

기능적 의미에 기반한 복합 웹 서비스 자동 구성

신동훈[†], 이경호^{**}

요 약

최근 들어 복합 웹 서비스를 자동으로 구성하기 위한 많은 연구들이 진행되었다. 이들 연구의 대부분은 서비스의 기능을 고려하지 않고 단순히 웹 서비스들의 입력과 출력을 연결하여 복합 웹 서비스를 구성한다. 그러므로 사용자의 의도에 부합하지 않는 복합 웹 서비스를 생성할 수 있다. 또한 가용한 웹 서비스의 모든 조합을 고려하기 때문에 시간 복잡도가 매우 크다. 이러한 문제를 해결하기 위해, 본 논문에서는 웹 서비스의 기능적 의미를 명시적으로 기술하고 이를 기반으로 복합 웹 서비스를 자동으로 구성하는 방법을 제안한다. 제안된 방법은 서비스 간의 입, 출력 의존성 및 기능 정보를 그래프 구조를 사용하여 조직화한다. 그리고 그래프 구조에서 사용자가 요구하는 기능을 제공하는 핵심 서비스와 입,출력 타입 간의 변환을 지원하는 부가 서비스를 찾아 이들 간의 조합으로 복합 웹 서비스를 구성한다. 제안된 방법은 웹 서비스의 기능을 고려함으로써 구성된 복합 웹 서비스의 의미적 정확성을 높이고, 의미적으로 연관성이 있는 서비스들 간의 조합만을 고려함으로써 시간 복잡도를 줄인다.

Automated Generation of Composite Web Services based on Functional Semantics

Dong-Hoon Shin[†], Kyong-Ho Lee^{**}

ABSTRACT

Recently, many studies on automated generation of composite Web services have been done. Most of these works compose Web services by chaining their inputs and outputs, but do not consider the functional semantics. Therefore, they may construct unsatisfied composite services against users' intention. Furthermore, they have high time-complexity since every possible combinations of available services should be considered. To resolve these problems, this paper proposes a sophisticated composition method that explicitly specifies and uses the functional semantics of Web services. Specifically, A graph model is constructed to represent the functional semantics of Web services as well as the dependency among inputs and outputs. On the graph, we search core services which provide the requested functionality and additional services which transform between I/O types of the user request and the core services. Then, composite services are built from combinations of the discovered services. The proposed method improves the semantic correctness of composite services by the functional semantics of Web services, and reduces the time complexity by combinations of functionally related services.

Key words: Web Services(웹 서비스), Composition(조합), Functional Semantics(기능적 의미)

※ 교신저자(Corresponding Author): 이경호, 주소: 서울 특별시 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: 2008년 7월 15일

[†] 준회원, 연세대학교 컴퓨터과학과

(E-mail: dhshin@icl.yonsei.ac.kr)

^{**} 중신회원, 연세대학교 컴퓨터과학과

※ 이 논문은 2006년도 교육과학기술부(MEST)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2006-000-100864-0)

1. 서 론

최근 들어 웹 서비스가 다양한 분야에서 널리 사용되면서 고수준의 비즈니스 프로세스를 지원하는 복합 웹 서비스 구성에 대한 요구가 증가하고 있다. 복합 웹 서비스는 다수의 웹 서비스들을 조합하여 단일 웹 서비스로는 제공하기 힘든 복잡한 기능을 제공할 수 있다는 장점 때문에 많은 관심을 받고 있다. 그러나 수많은 웹 서비스들 중에서 필요한 웹 서비스들을 찾고 이들을 수동으로 조합하여 복합 웹 서비스를 구성하는 일은 매우 어렵고 시간이 오래 걸리는 작업이다. 또한 가용한 웹 서비스들이 엄청난 속도로 늘어나고 있기 때문에 수동으로 복합 웹 서비스를 구성하는 작업은 점점 더 어려운 일이 되고 있다.

이러한 복합 웹 서비스 수동 구성의 어려움을 해결하기 위해 사용자의 요구사항으로부터 자동으로 복합 웹 서비스를 구성하는 많은 방법들이 제안되었다[1]. 기존 연구들은 서비스 제공자에 의해 제공되는 각각의 웹 서비스를 OWL [2] 과 같은 온톨로지 언어로 생성된 도메인 온톨로지에 기반하여, OWL-S [3], WSMO [4], SAWSDL [5] 과 같이 웹 서비스의 의미적 정보를 기술할 수 있는 웹 서비스 명세를 통해 기술한다. 또한 사용자 요구사항을 입력, 출력, 전제조건, 효과로 기술하며, 이렇게 기술된 사용자 요구사항과 서비스 명세에 대해 다양한 인공지능 기법을 적용하여 복합 웹 서비스를 자동으로 구성한다.

기존 방법들은 선행하는 서비스의 출력을 입력으로 받아들이는 서비스를 찾고 이들을 서로 연결하는 과정을 반복적으로 수행하여 사용자가 제공하는 입력으로부터 요구되는 출력을 반환하는 서비스의 조합을 생성한다. 이 때, 연결되는 두 웹 서비스에 대해 선행 서비스의 효과는 연결되는 서비스의 전제조건을 만족시켜야만 한다. 일반적으로 입력, 출력, 전제조건, 효과가 모두 동일한 서비스는 같은 기능을 하는 서비스로 간주될 수 있다. 따라서 사용자가 제공하는 입력으로부터 요구되는 출력을 반환하고 전제조건과 효과를 만족시키는 복합 웹 서비스는 사용자의 요구사항을 만족시키는 서비스라 할 수 있다. 그러나 개별 웹 서비스나 사용자 요구사항이 전제조건과 효과를 갖지 않는 경우, 기존의 방법을 통해 생성된 복합 웹 서비스는 그림 1과 같이 사용자 요구사항

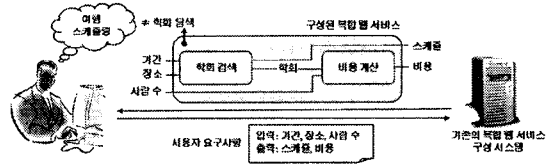


그림 1. 사용자의 의도에 부합되지 않는 복합 웹 서비스 구성의 예

을 만족시킴에도 불구하고 사용자의 의도와는 전혀 다른 기능을 제공하는 서비스가 될 수 있다.

