수 있는

정보시스템 구조를 제안한다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 관련된 문헌을 고찰하고, 제 3장에서는 본 연구의 동기가 되는 융합 비즈니스 시나리오가 제시된다. 제 4장에서는 프로파일 기반 융합 서비스의 동적 바인딩 기술의 개념과 구조가 제안된다. 제 5장에서는 프로파일 기반 융합 서비스의 동적 바인딩 기술에 대한 프로토타입 구현을 소개한다. 마지막으로 제 6장에서는 연구에 대한 결론이 언급된다.

2. 문헌 연구

융합 서비스를 위한 기존의 정보시스템 아키텍처에 대한 연구와 프로파일 기반 동적 바인딩 기술과 관련된 관련 연구를 소개한다.

융합 서비스를 위한 정보시스템 아키텍처인 웹 서비스 기반 SOA[8]는 서비스의 융합을 지원하기 위한 최적의 정보시스템 구조로서 다음과 같은 장점을 가진다.

첫째, 융합 서비스는 기존의 제품/서비스와의 연결뿐 아니라 고객의 컴퓨터, 스마트 기기 등 다양한 단말과의 연결을 지원해야 할 필요성이 증대될 것이다. 그리고 연결해야 할 개체의 수는 지속적으로 증가해 나갈 것이다. 웹 서비스는 HTTP 기반의 분산 컴퓨팅 기술로서 Google 등 전 세계적인 서비스를 하는 많은 서비스 사업자가 채택해온 서비스 플랫폼으로서 보편성, 개방성, 확장성에 탁월한 장점을 보여주고 있다. 따라서 웹 서비스를 기반으로 한 정보시스템 구조는 융합 서비스를 위한 최적의 서비스 인터페이스를 구성하게 해 줄 것이다.

둘째, 융합 서비스는 기존의 제품/서비스와 외부의 IT 서비스 등의 다양한 요소를 혁신적으로 조합하여 새로운 서비스 프로세스를 창출해 내야 한다. 그리고 이러한 서비스 프로세스는 고객의 욕구와 서비스 환경의 변화에 따라 끊임없이 재구조화 되어야 한다. SOA는 정보시스템의 구조를 재사용 가능한 본질적인 서비스로 분할한 뒤 비즈니스 프로세스의 변화에 따라 이를 신속하게 재구조화할 수 있도록 비즈니스 프로세스 계층과 서비스 계층을 분할한다. 융합 서비스가 이러한 SOA를 기본 구조로 한다면 신속한 서비스 재구조화를 이루어낼 수 있을 것이다.

셋째, 융합 서비스가 기존 제품 또는 내부 정보시스템과 정보를 연계시키고, 외부 IT 서비스와 정보를 연계시킬 때 다양한 형태의 데이터 변환, 통신 프로토콜의 변환 등이 요구된다. SOA는 이러한 시스템 통합 작업을 지원하기 위한 ESB(Enterprise Service Bus)를 지원한다.

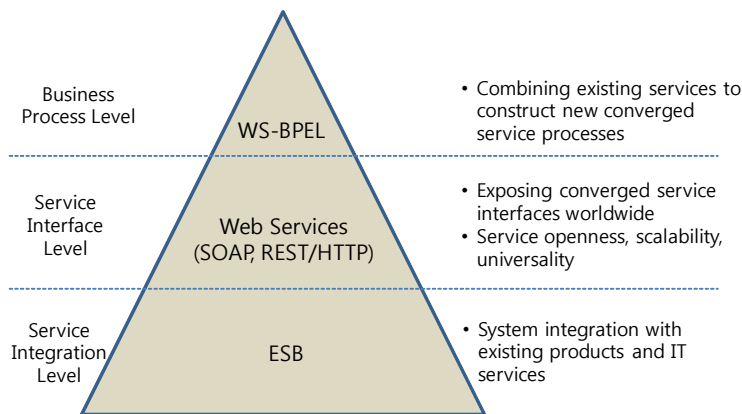
아울러 이러한 SOA의 장점에 반해 동적인 융합 환경에 따른 보안 문제가 발생할 수 있으

므로 이에 대한 세심한 주의가 필요하다[15].

<Figure 1>은 웹 서비스 기반 SOA의 관련 기술들을 프로세스 계층/서비스 인터페이스 계층/서비스 통합 계층으로 나누어 분류하고, 각각이 어떻게 융합 서비스를 지원하는지를 도시하였다. SOA 기술을 융합 서비스를 위한 정보시스템의 기반 기술로 사용한 예로는 IT와 통신 서비스의 융합을 위한 다양한 연구들을 참조할 수 있다[2, 6, 18, 23].

융합 서비스의 신속한 구현을 위해서는 외부의 서비스를 적극적으로 활용할 필요가 있다. 그런데 외부 서비스를 융합하여 새로운 서비스를 만들면 외부 서비스의 품질에 따라 융합 서비스의 품질이 크게 변동되거나 외부 서비스에 잠금효과(lock-in effect)가 발생하게 된다. 이 문제를 해결하는 한 방법은 서로 경쟁적인 외부 서비스를 묶어서 실행 시점에 서비스 품질 및 고객의 선호 등을 고려하여 동적으로 실행할 서비스를 선택하여 실행하는 것이다.

웹 서비스의 동적 바인딩 기술은 유사한 외부 서비스를 동적으로 선택하여 실행시킬 수 있는 유력한 기술적 방법이다. 웹 서비스

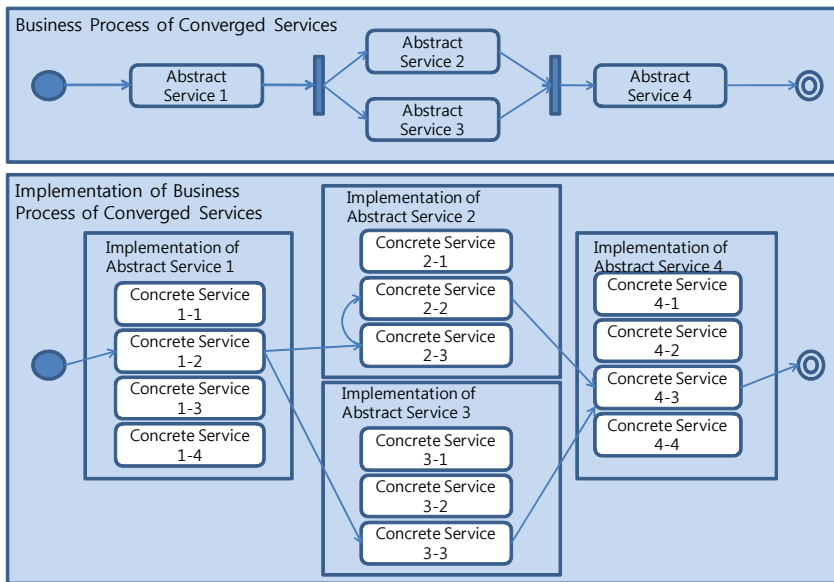


<Figure 1> Converged Services and Web Service-based SOA Technology Stack

의 동적 바인딩 기술에서는 <Figure 2>와 같이 융합 서비스를 위한 비즈니스 프로세스는 추상 서비스(Abstract Services)들에 의해 구성된다. 그리고 추상 서비스는 다수의 실제 서비스(Concrete Services)와 연결된다. 비즈니스 프로세스 상의 추상 서비스가 실행될 때 추상 서비스와 연결되어 있는 실제 서비스 중 하나가 선택되어 실행되게 된다. 예를 들어 지도 서비스라는 추상 서비스를 융합 서비스의 요소로 이용하고, 지도 서비스(추상 서비스)는 실제 서비스인 구글 지도 서비스, 다음 지도 서비스 등과 연결되어 고객의 선호나 서비스 품질들을 고려하여 실행 시점에 적절히 바인딩 될 수 있다.

