

# 동적인 웹서비스들에 대한 QoS 기반 합성 기법 연구

강용혁\*

\*극동대학교 경영학부 정보경영전공  
e-mail:yhkang@kdu.ac.kr

## A Scheme of QoS-based Compositions of Dynamic Web Services

Yong-hyeog Kang\*

\*School of Business Administration, Far East University

### 요 약

분산 컴퓨팅의 새로운 패러다임인 웹서비스(web service)는 W3C에서 정의한 네트워크 상에서 컴퓨터들 간에 상호작용을 하기 위한 소프트웨어 시스템이다. 웹서비스는 웹서비스의 제공자와 사용자간의 상호 인터페이스를 통해 이용된다. 복잡한 웹서비스는 다른 웹서비스들의 합성으로 이루어질 수 있으며, 웹서비스의 합성에 대한 기법들은 많이 연구되었다. 본 논문에서는 웹서비스의 질적 수준을 높이기 위하여 서비스의 비기능적인 속성인 QoS를 고려한 웹서비스의 합성을 연구한다. 본 논문에서 제안하는 기법은 중계사이트로 하여금 웹서비스 합성 및 QoS 정보 그래프를 통해 최신의 웹서비스의 합성 및 QoS 상태 정보를 저장하도록 하며, 이를 통해 웹서비스의 사용자가 복잡한 웹서비스를 요청했을 때 QoS 상태 정보를 이용하여 동적이며 QoS를 고려한 웹서비스의 합성을 통해 보다 빠르고 보다 적합한 서비스를 제공할 수 있는 기법을 제안한다.

### 1. 서론

분산 컴퓨팅의 새로운 패러다임인 웹서비스(Web Service) 및 서비스 기반 구조(SOA: Service-Oriented Architecture)에 대한 연구는 지난 몇 년간 많이 진행되었다[1, 2]. 하지만 현재 웹서비스에 대한 산업계의 이용은 여러 가지 이유로 인해 미진한 상태로 있으며 그중 가장 큰 이유들로는 웹서비스의 보안 문제와 서비스의 질적 수준이 떨어지기 때문이다. 현재 보안 문제에 대한 연구는 표준화를 통하여 많은 진전이 이루어져 있지만, 웹서비스의 질적 수준을 높여주는 연구는 연구가 미진한 상태에 있다[3]. 웹서비스에 대한 QoS(Quality of Service)는 응답시간, 처리량, 가용성, 신뢰성 등과 같은 비 기능적인 속성들이다[4]. 본 논문에서는 웹서비스의 질적 수준을 높이기 위하여 QoS를 고려한 웹서비스의 합성을 통해 보다 빠르고 보다 정확하게 서비스를 제공할 수 있는 기법을 연구한다.

웹 기술은 현재 기계 중심의 웹 2.0에서 사용자 중심의 웹 3.0으로 발전해 나가고 있으며 그 중심 기술에 시맨틱 웹 기술이 있다[5, 6]. 시맨틱 웹 기술은 주로 RDFS와 OWL을 통한 웹 자원에 대한 표현과 웹 검색에 대한 연구가 활발히 이루어지고 있으나, 본 연구에서 수행할 QoS 까지 고려한 동적인 웹서비스의 합성에 대한 연구는 거의 없는 실정이다. 본 연구에서는 웹서비스 합성 정보와 QoS

정보를 위해 QoS 정보 그래프를 이용하여 동적인 웹서비스의 QoS 정보를 표현하고 변화하는 정보를 유지하여, 보다 빠르고 사용자의 요구에 맞는 웹서비스를 제공하는 기법을 연구한다.

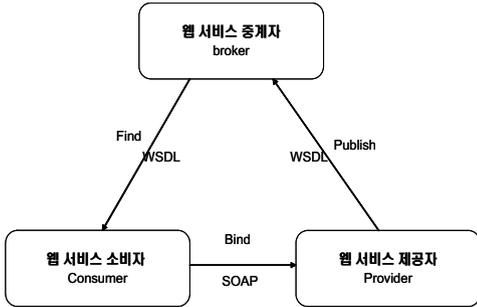
본 논문의 2장에서는 관련연구로 웹서비스 구조와 웹서비스의 합성에 대하여 설명하고 3장에서는 본 논문에서 제안하는 기법에 대한 설명과 제안기법의 성능에 대한 간략한 설명을 제시한다. 4장에서는 결론과 향후 연구 과제를 제시한다.