그림 1은 사용자가 기간, 장소, 사람 수를 입력으로 제공하고 여행 계획과 소요되는 비용을 출력으로 요구하는 경우 기존의 복합 웹 서비스 자동 구성 시스템이 해당 기간에 개최되는 회의의 일정과 참가비용을 계산하는 서비스를 생성하는 예이다. 일반적으로 사용자에게 정보를 제공하는 정보 서비스는 전제조건이 필요치 않으며 서비스가 실행된 후 아무런 효과를 갖지 않는다. 따라서 사용자가 이러한 정보 서비스들을 조합한 복합 웹 서비스를 요구하는 경우 기존 연구는 앞의 예에서와 같이 생성되는 복합 웹 서비스의 의미적 정확성을 보장하지 못한다. 또한 기존 연구는 복합 웹 서비스를 생성하기 위해 연결 가능한 모든 웹 서비스의 조합을 고려하기 때문에, 복합 웹 서비스 구성 과정에 소요되는 시간이 가용한 서비스의 수에 지수승으로 비례하는 매우 큰 시간 복잡도를 갖는다.

본 논문에서는 이러한 문제점을 해결하기 위해 웹 서비스의 기능적 의미를 명시적으로 기술하고 이 정보를 복합 웹 서비스 구성 과정에서 활용하는 방법을 제안한다. 제안된 방법은 입, 출력 정보뿐만 아니라 기능적 의미를 포함하여 웹 서비스를 기술하고 이를 그래프 구조를 사용하여 저장한다. 그리고 주어진 사용자 요구사항에 대해 제안된 그래프 탐색 방법을 사용하여 복합 웹 서비스를 구성한다. 제안된 방법은 복합 웹 서비스 구성 과정에서 웹 서비스의 기능을 고려함으로써 구성된 복합 웹 서비스의 의미적 정확성을 높이고, 의미적으로 연관성이 있는 서비스들 간의 조합만을 고려함으로써 시간 복잡도를 줄일 수 있다.

본 절 이후의 논문 구성은 다음과 같다. 2절에서는 복합 웹 서비스 자동 구성에 관한 기존 연구들의 특징과 문제점을 간략히 소개한다. 3절에서는 제안된 복합 웹 서비스 구성 방법에 대해 기능적 의미를 포

합한 웹 서비스 기술, 기능 연결 그래프와 핵심 서비스 그리고 복합 웹 서비스 구성 알고리즘, 시간 복잡도의 네 부분으로 나누어 자세히 설명한다. 4절에서는 실험을 통해 제안된 방법의 성능을 평가한다. 끝으로 5절에서는 결론 및 향후 연구 방향에 대하여 기술한다.

2. 관련연구

일반적으로 복합 웹 서비스는 그림 2와 같이 사용자가 제시한 전제조건을 만족시키고 사용자 입력을 받아들이는 초기 상태 S_1 으로부터, 사용자가 요구하는 효과와 출력을 제공하는 임의의 상태 S_n 으로의 상태 전이 시스템(State Transition System)에 대응될 수 있다. 이때 상태 전이를 위해 필요한 행위는 임의의 웹 서비스에 대응되며 이전 상태의 출력과 효과가 전이를 위해 필요한 행위(웹 서비스)의 입력과 전제조건을 만족시켜야 한다. 웹 서비스 자동 구성에 관한 대부분의 기존 연구들은 복합 웹 서비스 구성의 문제를 전술한 상태 전이 시스템을 찾는 문제로 바꾸어 해결하고자 한다. 이를 위해 HTN 플래닝, Contingency 플래닝, Linear Logic Theorem Proving 등의 다양한 인공지능 기술을 사용한다. [6-11]

인공지능 접근법에 기반한 기존 연구들은 사용자의 요구에 부합하는 상태 전이 시스템을 찾고 각각의 상태 전이를 위해 필요한 웹 서비스들을 연결하여 복합 웹 서비스를 구성한다. 이 과정에서 온톨로지를 사용한 입출력 매개변수 매칭과 전제조건과 효과에 대한 동등성(Equality) 평가가 이루어진다. Rama Akkiraju 등 [6]은 기존의 매개변수 매칭 방법 대신 도메인 의존적, 독립적 정보를 모두 사용하는 시맨틱 매칭을 사용하고 플래닝 과정에서 의미적 모호성을 고려하여 생성되는 복합 웹 서비스의 의미적 정확성을 향상시키고자 하였다. 또한 Agarwal 등 [7]은

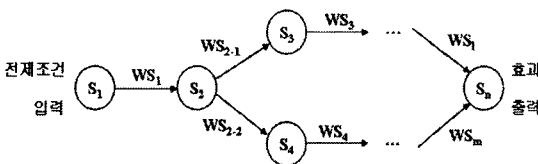


그림 2. 상태 전이 시스템으로 표현된 복합 웹 서비스

웹 서비스를 인터페이스에 따라 분류하고 논리적 조합과 물리적 조합의 두 단계를 통해 복합 웹 서비스를 구성하고, 논리적 조합 단계에서 Contingency 플래닝을 적용하여 조건부 분기를 포함하는 복합 웹 서비스의 구성을 지원하였다.