웹 서비스의 동적 바인딩 기술은 추상 서비스의 바인딩 대상이 되는 실제 서비스의 발견(find)과 추상 서비스와 실제 서비스 사이의 매개(mediation) 작업을 실행 시점에 할 것인지

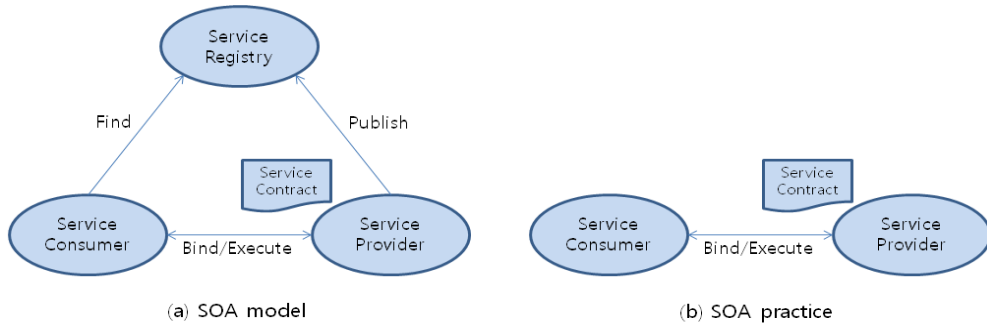
[17, 20, 21], 디자인 시점에 할 것인지[4, 16]에 따라 구분할 수 있다. 가장 이상적인 것은 실행 시점에 실제 서비스의 발견과 추상 서비스와 실제 서비스 사이의 차이의 매개 작업을 수행하는 것이다. 예를 들어 융합 서비스에 포함될 지도 서비스(추상 서비스)가 실행될 때 지도 서비스를 제공하는 실제 서비스의 목록을 웹 서비스 레지스트리에서 받아오고 그 중 가장 적합한 서비스를 선택하여 실행하는 것이다. 선택된 실제 서비스와 추상 서비스에서 제공하는 데이터 형식이나 통신 형식이 다른 경우에는 실행 시점에 데이터 및 통신 프로토콜 사이의 변환(매개)이 이루어진다. 반대로, 디자인 시점에 실제 서비스를 발견하여 추상 서비스와의 매개 작업을 미리 수행할 수도 있다. 이 경우 실행 시점에는 이미 발견되어 추상 서비스와 연결된 실제 서비스 중에서 가장 적절한 서비스가 바인딩 된다(<Table 1> 참조).



<Figure 2> Dynamic Binding of Web Services

〈Table 1〉 Comparison of Dynamic Binding Methods

Phase	Discovery and mediation at runtime	Discovery and mediation at design time
Abstract service definition	Design time	Design time
Service discovery	Runtime	Design time
Service mediation	Runtime	Design time
Selection and binding	Runtime	Runtime



〈Figure 3〉 SOA theory Versus Practice in Dynamic Service Binding (Adapted from [20])

사실 Michlmayr et al.[20]이 지적한 바와 같이 이론적으로는 SOA는 실행 시점의 서비스의 발견과 바인딩을 목표로 하고 있다. 그러나 <Figure 3>에서 볼 수 있는 바와 같이 실제 SOA의 구현은 디자인 시점에 실제 서비스의 발견과 바인딩이 주로 이루어진다. 이론과 실제가 차이가 나는 가장 큰 이유는 UDDI나 ebXML같은 서비스 레지스트리가 별로 성공하지 못했고, 기존의 서비스 레지스트리가 키워드 기반 검색만을 지원하여 실제 서비스를 동적으로 발견하는 데 충분하지 못했기 때문이다. 이러한 문제를 해결하기 위한 노력들이 있었는데 대표적으로는 Michlmayr et al.[20] 등의 연구가 있다. 여기서는 서비스

에 대한 다양한 메타 데이터를 제공하여 서비스를 실행 시점에 발견/선택하고 선택된 실제 서비스와 추상 서비스와의 차이를 메타 데이터에 기반하여 동적으로 매개하여 바인딩 하는 작업을 수행하도록 한다. 그러나 이러한 연구에도 불구하고 실행 시점에 발견/매개/선택/바인딩 하는 방식이 융합 서비스의 개발에 적용되기에는 현실성이 높지 않다. 왜냐하면 융합 서비스는 다양한 외부 서비스를 이용할 가능성이 높은데, 대부분의 외부 서비스 제공자가 서비스에 대한 통일된 메타 데이터를 제공해야만 이러한 실행 시점에 서비스 발견 및 매개가 이루어질 수 있기 때문이다. 그러나 실제 대부분의 유명한 웹 서비

스들은 자신만의 방식으로 서비스를 기술하고 있는 실정이다.

따라서 외부 서비스의 메타 데이터에 대한 특별한 가정을 하지 않기 위해서는 디자인 시점에 실제 서비스의 발견 및 매개 방식이 더 현실적인 동적 바인딩 방법이 된다. 따라서 실제 많은 경우 이미 발견된 실제 서비스를 ESB (Enterprise Service Bus) 등의 미들웨어를 이용하여 디자인 시점에 추상 서비스와 매개시키고 실행 시점에 동적으로 바인딩 하는 방식이 많이 이용된다[10, 16, 21]. 그러나 디자인 시점의 발견과 매개 방식은 추상 서비스가 맨 처음 도입될 때는 매우 성공적으로 작동하지만, 추상 서비스에 바인딩 되는 실제 서비스가 진화되어 추가, 삭제, 변경되는 경우 이를 동적으로 반영하기 어렵다는 약점을 가지고 있다. 따라서 실제 서비스가 변화될 때 이를 반영할 수 있는 방식에 대한 연구들이 수행되어 왔다 [9, 10, 17].

이러한 동적 바인딩 방식에 대한 연구 이외에도 동적 바인딩에 대한 다양한 연구가 수행되어 왔다. 대표적으로는 서비스의 동적 바인딩이나 조합이 이루어질 때 추상 서비스나 조합된 서비스의 QoS를 결정하는 방법에 대한 연구 등이 있다[3, 5, 19].