### 2. 관련연구

웹서비스(Web service)는 네트워크 상에서 서로 다른 종류의 컴퓨터들 간에 상호작용을 하기 위한 소프트웨어 시스템이다. 월드와이드웹(WWW)은 사람과 컴퓨터 간의 상호작용을 위한 시스템인데 반해, 웹서비스는 컴퓨터와 컴퓨터 간의 상호작용을 위한 시스템이다. 웹서비스 프로토콜 스택은 SOAP, WSDL, UDDI 등으로 이루어진다[1]. 이러한 웹서비스는 상호운용(interoperable)될 수 있는 기계들간의 대화(interaction)를 지원하기 위해 WSDL(Web Service Definition Language)로 기계가 처리할 수 있는 형식으로 인터페이스를 설명한다. 웹서비스의 클라이언트와 서버 시스템들은 일반적으로 HTTP 프로토콜을 이용하여 전송되는 SOAP(Simple Object Access Protocol) 메

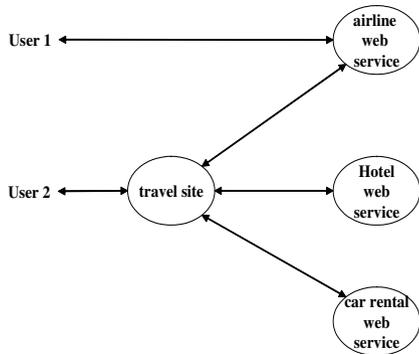
시지들을 이용하여 WSDL로 미리 정의된 웹서비스들과 상호작용을 한다.

웹서비스의 기본 구조는 <그림 1>과 같다[7]. 웹서비스 제공자가 제공하는 서비스를 웹서비스 소비자가 이용하는 데 웹서비스 중계자는 해당 서비스에 대한 인터페이스 정보를 저장하고 서비스의 검색에 사용된다.



<그림 1> 웹서비스의 기본 구조

웹서비스는 사용자와 웹서비스 제공자간의 상호인터페이스를 통해 이용된다. 한 개의 웹서비스만으로 충분한 경우에는 아래 그림의 User1처럼 직접 해당 웹서비스에 접근하여 해당 서비스를 이용하면 된다. 하지만 User2처럼 다양한 웹서비스들이 필요한 경우에는 웹서비스 합성을 중재해주는 travel site를 통해서 해당 서비스를 이용하는 것이 편리하며[8], 웹서비스를 중재해 주는 사이트 역시 웹서비스를 통합해서 제공하는 서비스 제공자 역할을 수행한다. 중재해주는 사이트에서 사용자에게 QoS 서비스를 제공하기 위해서는 합성되는 웹서비스들의 QoS 정보도 필요하게 된다.



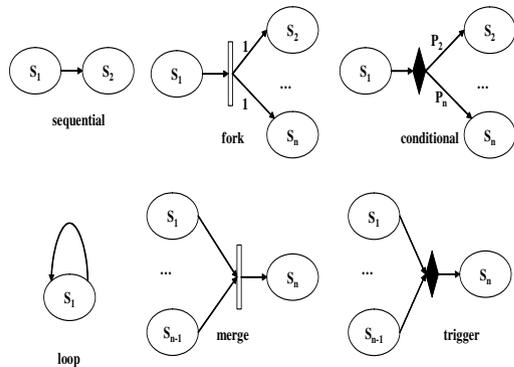
<그림 2> 웹서비스의 합성 예제

현재 웹서비스 표준은 QoS와 같은 서비스의 비기능적인 속성들을 표현하는 데 부족한 점이 있으며, 이러한 QoS 속성은 사용자의 요구를 맞추기 위해 서비스를 선택하는 데 중요한 요소가 된다[9]. QoS 속성 중에서 보안 (security)이나 상호 운영성(interoperability)은 어느 정도

연구가 진행되었으나 다른 분야는 더 많은 연구가 필요하다. 또한, 이러한 웹서비스들은 한 개의 서비스로만 제공되는 경우도 있지만, 특정 응용인 경우에는 여러 웹서비스들을 합성하여 제공되어야 한다. 웹서비스들을 합성할 경우 어떤 방식으로 혼합하는 것이 좋은지에 대한 연구도 여러 곳에서 진행되고 있다. 본 논문에서는 여러 웹서비스들을 합성하여 제공하는 방식을 연구하며, 서비스되는 웹서비스의 QoS 상태 정보를 중재사이트에서 저장하는 방식을 통해 서비스를 제공하는 기법을 연구한다.