그러나 Rama kkiraju 등 과 Agarwal 등의 연구는 가용한 모든 서비스들이 전제조건과 효과를 갖는다고 가정한다. 따라서 전제조건과 효과를 갖지 않는 정보 서비스에 대해 입력과 출력 매개변수의 매칭만을 통해 복합 웹 서비스를 구성할 수밖에 없다. 이는 상태 전이 시스템이 상태 전이를 유발하는 행위 자체가 아니라 행위를 일으키기 위한 조건과 일어난 후의 상태에만 관심이 있기 때문이다. 일반적으로 전제조건과 효과가 없는 서비스의 경우 같은 타입의 입력과 출력을 갖지만 서로 다른 기능을 제공하는 서비스가 존재한다. 그러므로 사용자의 의도에 부합하는 복합 웹 서비스를 구성하기 위해서는 서비스 자체의 기능을 명시적으로 기술하고 이를 복합 웹 서비스 구성 과정에서 반영할 수 있는 방법이 필요하다.

한편 Sirin 등 [8]은 서비스 도메인에서 미리 정의된 HTN을 사용하여 복합 웹 서비스 구성 시에 서비스의 기능을 고려하고 시간 복잡도 문제를 개선하는 방법을 제안한다. 그러나 HTN은 도메인 내에 존재하는 모든 서비스들이 어떻게 연결되고, 어떤 방법으로 조합되며 분할될 수 있는지에 대한 모든 정보를 정의해야만 한다. 일반적으로 레지스트리에 존재하는 모든 서비스들의 기능과 이들 간의 조합 및 분할 여부를 사전에 파악하여 HTN을 구성하는 것은 불가능한 일이다. 이 연구는 규모가 큰 특정 복합 웹 서비스에 대해 주어진 입력을 처리할 수 있는 가장 적절한 실행 경로를 찾는 데 의미가 있다. 그러나 사용자의 요구사항으로부터 다수의 개별 웹 서비스를 조합하여 기존에 존재하지 않는 새로운 기능을 제공하는 일반적인 의미의 복합 웹 서비스 구성에는 적합하지 않다.

Hashemian과 Mavaddat [9]은 기존 연구들이 사용자의 요구사항을 만족시키는 정확한 복합 웹 서비스 구성에 초점을 맞춘 것과 달리 복합 웹 서비스 구성 과정의 시간 복잡도에 주목하였다. 인공지능 기법에 기반한 기존 연구들은 모든 상태에 대해 현재 상태에서 적용 가능한 모든 행위(웹 서비스)를 고려한다. 따라서 복합 웹 서비스 구성 과정에 소요되는 시간이 가용한 서비스의 수에 대해 지수승으로 비례

하는 지수시간 복잡도를 갖는다. 이에 비해 Hashemian과 Mavaddat의 방법은 서비스 저장소에 등록된 모든 웹 서비스의 의존 관계를 제안된 의존 그래프 형태로 저장하고 그래프 탐색 방법을 이용하여 복합 웹 서비스를 구성한다. 의존 그래프 상에서 각각의 입출력 매개변수는 정점으로 표현되며 가용한 서비스들은 서비스의 입력에 해당하는 매개변수 정점으로부터 출력에 해당하는 매개변수 정점으로서의 간선 위에 레이블로 표현된다.

제안된 의존 그래프를 사용하여 Hashemian과 Mavaddat의 방법은 복합 웹 서비스 구성 과정에서 서로 연결 가능한 서비스만을 탐색할 수 있고, 이로 인해 기존의 인공지능 기반 방법들에 비해 상대적으로 작은 시간 복잡도를 갖는다. 그러나 Hashemian과 Mavaddat의 방법 역시 서비스 자체의 기능을 고려하지 않고 단순히 입출력 매개변수의 타입만을 고려하기 때문에 생성된 복합 웹 서비스가 사용자가 요구하는 서비스와 기능적 의미에서 동일함을 보장하지 못한다.

Ye와 Zhang [12]은 웹 서비스 탐색의 정확성을 향상시키기 위하여 웹 서비스 자체의 기능을 명시적으로 기술하는 방법을 제안하였다. Ye와 Zhang은 대상과 행위 및 제약사항으로 웹 서비스 및 사용자 요구사항을 기술하고 이를 사용하여 단일 웹 서비스를 정확하게 탐색할 수 있는 방법을 제안하였다. 그러나 복합 웹 서비스 구성 과정에서 이러한 웹 서비스 기능을 적용할 수 있는 방법은 제공하지 않는다.

본 논문에서는 전제조건과 효과가 존재하지 않는 정보 서비스에 대해 기능적 의미의 정확성을 보장할 수 있는 복합 웹 서비스 구성 방법을 제안한다. 또한 Hashemian과 Mavaddat의 방법에서와 유사하게 사용자가 요구하는 기능과 의미적으로 연관성이 없는 서비스를 탐색 과정에서 제외함으로써 시간 복잡도를 향상시킬 수 있는 방법을 제안한다. 이를 위해 Ye와 Zhang의 방법에서와 같이 서비스의 기능적 의미를 대상과 행위를 사용하여 기술하고, Hashemian과 Mavaddat가 제안한 의존 그래프를 서비스의 기능적 의미를 포함할 수 있도록 확장한다.