그런데 실행 시점에 발견 및 매개가 이루어지는 방식이든 디자인 시점에 발견 및 매개가 이루어지는 방식이든 모두 공통적으로 추상 서비스의 인터페이스는 고정된 상태에서 실제 서비스가 변화하게 되면 이를 어떻게 동적으로 발견하고 매개할 것인가를 주요 점을 두고 있다. 그런데 이러한 추상 서비스의 인터페이스를 고정시키는 동적 바인딩 방식은 다음의 약점을 가지고 있다.

첫째, 추상 서비스는 실제 서비스의 공통 기능으로 한정되기 때문에 추상 서비스를 이용하는 융합 서비스는 실제 서비스들이 제공하는 풍부한 기능 보다는 다수의 실제 서비스가 제공하는 평범한 공통 기능만을 기반해야 한다.

둘째, 실제 서비스가 계속적으로 진화하더라도 추상 서비스는 이러한 변화를 반영하지 못 하므로 이를 이용하는 융합 서비스의 진화도 한계를 가질 수밖에 없다.

셋째, 특정 실제 서비스만 제공하는 개인화된 기능이 있더라도 추상 서비스에 이를 반영하기 어려워 사용자에게 개인화된 융합 서비스 제공에 한계를 가진다.

본 연구에는 이러한 기존의 공통된 추상 서비스 인터페이스만을 이용한 동적 바인딩 방식의 한계를 개선하기 위해 다음의 프로파일 기반 동적 바인딩 기술을 제안한다.

3. 융합 비즈니스 시나리오- 스마트 트레드밀 u-Health 융합 서비스

본 절에서는 융합 서비스를 위한 프로파일 기반 동적 바인딩 시스템의 제안 동기와 타당성을 제시하기 위해 다음과 같은 가상의 비즈니스 시나리오를 기반으로 스마트 트레드밀(treadmills)을 이용한 u-Health 융합 서비스를 고려한다.

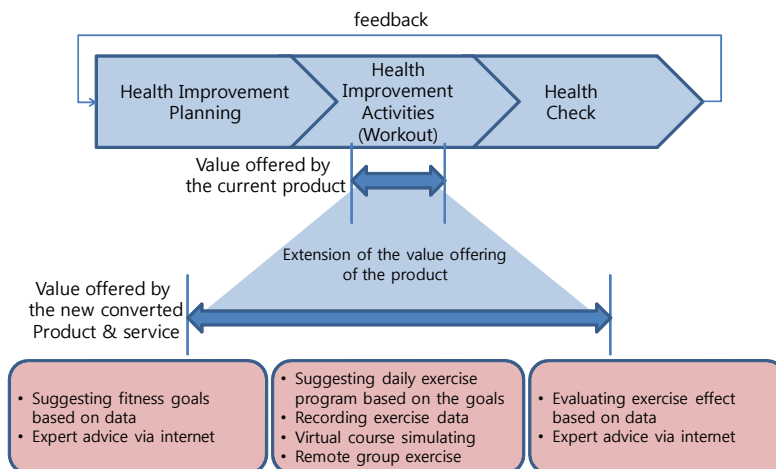
- 건강에 대한 관심이 커짐에 따라 언제 어디서나 건강관리를 받을 수 있는 u-Health에 대한 관심도 커지고 있다. 트레드밀 회사 A는 자신의 제품에 IT 서비스를 융합하

여 새로운 u-Health 서비스를 제공하고 자 한다.

- A사는 고객이 트레드밀을 이용하는 본인적인 욕구는 건강 증진을 위한 노력에 있고, 고객의 건강 증진 노력은 <Figure 4> 같이 건강 증진 목표 수립, 건강 증진 활동 (달리기 운동 등), 건강 상태 평가의 사이클로 이루어져 있다고 파악하였다.
- <Figure 4>에서 보는 바와 같이, 현재 A사는 자사의 트레드밀 제품은 고객의 건강 증진 노력 사이클의 한 단계인 건강 증진 활동에만 가치를 제공하고 있음을 파악하고, IT 서비스와의 융합을 통해 고객의 건강 증진 노력의 전 사이클에 가치를 제공하는 융합된 제품 및 서비스를 제공하기로 결정하였다.
- 먼저 건강 증진 목표 수립 단계에서는 수집된 데이터를 기반으로 나이, 성별, 운동 이력 등을 종합적으로 고려하여 달리기 운동 목표를 고객의 컴퓨터 또는 스마트

폰 등에 제시하고, 고객이 운동 목표를 수락하면 이에 따라 트레드밀에 운동 프로그램을 입력한다. 또한 운동 목표 수립 시 전문가의 조언을 음성, 문자, 화상 등을 통해 얻을 수 있도록 서비스 한다.

- 건강 증진 활동 단계에서는 고객이 수립한 운동 목표와 현재까지 수행된 운동 활동 기록을 이용하여 현 단계의 운동 프로그램을 제시해 준다. 고객이 실제 운동을 수행하면 달리기 기록을 상세 구간별로 기록하고 여러 각도로 고객이 현 단계 운동 프로그램을 달성하고 있는지 보여준다. 또한, 운동이 지속되기 위해서는 재미와 함께 소셜 활동이 크게 도움이 된다고 파악하여, 고객이 실제 달리기 코스를 선택하면 실제 코스에 맞추어 트레드밀의 경사도 등을 조정해 주고 각 달리기 코스의 완주 및 구간별 순위 등을 알려준다. 또한 SNS를 이용해 원격으로 떨어진 트레드밀을 통해 단체 달리기 등을 지원한다.



<Figure 4> Value Offering Extension of an Existing Product Via Converged Services

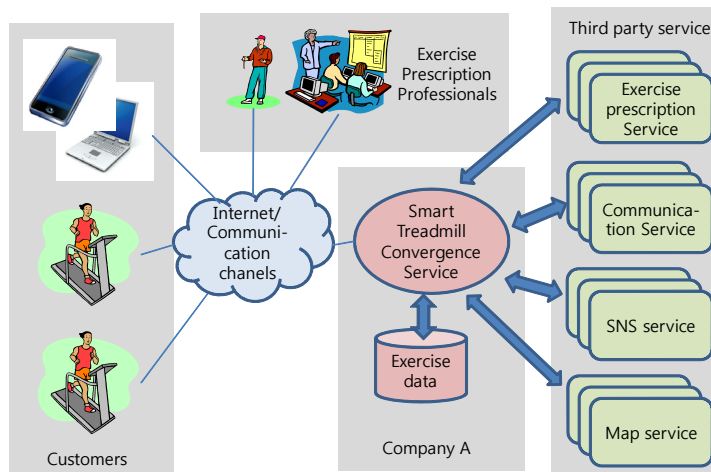
- 건강 증진 평가 단계에서는 건강 증진 활동 단계에서 축적된 고객 데이터와 통계 데이터 등을 이용하여 컴퓨터나 스마트폰 등을 통해 건강 증진 목표 달성을 평가하고 개선되어야 할 사항 등을 조언해 준다. 전문가의 직접적인 조언을 위해 전문가와 음성, 화상, 문자로 의견을 교환할 수 있도록 지원한다.
- A사는 고객에게 이러한 확장된 가치를 제공하기 위해서는 제품과 결합된 다양한 서비스의 구축이 필요하나 제조업 기반의 A사가 이러한 모든 서비스를 제공하기는 역량이 한정되어 있음을 판단하였다. 따라서 기존의 최고의 IT 서비스 제공자의 서비스를 융합하여 ‘스마트 트레드밀 융합 서비스’를 제공하기로 결정하였다.