### 3. 동적인 웹서비스에 대한 합성 기법

<그림 3>은 웹 서비스의 다양한 합성 패턴을 보여주는 그림이다[10]. 이러한 패턴들이 혼합되어서 복잡한 웹서비스들의 합성 정보를 표현한다. Sequential 패턴은 S<sub>1</sub> 서비스 후에 S<sub>2</sub> 서비스가 실행된다는 의미이며, fork 패턴은 S<sub>1</sub> 서비스는 S<sub>2</sub>에서 S<sub>n</sub>까지 서비스가 모두 실행된다는 의미이며, conditional 패턴은 S<sub>1</sub> 서비스 후에 S<sub>2</sub>에서 S<sub>n</sub>까지의 서비스가 P<sub>2</sub>에서 P<sub>n</sub>까지의 확률로 실행된다는 의미이다. Loop 패턴은 K번 S<sub>1</sub> 서비스가 반복된다는 의미이며, Merge 패턴은 S<sub>n</sub> 서비스가 실행되려면 S<sub>1</sub>에서 S<sub>n-1</sub>까지 서비스가 모두 실행되어야 한다는 의미이며, Trigger 패턴은 S<sub>1</sub>에서 S<sub>n-1</sub>까지 서비스 중에 하나의 서비스가 실행되면 S<sub>n</sub> 서비스가 실행될 수 있는 의미이다.



<그림 3> 웹서비스 합성 패턴들

이러한 웹서비스들의 합성 정보 및 QoS 정보는 1) 웹서비스를 요청하는 곳에서 유지할 수 있으며 2) 웹서비스를 중재하는 곳에서 유지할 수 있다. 본 연구에서는 웹서비스를 중재하는 곳에서 유지하는 기법을 사용한다.

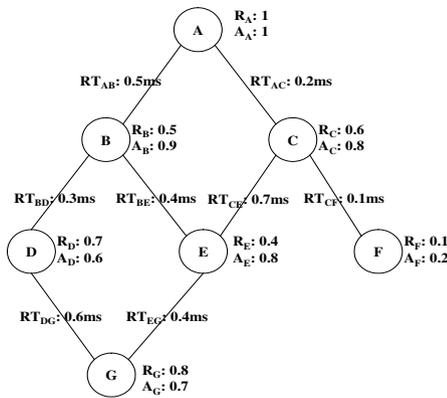
웹서비스들의 합성 정보 및 QoS 정보를 저장하기 위해 표현하는 방식을 고려하면 다음과 같은 세 가지 방식을 고려할 수 있다.

- 1) 온톨로지를 사용하여 표현
- 2) XML을 통하여 WSDL에 해당 정보를 유지
- 3) 시스템 내에서 그래프 형식으로 유지

온톨로지를 사용하는 방식이 최신의 기술을 이용하여 웹

서비스 합성 및 QoS 정보를 표현하는 방식으로 검색 및 추론에 의해 자동으로 검색하고 합성될 수 있는 장점이 있다. 그러나, 텍스트로 표현되어 있기 때문에 파서와 같은 전처리가 필요하여 속도가 느려질 수 있으며, 온톨로지로 표현한 정보를 변환 및 추론하는 데에 속도가 느려질 수 있으므로, QoS 정보를 처리하는 데에는 속도를 높일 수 있는 기술이 필요하다. XML을 통해서 표현하는 방식도 파서와 같은 전처리가 필요하며 동적인 상태 정보를 유지하기 위해서는 잦은 정보 갱신이 필요하여 속도가 느려질 수 있다. 본 논문에서는 웹서비스 합성 및 QoS 상태 정보를 XML을 통하여 WSDL에 그 정보를 유지하여 검색 및 인터페이스에 이용되게 하며, 해당 정보들을 중재시스템들이 수집하여 합성 및 QoS 상태 그래프로 표현하여 중재시스템에 유지함으로써 빠른 이용과 자동화가 가능하도록 한다.