3. 제안된 복합 웹 서비스 구성 방법

제안된 복합 웹 서비스 구성 방법은 그림 3과 같이

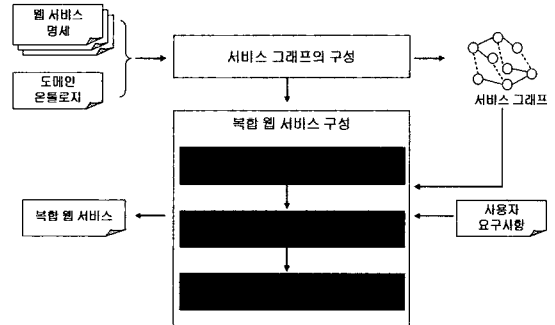


그림 3. 제안된 복합 웹 서비스 구성 방법

후보 서비스 집합 구성과 서비스 그래프 탐색 그리고 웹 서비스 조합의 세 단계로 이루어진다.

본 절에서는 도메인 온톨로지를 데이터 온톨로지 와 기능 온톨로지 로 구분하여 정의하고, 이를 이용하여 웹 서비스 및 사용자의 요구사항을 기능적 의미를 포함하여 기술할 수 있는 방법을 제안한다. 또한 웹 서비스 기술 정보를 기반으로 제안된 그래프 모델을 구성하고 그래프 탐색 방법을 사용하여 사용자 요구 사항으로부터 복합 웹 서비스를 구성하는 방법을 예를 통해 자세히 설명한다. 마지막으로, 제안된 방법과 Hashemian과 Mavaddat의 방법의 시간 복잡도 분석을 통해 제안된 방법이 기존 방법들에 비해 빠른 시간 안에 복합 웹 서비스를 구성할 수 있음을 보인다.

3.1 기능적 의미를 포함한 웹 서비스의 기술

일반적으로 사용자에게 정보를 제공하는 많은 서비스들이 정보 서비스에 해당하며 이러한 서비스들은 기존의 연구에서 전제조건과 효과 없이 입력과 출력만으로 기술된다. 그러나 서론에서 언급한 바와 같이 입력과 출력만으로는 웹 서비스 자체의 기능을 온전히 표현할 수 없다. 두 정보 서비스가 서로 입력과 출력이 같다고 해도 제공하는 기능이 서로 다를 수 있다. 따라서 사용자가 요구하는 서비스가 정보 서비스만으로 구성되는 경우 기존 방법을 통해 생성되는 복합 서비스는 사용자가 요구하는 기능을 제공함을 보장하지 못한다. 그림 1의 예에서와 같이 입력과 출력만을 연결하여 복합 웹 서비스를 구성할 경우 구성된 서비스는 사용자의 의도와는 전혀 다른 기능을 하는 서비스가 될 수 있다.

본 논문에서는 이러한 문제를 해결하기 위하여 Ye와 Zhang의 방법에서와 같이 서비스의 기능을 명

시적으로 기술하는 방법을 사용한다. 제안된 방법에서 웹 서비스의 기능¹⁾은 다음과 같이 웹 서비스가 실제로 수행하는 행위와 행위 대상의 조합으로 표현된다. 예를 들어, 거리를 계산하는 기능을 제공하는 웹 서비스의 기능은 {계산하다, 거리}와 같이 표현될 수 있다.

○ 웹 서비스의 기능 = {행위, 대상}

또한 웹 서비스는 다음과 같이 웹 서비스의 기능, 입력 매개변수 집합 그리고 출력 매개변수의 집합으로 기술된다. 예를 들어, 두 도시 사이의 거리를 계산하여 Km 단위로 반환하는 기능을 제공하는 웹 서비스 C는 $C = \{ \{계산하다, 거리\}, \{도시, 도시\}, \{거리_Km\} \}$ 로 표현될 수 있다.

○ 웹 서비스(WS) = (기능, 입력 매개변수 집합, 출력 매개변수 집합)

본 논문에서는 웹 서비스를 기술하기 위해 데이터 온톨로지와 기능 온톨로지²⁾로 분리하여 기술된 도메인 온톨로지를 사용한다. 데이터 온톨로지는 도메인 내에서 사용되는 입, 출력 매개변수 및 대상을 기술하는데 사용되는 개념과 이들 간의 관계를 정의한다. 그리고 기능 온톨로지는 도메인 내에서 웹 서비스가 수행하는 기능을 기술하는데 사용되는 개념과 이들 간의 관계를 정의한다. 웹 서비스의 입력 및 출력 집합의 원소들과 기능 집합의 대상은 데이터 온톨로지 상의 개념과 대응되고, 기능 집합의 행위는 기능 온톨로지 상의 개념과 대응된다.

한편, 기능 온톨로지는 데이터 온톨로지의 개념과 결합되어 다양한 기능을 표현하고 복합 웹 서비스 구성 과정을 용이하게 하기 위해 다음과 같은 제약사항을 만족시키도록 구성한다.

- ① 기능 온톨로지의 개념은 일반적인 행위를 표현하는 동사를 사용한다.
- ② 하나의 기능은 온톨로지 상에서 하나의 개념으로만 정의한다.
- ③ 상위 개념이 표현하는 기능은 하나 이상의 자식 개념들의 조합으로만 제공 가능하다.

1) 실제 웹 서비스는 서로 다른 기능을 하는 다수의 오퍼레이션을 가질 수 있으며 이들의 집합으로 기술되어야 한다. 그러나 본 논문에서는 편의를 위해 모든 웹 서비스가 하나의 오퍼레이션을 가진다고 가정한다.