<Figure 5>는 ‘스마트 트레드밀 융합 서비스’의 구현을 위한 정보시스템의 개념도이다. A사는 ‘스마트 트레드밀 융합 서비스’를 위한

정보시스템을 다음과 같이 개발하고자 한다.

- A사는 ‘스마트 트레드밀 융합 서비스’에서 트레드밀과 통신하고 데이터를 저장하는 기능과 고객의 스마트폰이나 컴퓨터를 통해 운동 정보를 제공하는 것이 핵심 역량이라 고려하여 이 기능은 A사가 구현하기로 하였다.
- 전문가와의 통신을 위한 음성, 화상, 문자 통신 서비스, SNS를 이용한 단체 운동 활동 지원 서비스, 실제 달리기 코스를 시뮬레이션 하기 위한 지도 서비스 등은 핵심 역량이 아니라 보고 기존 서비스 사업자의 서비스를 융합하기로 하였다. 또한 향후 사업 확장 영역이지만 아직 데이터 수집 및 분석 능력의 미비로 외부 운동 처방과 관련된 데이터 제공 서비스를 융합하여 서비스를 제공하기로 하였다.

A사는 스마트 트레드밀 융합 서비스와 외부 서비스 제공자의 서비스를 융합할 때 다



<Figure 5> Information System for a Smart Treadmill Service

음과 같은 세 가지 요구사항을 제시하였다.

첫째, 고객의 선호와 상태에 따라 이를 특화하여 서비스 할 수 있는 최적의 외부 서비스가 동적으로 바인딩 될 수 있어야 한다. 예로 어떤 고객은 화상 통신을 통한 전문가 조언을 얻기를 원하지만 어떤 고객은 화상 통신이 불가하거나 문자를 통한 통신을 선호할 수 있다. 따라서, 통신 서비스의 융합은 한 서비스 사업자나 서비스 타입에 종속되지 않고 고객의 선호와 상태에 따라 특화된 최적의 서비스가 동적으로 바인딩 되기를 희망한다.

둘째, A사는 외부 서비스 제공자의 서비스 상태에 완전히 종속되는 사항을 피하기를 원한다. 외부 사업자의 서비스 제공 형태나 품질이 변하거나 서비스 자체가 중단됨에 따라 자신의 서비스가 크게 바뀌거나, 품질이 저하되거나, 중단되어서는 안 되며 항상 안정적인 융합 서비스를 제공하기를 원한다. 이를 위해 다수의 유사한 서비스 제공자를 지속적으로 평가하여 항상 최적의 서비스 제공자를 동적으로 선택하여 실행될 수 있기를 바란다.

셋째, A사는 서비스의 지속성을 중요하게 생각하므로 융합 서비스의 중단 없이 외부 서비스의 변화를 수용할 수 있기를 원한다.

4. 프로파일 기반 동적 바인딩 기술의 개념 및 구조

본 장에서는 제 3장에서 제시한 융합 서비스 시나리오를 고려하여 융합 서비스를 위한 프로파일 기반 동적 바인딩 기술이 구현해야 할 기본 구조를 제시한다.

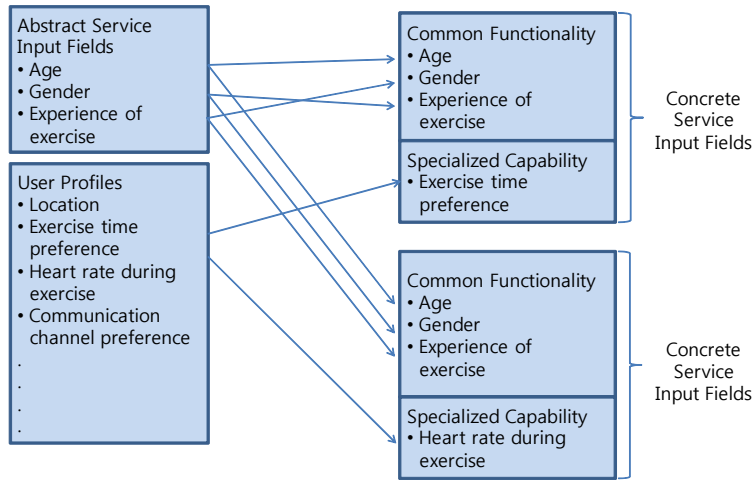
융합 서비스를 위한 프로파일 기반 동적

바인딩 기술은 실제 서비스가 진화함에 따라 추상 서비스도 같이 진화할 수 있도록 한다. 이를 통해 추상 서비스를 이용하는 융합 서비스도 동적으로 계속 진화해 나갈 수 있도록 한다.

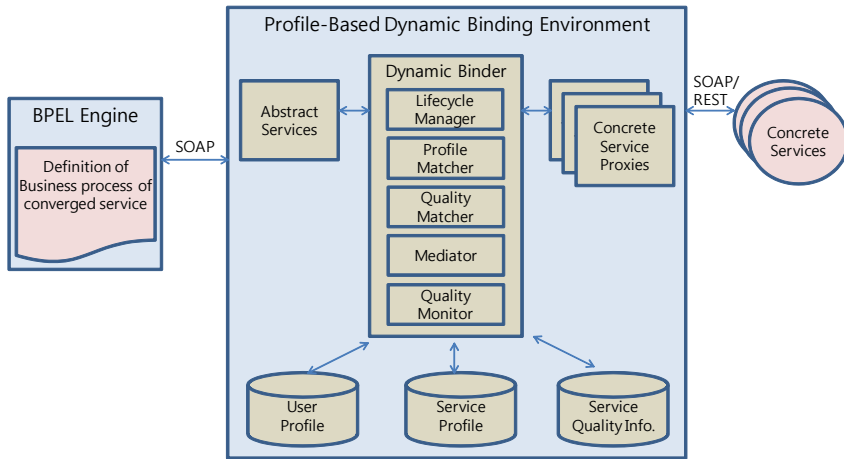
융합 서비스를 위한 프로파일 기반 동적 바인딩 기술의 핵심 개념은 실제 서비스의 기능을 공통 기능 요소(functionality)와 특화 역량 요소(capability)로 나뉘서 공통 기능 요소는 추상 서비스의 입력 파라미터로 특화 역량은 사용자 프로파일로 기술한다는 데 있다.