본 논문에서는 웹서비스 합성 정보 및 QoS 정보는 다음과 같은 그래프를 통해 유지되고 관리된다. 이러한 웹서비스 합성 및 QoS 정보 그래프는 각각의 중재 시스템에서 유지 관리되며 주기적으로 갱신되어 최신의 정보를 유지하게 된다. 아래 그래프는 A 사이트에서 관리되는 웹서비스 합성 및 QoS 정보 그래프이다. 그래프의 노드에 신뢰성처럼 QoS 정보가 있는 경우도 있으며 그래프의 선분(edge)에도 응답시간(response time)처럼 QoS 정보가 유지될 수 있다.



<그림 4> 웹서비스 합성 및 QoS 정보 그래프

이러한 웹서비스 합성 및 QoS 정보들은 사이트들 사이에 정보 교환을 통해 최신의 정보를 유지한다. 즉, 이러한 정보를 이웃 사이트에 송수신함으로써 최신의 정보를 유지한다. 정보를 전송하는 방식에는 1) 주기적으로 이웃 사이트들 간의 정보교환하는 방식과 2) 변경된 정보가 있을 경우에만 정보를 전송하는 방식이 있으며, 3) 이 두가지 방식을 혼합한 방식이 있을 수 있다. 본 논문에서는 주기적으로 이웃 사이트에 정보를 교환하는 것을 기본으로 하여 주요 정보가 변경될 경우 QoS 정보들을 전송하는 혼

합방식을 사용한다.

QoS를 고려한 웹서비스 합성에 대한 기존 연구에서는 대부분의 기법[4, 11]이 전역적인 정보를 가진 상태에서 다양한 QoS를 고려하여 최적과 근접한 QoS 서비스를 제공하고 있다. 이러한 기법들의 문제점들을 나열하면 다음과 같다.

- 1) 서비스 제공 시간이 늦어짐
- 2) 다양한 형태의 웹서비스를 통합하는 데 어려움
- 3) 온톨로지를 이용한 웹서비스인 경우 웹서비스의 합성이 어려움
- 4) 제약조건을 기반으로 하므로 최적의 QoS 서비스를 제공하기 어려움

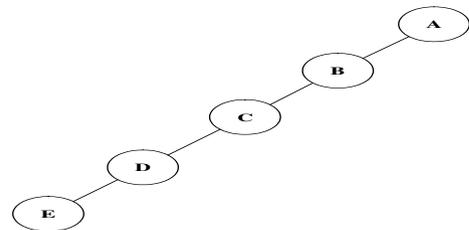
본 논문에서 제안하는 기법이 기존기법에 비해 가지는 장점을 나열하면 다음과 같다.

- 1) 빠른 QoS 웹 서비스 검색
- 2) 최적의 QoS를 갖는 웹서비스 검색
- 3) 추론 및 변환 기법을 이용할 경우 다양한 형태의 웹서비스를 지원할 수 있는 확장가능성

본 논문에서 제안하는 기법은 QoS 상태정보를 유지하지 않는 다른 기법에 비해 웹서비스 합성에 대한 검색시간이 효율적이다. 한 사이트에서 웹서비스 합성 정보를 유지하기 때문에 바로 검색할 수 있으며, 최적의 QoS를 갖는 웹서비스도 검색이 가능하다.

본 논문에서 제안하는 기법의 장점을 보장하기 위해서는 중재시스템에서 최신의 정보를 정확하게 유지하는 것이 필요하다. 이를 위해 각각의 중재 시스템들이 웹서비스 제공자로부터 최신 정보를 얻어 와야 한다. 이를 위해 중재 시스템들이 주기적으로 웹서비스 제공자에게 최신의 정보를 요구하여 수행하거나 웹서비스 제공자에게 주기적으로 최신의 정보를 전송하도록 한다.

중재시스템에서 유지되고 있는 정보가 최신의 정보를 유지하지 못해 사용자의 요구에 맞추지 못할 경우를 고려하면 다음과 같이 평가할 수 있다. 우선 최신의 정보를 유지하지 못할 확률을  $UP_i$ 라 하고 최신의 정보를 유지하지 못하더라도 영향을 받지 않을 확률을  $EP_i$ 라 할 경우 그 확률이 동시에 발생할 경우에만 중재시스템이 지원하는 서비스가 실패된다. 다음 <그림 5>과 같이 최악의 경우 A, B, C, D와 E 서비스가 필요한 경우에 중재 시스템이 실패할 확률은 다음과 같다.