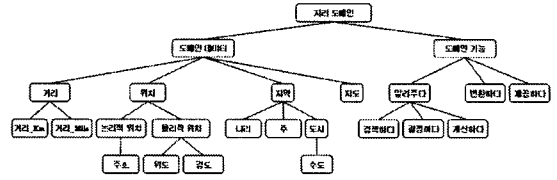


그림 4. 도메인 온톨로지의 예

그림 4는 이러한 제약사항을 만족시키는 기능 온톨로지와 데이터 온톨로지³⁾로 구성된 도메인 온톨로지의 예이다.

3.2 기능 연결 그래프와 핵심 서비스

복합 웹 서비스 구성을 용이하게 하기 위해 웹 서비스에 대한 기술은 제안된 서비스 그래프 형태로 구조화되어 저장된다. 서비스 그래프는 가용한 서비스와 도메인 데이터 개념 간의 관계를 표현하는 데이터 의존 그래프와 도메인 행위 개념 간의 관계를 표현하는 행위 연결 그래프 그리고 이들 두 그래프 간의 대응으로 구성된다. 그림 5는 제안된 웹 서비스 기술 방법으로 기술된 가용한 웹 서비스들의 예이고, 그림 6은 그림 5의 서비스들로 구성된 서비스 그래프이다.

데이터 의존 그래프는 Hashemian과 Mavaddat가 제안한 의존 그래프와 유사하게 도메인에서 가용

- $C_1 = \{ \{알려주다, 수도\}, \{나라\}, \{수도\} \}$
- $C_2 = \{ \{계산하다, 거리\}, \{도시, 도시\}, \{거리_Km\} \}$
- $C_3 = \{ \{결정하다, 물리적 위치\}, \{주, 도시\}, \{위도, 경도\} \}$
- $C_4 = \{ \{변환하다, 거리\}, \{거리_Mile\}, \{거리_Km\} \}$
- $C_5 = \{ \{변환하다, 거리\}, \{거리_Km\}, \{거리_Mile\} \}$
- $C_6 = \{ \{검색하다, 논리적 위치\}, \{전화번호\}, \{주소\} \}$
- $C_7 = \{ \{제공하다, 지도\}, \{도시\}, \{지도\} \}$

그림 5. 지리 도메인에서 가용한 서비스들의 예

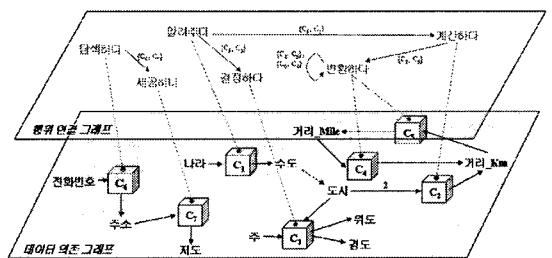


그림 6. 그림 5의 서비스들로 구성된 서비스 그래프

한 웹 서비스들과 이들의 입력 개념과 출력 개념을 연결하여 구성된다. 그러나 의존 그래프와는 다르게 웹 서비스와 도메인 데이터 온톨로지의 개념이 모두 정점이 된다. 각각의 서비스 정점은 입력 개념 정점과 들어오는 간선으로, 출력 개념 정점과 나가는 간선으로 연결된다. 그림 6의 C₂와 같이 한 서비스가 두 개 이상의 동일한 개념을 입력이나 출력으로 가지는 경우, 그 개수를 간선의 레이블로 사용한다. 또한 그림 6의 도시와 수도 같이 데이터 온톨로지 상에서 부모 자식 관계에 있는 개념 정점들에 대해 자식 개념 정점으로부터 부모 개념 정점으로의 간선을 생성한다.

행위 연결 그래프는 행위 온톨로지 상의 행위 개념, 중 웹 서비스의 기술에 사용된 개념들로 구성된다. 데이터 의존 그래프에서 모든 서비스 정점은 자신이 제공하는 행위에 해당하는 행위 개념과 대응된다. 데이터 의존 그래프 상에서 이웃하는 두 서비스 정점 사이에 간선이 존재하면 두 서비스와 대응되는 행위 개념 사이에 두 서비스 정점 간의 간선과 같은 방향의 간선을 생성한다. 그리고 두 서비스 정점을 원소로 하는 집합을 생성하고, 이를 행위 연결 그래프에 새로 생성된 간선의 레이블에 추가한다.

한편, 제안된 방법은 복합 웹 서비스를 구성하는 웹 서비스들을 핵심 서비스와 부가 서비스로 나누고 복합 웹 서비스 구성 과정에서 이들을 나누어 탐색함으로써 시간 복잡도를 줄인다. 핵심 서비스와 부가 서비스는 사용자가 요구하는 기능에 따라 실행 시간에 동적으로 결정된다. 핵심 서비스는 사용자가 요구하는 기능, 즉 대상과 행위를 만족시키기 위해 반드시 필요한 서비스를 말한다. 이와 달리 부가 서비스는 사용자가 요구하는 기능을 직접적으로 구현하는 서비스는 아니지만 핵심 서비스와 사용자 요구사항 간의 입, 출력을 일치시키기 위해 필요한 서비스를 말한다.

예를 들어, 사용자가 두 개의 도시를 입력으로 받아들이며 두 도시 간의 거리를 마일 단위로 반환하는 서비스를 요청한다고 가정하자. 사용자의 요구사항은 {{계산하다, 거리}, {도시, 도시}, {거리_Mile}}로 표현될 수 있다. 그림 6에서 이러한 기능을 제공하는 단일 웹 서비스는 존재하지 않지만 C₂와 C₀를 연결하면 요구되는 기능을 제공할 수 있다. 이 때, 사용자가 요구하는 {계산하다, 거리}의 기능을 수행하는 C₂가

핵심 서비스, 데이터 타입 간의 변환을 제공하는 C₀가 부가 서비스가 된다.