<Figure 6>에서 보는 바와 같이 실제 서비스는 여러 가지 기능을 구현하기 위해 다양한 입력 필드가 정의되어 있고 실행 시점에 입력된 정보에 따라 결과를 반환하게 된다. 예를 들어 사용자의 나이, 성별, 운동 경력에 따라 적절한 운동 처방을 해 주는 두 개의 실제 서비스가 있다고 하자. 한 서비스는 사용자의 운동 시점의 심박수 등의 메디컬 정보를 추가로 이용하여 운동 처방을 해주는 서비스이고, 다른 하나의 서비스는 선호 운동 시간대에 따른 세분화된 운동 처방을 해주는 특화된 기능이 있다고 하자. 그러면 두 실제 서비스의 공통 기능 요소는 추상 서비스의 입력필드로 정의하여 추상 서비스 호출 시 항상 입력되도록 하고, 서로 다른 기능은 특화 역량 요소로 정의하여 사용자 프로파일에서 입력되도록 한다.

<Figure 7>은 프로파일 기반 동적 바인딩 시스템의 전반적인 구조를 보여주고 있다. 융합 서비스는 WS-BPEL에 의해 비즈니스 프로세스가 정의되고 프로파일 기반 동적 바인딩 시스템에 있는 추상 서비스를 호출한다. 프로파일 기반 동적 바인딩 시스템은 추상 서



〈Figure 6〉 Profile-based Dynamic Binding Method



〈Figure 7〉 Overall Architecture of the Profile-based Dynamic Binding System

비스를 SOAP 기반 웹 서비스로 노출시키고, 추상 서비스가 호출될 때마다 동적 바인더가 실행된다. 동적 바인더는 추상 서비스에 연결된 실제 서비스 중 가장 적절한 실제 서비스를 실제 서비스 프락시를 통해 호출하고 그 결과를 추상 서비스에 반환한다. 동적 바인더는 가장 적절한 실제 서비스를 선택하고 실행하기 위하여 사용자 프로파일, 서비스 프로파

일, 서비스 품질 정보 데이터를 참조한다.

사용자 프로파일은 사용자의 직접 입력 또는 사용자가 이용하는 기기를 통해 컨텍스트 정보를 수집하여 생성된다. 예를 들어 융합 서비스를 이용하는 고객이 융합 서비스 이용 등록 시 입력한 정보나 트레드밀이나 스마트 기기 이용 시에 수집되는 데이터들이 사용자 프로파일에 저장된다. 사용자 프로파일에 저

장될 수 있는 데이터는 융합 서비스 별로, 사용자 별로 다양할 수 있으므로 저장되는 정보의 유연성을 위해 키-값(key-value)의 형태로 저장된다. 키의 명명은 일관된 방식으로 수행되어야 하며 ‘.’으로 구분되는 계층구조로 키를 명명할 수 있다. 예를 들어 운동 시간에 대한 선호는 workout.preference.time 등으로 명명할 수 있다.

서비스 프로파일은 추상 서비스와 실제 서비스에 대한 정보를 사용자 프로파일과 마찬가지로

키-값 형태로 저장한다. 사용자 프로파일이 융합 서비스 별로 사용자 별로 임의의 키 값을 가질 수 있는 데 반해, 서비스 프로파일은 추상 서비스와 실제 서비스의 매칭을 위해 정해진 프로파일 키 값이 있다. 서비스 프로파일의 주요 키에 대한 설명은 <Table 2>에 제시되어 있다.

서비스 품질 정보는 실제 서비스의 응답 시간, 성공/실패/시간초과 여부에 대한 정보를 가지고 있다. 그리고 이러한 정보를 기반

<Table 2> Description of Service Profile Keys

Key names	Description
service.id	IDs to uniquely identify an abstract or concrete service in the system
service.type	Type describing whether the service is abstract or concrete.
binding.interface	The qualified name of a binding interface through which abstract and concrete services are bound. Each concrete service proxy has to implement more than one binding interface.
service.capabilities	Service capabilities that a concrete service specializes in. All user profile items which the service deals with are listed.
service.required	Service capabilities that a abstract service require for concrete services. If not defined, the abstract service considers all service capabilities of concrete services for selection.
binding.profile.matching.strategy	The profile matching strategy that the abstract service takes for selection of concrete services. There are EXACT_MATCH and BEST_MATCH strategies.
binding.quality.matching.strategy	The quality matching strategy that the abstract service take for selection of concrete services. There are ALWAYS_BEST_ONE, RANDOM, and QUALITY_WEIGHTED_RANDOM strategies
service.quality.max_score_per_invocation	The max score that a concrete service can get as quality score on each invocation.
service.quality.initial_score	The initial quality score that a concrete service has at the beginning of its installation in the system. Generally, it is set between 0 and service.quality.max_score_per_invocation.
service.quality.moving_ave rage_coefficient	The quality score of a concrete service is cumulated on each invocation using the exponential moving average. This key is the coefficient of the exponential moving average, and is set between 0 and 1.
service.response_time.thres hold	A threshold until which the abstract service waits the response of a concrete service it invokes. If the response time exceeds the threshold, then an exception occurs and this invocation is recorded as TIME_OUT for quality evaluation.

으로 서비스 품질에 대한 순위 산정 정보를 가지고 있다.

동적 바인더는 호출되면, 앞서 설명한 사용자 프로파일, 서비스 프로파일, 서비스 품질 정보를 가지고 크게 다음 5가지 기능을 통해 실제 서비스를 선택하여 실행하게 된다.

- **라이프사이클 관리** : 프로파일 기반 동적 바인딩 기술은 추상 서비스나 실제 서비스 프락시가 변경을 위해 재 인스톨 되거나 관리를 위해 시작, 정지되는 상황을 동적으로 파악하여 이를 동적 바인딩에 적절히 반영할 수 있어야 한다. 이를 위해 동적 바인더는 추상 서비스와 실제 서비스 프락시가 시스템에 인스톨/시작/정지/언인스톨되는 라이프사이클 이벤트를 모니터링 한다. 동적 바인더는 추상 서비스와 실제 서비스 프락시의 라이프사이클 이벤트가 발생하면 서비스 프로파일에서 서비스 ID, 서비스 타입, 바인딩 인터페이스를 확인하여 추상 서비스와 실제 서비스와의 관계를 업데이트한다. 각각의 라이프사이클 이벤트에서 발생하는 바인딩 관계의 변화는 서비스 프로파일에 기술되어 있다.
- **프로파일 매칭** : 추상 서비스가 호출되면 동적 바인더는 추상 서비스를 호출한 고객 ID를 확인하여 고객 ID에 등록된 사용자 프로파일 키와 값을 가져온다. 추상 서비스에 service.required 정보가 있으면 여기에 부합되는 사용자의 프로파일 키와 값만을 필터링 한다. 필터링 후의 사용자 프로파일 키와 실제 서비스의 service.capabilities에 기술된 키 값을 - 실제 서비스가 처리할 수 있는

사용자 프로파일 키 값이 나열되어 있다 (<Table 2> 참조) - 매칭하여 실제 서비스를 평가한다. 추상 서비스에 기술된 프로파일 매칭 전략이 EXACT_MATCHING이면 필터링 후의 사용자 프로파일 키와 실제 서비스의 특화 역량의 차이가 가장 적은 실제 서비스가 선택 된다. BEST_MATCHING이면 필터링 후의 사용자 프로파일 항목을 가장 많이 처리할 수 있는 실제 서비스가 선택 된다. <Figure 8>에 프로파일 매칭의 개략적인 알고리즘을 의사 코드로 제시 되어 있다.