<그림 5> 웹서비스 합성 그래프의 예

$$P(\text{Fail}_{UP,EP}) = 1 - \prod_{x=1}^4 ((1 - UP_i) * (1 - EP_i))^x$$

UP<sub>1</sub>와 EP<sub>1</sub>의 값들이 각 노드에서 같다고 가정하고 UP<sub>1</sub>를 0.1로 EP<sub>1</sub>를 0.1로 할 경우하고 앞의 식은 다음과 같이 계산된다.

$$P(\text{Fail}_{0.1,0.1}) = 1 - 0.99^{\sum_{i=1}^4 1} = 1 - 0.99^{10} = 1 - 0.90 = 0.1$$

즉, 10번 중에 1번만이 실패하고 9번은 성공하는 확률을 보여줄 수 있다. 하지만, UP<sub>1</sub>를 0.1로 하고 EP<sub>1</sub>의 값을 0.5로 할 경우 실패확률은 급속히 늘어난다.

$$P(\text{Fail}_{0.1,0.5}) = 1 - 0.95^{\sum_{i=1}^4 1} = 1 - 0.95^{10} = 1 - 0.60 = 0.4$$

이 경우에는 10번 중에 4번은 실패하는 경우를 보여주며 이 경우에는 시스템의 성능이 급속히 저하될 수 있다.

UP<sub>1</sub>의 값은 시스템 운용을 통해 0.1미만으로 줄일 수 있지만, EP<sub>1</sub>의 값은 QoS 특성에 따라 달라진다. 응답시간 같이 QoS 값이 크게 변화하지 않는 경우에는 그 값을 줄일 수 있지만, 가용성같은 QoS 정보는 임의의 자원이 소비되었을 경우에 EP<sub>1</sub>의 값은 크게 영향을 받는다. 이러한 정보는 UP<sub>1</sub>의 값을 줄이거나 그래프의 크기를 축약하여 실패 확률을 줄이는 기법이 필요하다.

#### 4. 결론 및 향후 연구 계획

본 논문에서는 웹서비스의 비기능적인 속성 중에 하나인 QoS를 고려한 웹서비스 합성에 대하여 연구하였다. 제안기법은 중계사이트를 이용하여 웹서비스의 합성 및 QoS 정보를 유지하여 사용자가 복잡한 웹서비스를 요청했을 때 효율적인 검색과 최적의 검색을 수행할 수 있는 기법이다. 간단한 성능 분석을 통해 제안기법의 효율성과 제안기법의 장점이 효과적으로 이루어질 수 있는 환경에 대한 내용을 제시하였다.

향후 연구 계획으로는 우선 제안기법에 대한 성능 평가를 확장하고 시뮬레이션을 통한 성능 평가를 통해 본 제안기법의 효율성을 검증할 것이며, 두 번째 향후 연구계획으로는 시맨틱 웹[12, 13]과 연동하는 방식을 연구하여 제안기법에서 제시한 QoS 정보를 RDF나 OWL로 상호 변환하는 방식을 연구하고 추론 및 자동화에 대한 연구도 진행할 것이다.

#### 참고문헌

[1] Cary Pennington et al, "Introduction to Web Services," in Semantic Web Services: Theory, Tools and Applications, J. Cardoso, editor, Idea Group Publishing, Hershey, PA, pp. 134-154, 2007.  
[2] 이규철, "웹서비스 기술 동향," Journal of the Research Institute of Industrial Technology, 2005.

[3] Liam Q'Brien et al, "Quality Attributes for Service-Oriented Architectures," SDOA'07, 2007.

[4] Tao Yu, et al, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints," ACM Transactions on the Web, 2007.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. "The Semantic Web," Scientific American, May, 2001.

[6] Lassila, O. and Hendler, J, "Embracing Web 3.0," IEEE Internet Computing, 2007.

[7] k. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to Web services architecture," IBM System Journal, July, 2002.

[8] Daniel A. Menascé, "QoS Issues in Web Services," IEEE Internet Computing, 2002.

[9] Daniel A. Menascé, "Response-Time Analysis of Composite Web Services," IEEE Internet Computing, 2004.

[10] Daniel A. Menascé, "Composing Web Services: A QoS View," IEEE Internet Computing, 2004.

[11] Sheng, Quan Z. et al, "SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment," Proceedings of the 28th VLDB Conference, 2002.

[12] E. Michael Maximilien and Munindar P. Singh, "A Framework and Ontology for Dynamic Web Services Selection," IEEE Internet Computing, 2004.

[13] Siobhán Clarke, "Current Solutions for Web Service Composition," IEEE Internet Computing, 2004.