3.3 복합 웹 서비스 구성 알고리즘

제안된 복합 웹 서비스 구성 방법은 후보 서비스 집합 구성, 행위 연결 그래프 탐색, 웹 서비스 조합의 세 단계로 나누어진다. 후보 서비스 집합 구성 단계에서는 복합 웹 서비스의 구성원으로 참여할 가능성이 있는 웹 서비스들이 탐색된다. 사용자가 제공하는 입력 중 적어도 하나의 입력을 받아들이는 서비스들로 CWS_{ia} (Input Acceptable Candidate Web Services) 집합이 구성되고, 사용자가 원하는 출력 중 적어도 하나의 출력을 제공하는 웹 서비스들로 CWS_{oa} (Output Acceptable Candidate Web Services) 집합이 구성된다. 또한 사용자가 요구하는 기능(행위, 대상)을 만족시키는 웹 서비스들로 CWS_{sc} (Single Core Candidate Web Services) 집합이 구성된다. 그림 7은 사용자 요구사항, {{계산하다, 거리}, {나라, 나라}, {거리, Mile}}, 으로부터 후보 서비스 집합을 생성하는 예이다. 사용자의 입력 '나라'를 받아들이는 서비스 C₁이 CWS_{ia}의 원소로 선택되고, 사용자가 요구하는 출력 '거리_Mile'을 제공하는 서비스 C₅가 CWS_{oa}의 원소로 선택된다. 또한 사용자가 요구하는 기능 {계산하다, 거리}를 제공하는 서비스 C₂가 CWS_{sc}의 원소로 선택된다.

한편 행위 온톨로지 구성 제약조건에 의해 행위 온톨로지 상의 부모 개념은 하나 이상의 자식 개념의 조합으로 제공될 수 있다. 따라서 사용자가 요구하는 기능의 행위 개념이 온톨로지 상에서 자식 개념을 갖는 경우, 사용자가 요구하는 기능은 이들 자식 개념을 행위로 갖는 하나 이상의 웹 서비스들의 조합으로 제공될 가능성이 있다. 복합 서비스 구성 과정에 이를 반영하기 위해 사용자가 요구하는 행위 개념의 자식 개념을 행위로 갖는 서비스들로 CWS_{cc}

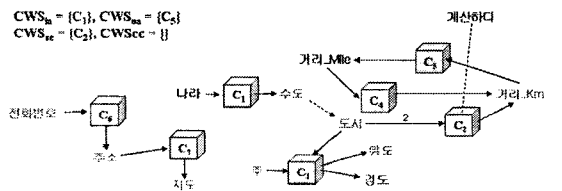


그림 7. 후보 서비스 집합 구성의 예

(Composite Core Candidate Web Services) 집합이 구성된다. CWS_{sc} 와 CWS_{cc} 는 사용자가 요구하는 기능을 제공하기 위해 필요한 후보 핵심 서비스 집합이고, CWS_{ia} 와 CWS_{oa} 는 사용자로부터 입력을 받아들이고 핵심 서비스의 결과를 사용자가 원하는 형태로 변환하는 후보 부가 서비스 집합이다.

행위 연결 그래프 탐색 단계에서는 행위 연결 그래프 상에서 CWS_{ia} 원소들의 행위 개념으로부터 CWS_{sc} 또는 CWS_{cc} 원소들의 행위 개념 중 적어도 하나를 경유하여 CWS_{oa} 원소들의 행위 개념으로 향하는 모든 경로를 찾는다. 행위 연결 그래프에서 이웃하는 두 행위 개념 정점 사이에 간선이 존재하지 않는 것은 두 행위 개념을 갖는 어떤 웹 서비스들도 서로 연결될 수 없음을 뜻한다. 따라서 CWS_{ia} 원소들의 행위 개념으로부터 CWS_{oa} 원소들의 행위 개념으로의 경로가 존재하지 않으면 사용자가 요구하는 입력과 출력을 만족시키는 복합 웹 서비스는 구성될 수 없다.

또한 생성되는 복합 웹 서비스가 사용자가 요구하는 기능을 제공함을 보장하기 위해서는 구성된 복합 웹 서비스가 핵심 서비스를 반드시 포함해야만 한다. 핵심 서비스는 사용자가 요구하는 기능을 제공하는 서비스로서 CWS_{sc} 에 속한 임의의 서비스 혹은 CWS_{cc} 에 속한 하나 이상의 웹 서비스의 조합이 핵심 서비스가 될 수 있다. 따라서 행위 연결 그래프 탐색 과정에서 찾아지는 경로는 CWS_{sc} 또는 CWS_{cc} 원소들의 행위 개념 중 적어도 하나의 행위 개념을 중간 정점으로서 포함해야만 한다. 그림 8은 그림 7에서 생성된 후보 서비스 집합에 대해 행위 연결 그래프 상에서 복합 웹 서비스 구성을 위한 경로를 탐색하는 예이다. CWS_{ia} 집합의 원소인 C_1 의 행위는 '알려주다'이고 CWS_{oa} 집합의 원소인 C_5 의 행위는 '변환하다'이며 CWS_{sc} 집합의 원소인 C_2 의 행위는 '계산하다'이다. 이로부터 행위 온톨로지 상에서 '알려주다'를 출발하여 '계산하다'를 거쳐 '변환하다'로 향하는 경로 '알려주다 계산하다 변환하다'가 찾아진다.