- **품질 매칭** : 동적 바인더는 프로파일 매칭을 수행한 후 선택된 실제 서비스가 다수인 경우 품질 매칭을 수행한다. 품질 매칭은 추상 서비스에 연결된 실제 서비스에 대한 평균 품질 점수를 기준으로 수행된다. 추상 서비스의 프로파일에 입력된 품질 매칭 전략이 ALWAYS_BEST_ONE 전략인 경우에는 항상 가장 높은 품질 점수를 가진 서비스가 호출된다. RANDOM 전략인 경우에는 같은 확률을 가지고 임의로 선택된다. QUALITY_WEIGHTED_RANDOM 전략인 경우에는 평균 품질 점수에 비례하는 확률로 선택된다.
- **추상 서비스와 실제 서비스의 매개** : <Figure 6>에서 볼 수 있듯이 실제 서비스의 입력 필드는 추상 서비스의 입력 필드와 사용자 프로파일의 정보로 구성되어 있다. 추상 서비스와 실제 서비스의 매개는 추상 서비스 입력 필드를 실제 서비스 입력 필드로 매개하여 이루어지고, 사용자 프로파일과 실제 서비스의 매개

는 프로파일 매칭에서 필터링 후의 사용자 프로파일 정보를 가지고 실제 서비스의 입력 필드와 매개한다.

- **품질 모니터** : 추상 서비스와 실제 서비스와의 매개 작업이 완료되면 호출이 된다. 시스템은 호출에 대한 응답 시간과 호출 성공/실패/시간 초과를 기록한다.

실제 서비스에 대한 품질 모니터링 후 호출된 실제 서비스에 대한 호출당 품질 점수와 평균 품질 점수를 다음과 같이 계산한다. 어떤 실제 서비스의 n 번째 호출에 의해 받는 호출 당 품질 점수를 Y_n 이라고 하면 Y_n 은 다음과 같이 계산된다.

〈Table 3〉 Lifecycle Events and the Changes of Binding Relationships

Lifecycle events	Abstract Service	Concrete Service Proxy
Install	Register relations with concrete service proxies of the identical binding interface	Register relations with abstract services of the identical binding interface
Start	-	Make the concrete service proxy available for binding targets of abstract service of the identical binding interface
Stop	-	Make the concrete service proxy unavailable for binding targets of abstract service of the identical binding interface
Uninstall	Unregister relations with concrete service proxies of the identical binding interface	Unregister relations with abstract services of the identical binding interface

```

1:   serviceCandidated ← getConcreteServiceProxy()
2:   if (# of serviceCandidates == 0) then
3:       return null
4:   end if
5:   if (# of serviceCandidates == 1) then
6:       return serviceCandidates[0]
7:   end if
8:   userProfiles ← getUserProfile(userId)
9:   if (userProfiles not null) then
10:      filteredProfile ← filterProfiles(userProfiles)
11:   end if
12:  sorted ← new sortedMap<Integer, List<ServiceCandidate>>
13:  for all (candidate in serviceCandidates) do
14:      profileScore ← contextMatched(filteredProfiles, candidate)
15:      sorted.get(profileScore).add(candidate)
16:  end for
17:  return sorted.get(sorted.lastKey());
    
```

〈Figure 8〉 Profile Matching Algorithm

$$Y_n = \begin{cases} Y \times \frac{R_T - R_n}{R_T} & \text{호출이 성공되었을 때} \\ 0, & \text{호출이 실패되었을 때} \end{cases}$$

단, Y 는 서비스 프로파일에 `service.quality.max_score_per_invocation`로 정의된 호출당 실제 서비스가 받을 수 있는 최대 품질 점수, R_T 는 서비스 프로파일에 `service.response_time.threshold`로 정의된 실제 서비스의 응답 시간의 최대 `threshold(ms)` 값, R_n 은 실제 서비스의 n 번째 호출 시에 측정된 응답 시간(`ms`).

실제 서비스의 n 번째 호출 후의 평균 품질 점수를 S_n 이라고 하면 S_n 은 다음과 같이 계산된다.

$$S_n = (1 - \alpha)S_{n-1} + \alpha Y_n, \quad n = 1, 2, \dots$$

단, α 는 서비스 프로파일에 `service.quality.moving_average_coefficient`로 정의된 `exponential moving average`의 `coefficient`, S_0 는 서비스 프로파일에 `service.quality.initial_score`로 설정된 값으로 운영자에 의해 초기에 설정되며 $0 < S_0 \leq Y$ 의 값을 가진다.

5. 프로파일 기반 동적 바인딩 기술의 프로토타입 구현

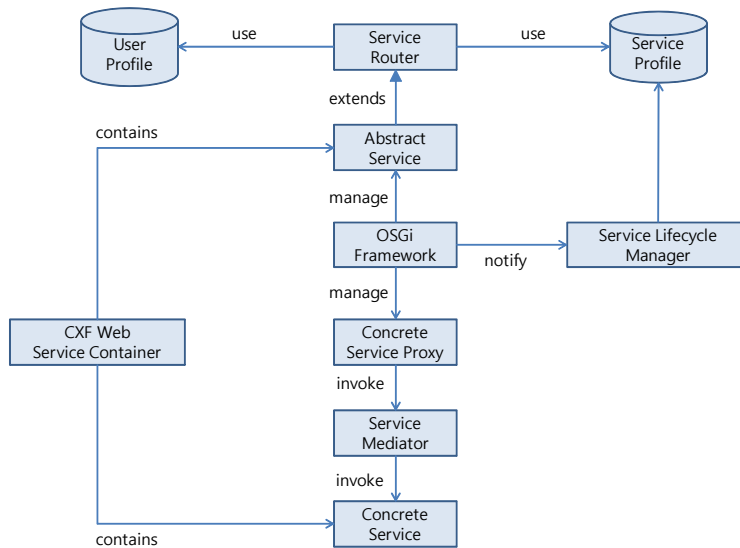
본 장에서는 융합 서비스를 위한 프로파일 기반 동적 바인딩 기술의 실제 구현 방법을 제시한다. 제시되는 구현 사례에서는 Java를 이용하여 구현되었고, 다음의 구현 기술을 활용하였다.