서비스 조합 단계에서는 행위 연결 그래프 탐색 단계에서 찾아진 경로 상에 포함된 서비스들을 조합하여 사용자의 요구사항을 만족시키는 복합 웹 서비스를 구성한다. 사용자가 요구한 각각의 입력에 대해 해당 입력을 받아들이는 서비스를 시작점으로 하는 경로들을 대응시키고, 유사한 방법으로 사용자가 요

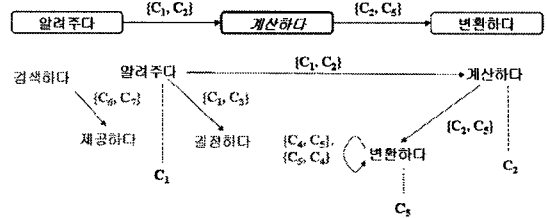


그림 8. 행위 연결 그래프에서 경로 탐색의 예

구한 출력에 대해 경로들을 대응시킨다. 그리고 사용자의 입력과 출력을 모두 포함하는 경로들의 집합을 구한 후, 집합에 속하는 경로들의 공통 정점을 하나로 합쳐 연결한다. 마지막으로 경로 상에 존재하는 웹 서비스들을 순서대로 연결하여 복합 웹 서비스를 구성한다. 그림 9는 그림 8에서 탐색된 경로로부터 복합 웹 서비스를 구성하는 예이다.

제안된 복합 웹 서비스 구성 방법의 알고리즘은 그림 10, 그림 11과 같다.

3.4 시간복잡도

복합 웹 서비스의 자동 구성 기법을 실제 세계에 적용하기 위해서는 아직 해결해야할 많은 문제들이 남아있다. 특히 자동 구성 알고리즘의 시간 복잡도를 줄이는 것은 해결해야 할 가장 중요한 문제 중의 하나이다. 일반적으로 주어진 요구사항을 만족시키는 유효한 복합 웹 서비스를 구성하는 것은 매우 복잡한 문제이다. 복합 웹 서비스의 자동 구성에 대한 대부분의 연구들은 단순히 웹 서비스의 입력과 출력들을 서로 연결하여 복합 웹 서비스를 구성한다. 그러나 이러한 방법은 연결 가능한 입력과 출력의 모든 조합을 고려해야만 하기 때문에 시간 복잡도가 매우 크다.

이미 언급한 바와 같이 제안된 복합 웹 서비스 자동 구성 알고리즘은 후보 서비스 집합의 생성, 서비스 그래프에서 복합 서비스 구성을 위한 경로 탐색, 복합 웹 서비스 구성의 세 단계로 구성된다. 이 절에서는 각각의 단계에 대한 시간 복잡도를 계산하고

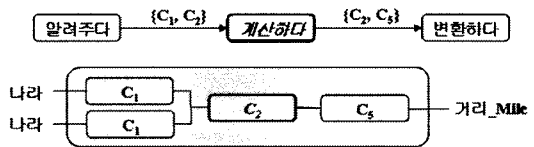


그림 9. 탐색된 경로로부터 복합 웹 서비스를 구성하는 예

```

CWlist findCompositeServices(act, obj, In, Out)
{
    List CWlist = null;

    // 후보 서비스 집합 구성
    Set CWSia = findCWSia(In);
    If(CWSia is an empty set)
        return null;
    Set CWSoa = findCWSoa(Out);
    If(CWSoa is an empty set)
        return null;
    Set CWSsc = findCWSsc(act, obj);
    Set CWScc = findCWScc(act);

    // 행위 연결 그래프 상에서의 경로 탐색 및 복합
    // 웹 서비스 구성
    If(CWSsc is not an empty set)
    {
        List pathList = findAllPaths(CWSia, CWSsc,
                                     CWSoa);

        if(pathList is not an empty list)
            CWlist = buildCompositeServices(pathList);
    }
    If(CWScc is not an empty set)
    {
        List pathList = findAllPaths(CWSia, CWScc,
                                     CWSoa);

        if(pathList is not an empty list)
            CWlist.add(buildCompositeServices(pathList));
    }

    return CWlist;
}
    
```

그림 10. 제안된 복합 웹 서비스 구성 알고리즘

이로부터 복합 웹 서비스 구성 단계 전체에 대한 시간 복잡도를 계산한다. 시간 복잡도를 계산하고 분석하기 위해 제안된 방법에서는 Hashemian과 Mavaddat의 방법에서와 같이 사용자 요구사항에서 입력과 출력 매개변수의 수(i 와 o)가 상수 상한값을 가진다고 가정한다. 또한 행위 그래프 상의 특정 행위를 구현하는 가능한 서비스의 수(t) 또한 상수 상한값을 가진다고 가정한다. 각각의 단계에 대한 시간 복잡도는 다음의 식 (1), 식 (2), 식 (3)과 같다.

$$\text{후보 서비스 집합 생성 단계의 시간 복잡도} = O(c) \quad (\text{단, } c \text{는 상수}) \quad \text{식 (1)}$$