- **OSGi 프레임워크** : OSGi 기술은 종종 Java를 위한 동적 모듈 시스템으로 일컬

어진다[22]. OSGi 기술은 원래 가정용 게이트웨이 등 소규모 시스템의 서비스의 라이프사이클 관리를 위해 도입되었으나 최근 대규모 서비스 플랫폼의 근간 기술로 확장되고 있다. OSGi framework은 Bundle이라 불리는 Java의 논리적 모듈의 설치, 시작, 중단, 삭제되는 전 과정을 통해 모듈 간의 의존관계 및 상호작용을 동적으로 관리해 준다. 본 연구에서는 추상 서비스와 실제 서비스 프락시의 동적인 라이프사이클 관리를 위해 OSGi 프레임워크의 라이프사이클과 서비스 계층을 적극적으로 활용하였다.

- **Apache Camel** : 시스템의 확장성을 고려할 때, 다양한 형태의 프로토콜과 데이터를 이용하여 추상 서비스와 실제 서비스를 매개할 수 있어야 한다. 따라서, 프로파일 기반 동적 바인딩 기술의 구현은 새로운 mediation 프레임워크를 개발하기 보다는 검증되고 폭넓은 mediation 기능을 가진 Apache Camel을 이용하여 매개 기능을 구현하였다. Apache Camel은 open source로 Java 기반의 framework으로 매우 빠르게 발전하고 있다[12].
- **Apache CXF** : 추상 서비스를 SOAP 방식의 웹 서비스를 노출하고, 실제 서비스 프락시가 SOAP이나 REST 방식으로 구현된 실제 서비스를 호출하기 위한 프레임워크로 Apache CXF를 이용하였다. Apache CXF는 JAX-WS와 JAX-RS를 구현하고 있어 SOAP 기반 웹 서비스와 REST 기반 웹 서비스 모두를 쉽게 구현하고 호출할 수 있다[1].

위의 기반 기술을 이용하여 프로파일 기반



〈Figure 9〉 A Prototype of Profile-based Dynamic Binding Systems

동적 바인딩 시스템은 〈Figure 9〉와 같이 구현되었다. 프로토타입의 구현에서는 추상 서비스는 OSGi 서비스로 구현되었으며 동적 바인더인 서비스 라우터 클래스를 구현하고 CXF를 통해 SOAP 방식 웹 서비스로 노출된다.

동적 바인더인 서비스 라우터는 OSGi 서비스 계층에 의존적이다. 서비스 라우터는 OSGi 플랫폼으로부터 OSGi 서비스 이벤트를 모니터링 한다. 이를 통해 추상 서비스와 실제 서비스 프락시의 추가/삭제 등을 감지한다. 추상 서비스가 호출되면 서비스 라우터는 OSGi 플랫폼을 이용해 추상 서비스와 연결된 실제 서비스 프락시 목록을 받아온 후 제 4장에서 설명한 프로파일 매칭과 품질 매칭을 수행하여 최적의 실제 서비스 프락시를 선택하여 호출을 한다.

실제 서비스 프락시도 OSGi 서비스로 구현되어 있다. Apache Camel 프레임워크를 이용하여 추상 서비스와 사용자 프로파일의

정보를 실제 서비스에 맞게 매개 작업을 수행하고, Apache CXF를 통해 실제 서비스를 호출하게 된다.

추상 서비스와 실제 서비스 프락시를 OSGi 서비스로 구현하는 기술로는 blueprint 기술을 이용하였다. blueprint 기술은 OSGi 서비스 컴포넌트 구현 기술로 Spring 프레임워크와 유사한 방식으로 의존관계를 관리한다. blueprint의 장점은 Spring 프레임워크와 마찬가지로 구현 클래스 간 의존관계 XML 설정 파일 등에 의해 외부화 할 수 있으며 서비스 프로퍼티 설정도 XML 설정 파일에서 수행할 수 있다는 데 있다.

본 장에서는 융합 서비스를 위한 프로파일 기반 동적 바인딩 시스템의 프로토타입을 구현하였다. 시스템에 대한 평가는 제 2장에서 제시한 스마트 트레드밀 융합 서비스 시나리오에 의해 도출된 요구사항이 프로파일 기반 동적 바인딩 기술을 얼마나 잘 지원하는가를

<Table 4> Evaluation of the Profile-based Dynamic Binding Based on the Prototype Implementation

Requirements of business scenario	Functions	Evaluation based on the prototype implementation
Personalized converged Service	Service matching based on profile	A communication channel is selected depending on the user's preference. Text or voice communication can be selected.
Dynamic replacement of external service based on quality records	Service matching based on quality	The response time and availability of each text or voice communication service are recorded, and the quality score of the service is evaluated at each invocation. The better quality score a service has, the more chance it has to be selected for dynamic binding
.Converge service change management without interruption	OSGi-based service lifecycle management	A new text or voice communication service can be added to the converged service without interruption of the converged service. Also, the deletion of an existing concrete service proxy can be performed without interruption of the converged service.

통해 이루어 질 수 있다. 실제 스마트 트레드밀 융합 서비스 시나리오의 구현은 트레드밀을 통한 정보 수집 또는 시뮬레이션이 필요하므로 본 프로토타입의 구현에서는 스마트 트레드밀 융합 서비스에 통신 서비스를 융합하는 시나리오 관점에서만 평가해 본다.

<Table 4>는 융합 서비스 시나리오 상의 요구사항 별로 프로파일 기반 동적 바인딩 기술 요소와의 대응 관계와 프로토타입 구현 관점에서의 평가 결과를 정리한 것이다. <Table 4>에서 볼 수 있는 바와 같이 프로파일 기반 동적 바인딩 기술은 융합 서비스의 요구사항을 적절히 대응하고 있는 것으로 평가된다.

6. 결 론

최근 모든 산업에서 융합을 통한 새로운 사업 기회를 창출하려는 노력이 가속화되고 있고, 서비스 산업 분야도 예외가 아니어서

기존의 서비스를 융합하거나 IT 서비스와 기존 서비스를 융합하려는 시도가 지속되고 있다. 그런데 융합 서비스를 빠른 시간 안에 시장에 제공하기 위해서는 외부 서비스와 자신의 핵심 역량을 융합하는 능력이 필요하며, 정보 시스템 아키텍처는 이러한 서비스 융합을 효과적으로 지원해야 한다. 서비스 지향 아키텍처는 이러한 서비스 융합을 성공적으로 지원할 수 있는 정보시스템 아키텍처이다.

본 논문에서는 융합 서비스의 동적인 진화를 지원하기 위한 프로파일 기반 동적 서비스 바인딩 기법의 도입을 제안하였다. 프로파일 기반 동적 서비스 바인딩 기술의 필요성을 제시하기 위해 스마트 운동 기구와 관련된 융합 서비스 시나리오를 제시하고 해당 기술이 구현해야 할 요구 사항을 도출하였다. 그리고 프로파일 기반 동적 서비스 바인딩 기술의 개념 및 구조를 제안하였다. 마지막으로 OSGi, Apache Camel, Apache CXF 등을 이용하여 프로파일 기반 동적 서비스 바인딩 기술의 프로토타입을

구현하였다. 프로토타입 구현 결과를 통한 간략한 평가에서 볼 수 있듯이 제안된 기술은 융합 서비스의 동적인 진화를 위한 적절한 기술로 판단된다.

본 연구의 한계는 프로토타입 구현 범위가 융합 서비스 시나리오 전체에 대한 구현을 통한 제안된 기술의 평가가 이루어지지 않았다는 점과 다양한 프로파일 매칭과 품질 매칭 전략에 따른 시스템의 성능 분석이 되지 못한 점이다. 이는 추후 연구로 보강될 수 있으리라 기대한다.

References

- [1] Balani, N. and Hathi, R., *Apache CXF Web Service Development*, Packt Pub Limited, 2009.
- [2] Bo, C., Yang, Z., Peng, Z., Hua, D., Xiaoxiao, H., Zheng, W., and Junliang, C., "Development of Web-Telecom based hybrid services orchestration and execution middleware over convergence networks," *Journal of Network and Computer Applications*, Vol. 33, No. 5, pp. 620-630, 2010.
- [3] Calinescu, R., Grunske, L., Kwiatkowska, M., Mirandola, R., and Tamburrelli, G., "Dynamic QoS management and optimization in service-based systems," *Software Engineering*, *IEEE Transactions on*, Vol. 37, No. 3, pp. 387-409, 2011.
- [4] Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L., "A framework for QoS-aware binding and re-binding of composite web services," *Journal of Systems and Software*, Vol. 81, No. 10, pp. 1754-1769, 2008.
- [5] Cardellini, V., Casalicchio, E., Grassi, V., Lo Presti, F., and Mirandola, R., "QoS-driven runtime adaptation of service oriented architectures," *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pp. 131-140. 2009.
- [6] Choi, Y. I., Park, Y. M., Lee, B. S., and Kim, S. H., "A Trend Analysis of Platform Technology for Next Generation Telecom Service Convergence(차세대 통신 서비스 융합을 위한 플랫폼 기술 동향)," *Information and Communications Magazine*, Vol. 24, No. 7, pp. 8-16, 2007.
- [7] Di Penta, M., Esposito, R., Villani, M. L., Codato, R., Colombo, M., and Di Nitto, E., "WS Binder : A framework to enable dynamic binding of composite web services," *Proceedings of the 2006 international workshop on Service-oriented software engineering*, pp. 74-80, 2006.
- [8] Erl, T., *Service-oriented architecture*, Prentice Hall Englewood Cliffs, 2004.
- [9] Garcia, I., Pedraza, G., Debbabi, B., Lalanda, P., and Hamon, C., "Towards a service mediation framework for dynamic applications," *Services Computing Conference (APSCC)*, 2010 IEEE Asia-Pacific,

- pp. 3-10, 2010.
- [10] González, L. and Ruggia, R., "Towards dynamic adaptation within an ESB-based service infrastructure layer," Proceedings of the 3rd International Workshop on Monitoring, Adaptation and Beyond, pp. 40-47, 2010.
- [11] Rho, M.-J., Kim, J.-H., and Lee, J.-B., "An Exploratory Study on Smart-Phone and Service Convergence," The Journal of Society for e-Business Studies, Vol. 15, No. 4, pp. 59-77, 2010.
- [12] Ibsen, C. and Anstey, J., Camel in action. Manning Publications Co., 2010.
- [13] Kim, D.-H., "A Study on Modeling Framework of Convergence Business," The Journal of Society for e-Business Studies, Vol. 17, No. 4, pp. 175-196, 2012.
- [14] Kim, E.-J., Ahn, J.-H., and Park, C.-W., "Understanding the Adoption of Telecommunications-Broadcasting Convergence Services : The Case of DMB," The Journal of Society for e-Business Studies, Vol. 12, No. 4, pp. 53-69, 2007.
- [15] Kim, Y.-K. and Doh, K.-G., "SOA Vulnerability Evaluation using Run-Time Dependency Measurement," The Journal of Society for e-Business Studies, Vol. 16, No. 2, pp. 129-142, 2011.
- [16] Ko, S., Kim, E., and Lee, C., "Dynamic Web Service Composition Support for OSGi Environment," Journal of the Korean Society of Computer and Information, Vol. 14, No. 11, pp. 145-157, 2009.
- [17] Leitner, P., Rosenberg, F., and Dustdar, S., "Daios : Efficient dynamic web service invocation," Internet Computing, IEEE, Vol. 13, No. 3, pp. 72-80, 2009.
- [18] Lu, H., Zheng, Y., and Sun, Y., "The Next Generation SDP Architecture : Based on SOA and Integrated with IMS," Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on, Vol. 3, pp. 141-145, 2008.
- [19] Mabrouk, N. B., Beauche, S., Kuznetsova, E., Georgantas, N., and Issarny, V., "QoS-aware service composition in dynamic service oriented environments," Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware, pp. 123-142, 2009.
- [20] Michlmayr, A., Rosenberg, F., Leitner, P., and Dustdar, S., "End-to-end support for qos-aware service selection, binding, and mediation in vresco," Services Computing, IEEE Transactions on, Vol. 3, No. 3, pp. 193-205, 2010.
- [21] Morand, D., Garcia, I., and Lalande, P., "Autonomic enterprise service bus," Emerging Technologies and Factory Automation (ETFA), 2011 IEEE 16th Conference on, pp. 1-8, 2011.
- [22] O. S. G. Alliance, Osgi service platform, release 3. IOS Press, Inc., 2003.
- [23] Shao, X., Chai, T. Y. , Lee, T. K., Ngoh, L. H., Zhou, L., and Kirchberg, M., "An integrated telecom and IT service delivery platform," Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE, pp. 391-396, 2008.

저 자 소 개



김길환 (E-mail : khkim@smu.ac.kr)
1994년 한국과학기술원 경영과학과 (학사)
1996년 한국과학기술원 산업경영학 (석사)
1998년~2003년 LG CNS 선임컨설턴트
2003년~2005년 기업정보화지원센터 선임연구원
2009년 한국과학기술원 산업 및 시스템공학 (박사)
2009년~2012년 한국전자통신연구원 선임연구원
2012년~현재 상명대학교 경영공학과 조교수
관심분야 확률적 최적화 이론, 서비스 플랫폼



김창섭 (E-mail : cskeum@etri.re.kr)
1992년 서울시립대학교 전산통계학 (학사)
2005년 카네기멜론 대학 소프트웨어공학 (석사)
2009년 KAIST 박사과정 수료
2001년 University of Florida 방문 연구원
1994년~현재 한국전자통신연구원 근무
미래인터넷서비스연구팀 책임연구원
관심 분야 SOA, Cloud Mobile Service, SDP, 서비스 플랫폼



배현주 (E-mail : hjbae@etri.re.kr)
1989년 부산대학교 전자계산학 (학사)
1991년 부산대학교 전자계산학 (석사)
1991년~현재 한국전자통신연구원 근무
인터넷서비스플랫폼연구실 실장
관심분야 서비스 플랫폼, 서비스 컴퓨팅, 클라우드 컴퓨팅, 미래 인터넷 서비스