후보 서비스 집합 생성 단계에서 네 개의 집합 CWS_{ia} , CWS_{oa} , CWS_{sc} , CWS_{cc} 가 사용자의 요구사항에 기반하여 탐색된다. 이 과정은 온톨로지 개념들의

```

CWlist buildCompositeServices(pathList)
{
    List CWlist = null;
    Set serviceListSet = null;

    // 이전 단계에서 탐색된 각각의 경로에 대해 연결
    // 가능한 모든 서비스 리스트를 탐색
    for(int i=0; i<pathList.size(); i++)
    {
        Path path = pathList.get(i);
        Set serviceListSet = findServiceList(path);
        if(serviceListSet is an empty set)
            continue;
    }

    if(serviceListSet is an empty set)
        return null;

    // 서비스 리스트를 조합하여 복합 웹 서비스 구성
    CWlist = findComposition(serviceListSet);

    return CWlist;
}
    
```

그림 11. 탐색된 행위 연결 그래프의 경로로부터 복합 웹 서비스를 구성하는 알고리즘

해시테이블 (hash table)을 사용하여 상수 시간 안에 수행될 수 있다. 예를 들어 주어진 요구사항에서 각각의 입력에 대해 데이터 의존 그래프 상의 연관된 개념 정점은 해시테이블에서 한 번에 찾을 수 있고, 이를 통해 해당 입력을 받아들이는 서비스 역시 쉽게 찾을 수 있다. 그러므로 CWS_{ia} 의 모든 원소를 찾기 위해 해시테이블을 i 번만 찾아보면 된다. 이와 유사하게 CWS_{oa} , CWS_{sc} , CWS_{cc} 를 구성하기 위해 해시테이블을 각각 o , 2 번 그리고 사용자가 요구하는 행위 개념의 자식 개념 수만큼 찾아보면 된다. 결과적으로 후보 서비스 집합 생성 단계의 시간 복잡도는 상수 시간이 된다.

$$\begin{aligned} & \text{서비스 그래프에서 복합 웹 서비스} \\ & \text{구성을 위한 경로 탐색 단계의} \\ & \text{시간 복잡도} = O(v^{l(v+e)}) \quad \text{식 (2)} \\ & (\text{단, } v \text{와 } e \text{는 행위 그래프에서 정점과} \\ & \text{간선의 수이고, } l \text{은 경로 길이의 상한값이다.}) \end{aligned}$$

두 번째 단계에서는 행위 그래프 상에서 CWS_{ia} 의 원소 서비스가 구현하는 행위 개념 정점으로부터 CWS_{oa} 의 원소 서비스가 구현하는 행위 개념 정점으로 향하는 모든 경로가 탐색된다. 이 때, 모든 경로는 CWS_{sc} 또는 CWS_{cc} 의 원소 서비스가 구현하는 행위

개념 정점을 적어도 한번 거쳐야 한다. 그래프에서 경로를 탐색하는 시간은 그래프의 사이즈에 비례하여 선형으로 커지게 되며, 시간 복잡도는 $O(v+e)$ 와 같다. 한편 제안된 행위 그래프가 순환 구조를 포함할 경우, 이로 인해 경로 탐색 과정이 끝나지 않는 문제가 발생할 수 있다. 따라서 제안된 방법에서는 어떠한 경우에도 알고리즘의 종료를 보장하기 위하여 탐색되는 경로의 길이 상한값을 상수 l 로 제한한다. 일반적으로 그래프 상에서 두 정점 사이의 최대 길이 l 인 모든 경로를 찾는 과정의 시간 복잡도는 $O(v^l(v+e))$ 가 된다. 그러나 제안된 알고리즘은 CWS_{sc} 또는 CWS_{cc} 의 원소 서비스가 구현하는 행위 개념 정점을 적어도 한번 거치는 경로만을 탐색한다. 따라서 두 번째 단계의 시간 복잡도는 식 (2)와 같이 계산된다.

$$\begin{aligned} & \text{복합 웹 서비스 구성 단계의 시간 복잡도} \\ &= O(v^{l-1}) + O(v^{i(l-1)}) \quad \text{식 (3)} \\ &= O(v^{i(l-1)}) \end{aligned}$$

마지막 단계에서는 이전 단계에서 탐색된 각각의 경로에 대해 가능한 순차 서비스의 목록을 구하고, 이들 순차 서비스 목록을 조합하여 사용자의 요구사항을 만족시키는 복합 웹 서비스를 구성한다. 이전 단계에서 탐색된 경로의 수가 v^{l-1} 이고 각각의 경로에 대한 길이 제한이 l 이기 때문에, 순차 서비스 목록을 찾는 과정은 $O(v^{l-1})$ 의 시간 복잡도를 갖는다. 다수의 순차 서비스 목록을 조합하여 유효한 복합 웹 서비스를 구성하는 과정은 조금 더 많은 시간이 걸린다. 모든 가능한 복합 서비스를 찾기 위해, 순차 서비스 목록의 모든 조합이 고려되어야 한다. 사용자가 제공하는 각각의 입력에 대해 v^{l-1} 개의 순차 서비스 목록이 존재한다. 그러므로 탐색된 순차 서비스 목록들로부터 유효한 복합 웹 서비스들을 구성하는 단계의 시간 복잡도는 $O(v^{i(l-1)})$ 가 된다.

$$\begin{aligned} & \text{복합 웹 서비스 구성} \\ & \text{과정 전체의 시간 복잡도} \\ &= O(c) + O(v^{l-1}(v+e)) + O(v^{i(l-1)}) \quad \text{식 (4)} \\ &= O(v^{i(l-1)}) \quad (\text{단, } i \geq 2) \end{aligned}$$

식 (1), 식 (2), 식 (3) 으로부터 복합 웹 서비스 구성 과정 전체의 시간 복잡도는 식 (4)와 같이 계산된다. 이는 사용자가 제공하는 입력의 수에 대한 지수 시간 복잡도이다.

Hashemian과 Mavaddat는 자신들이 제안한 복합 웹 서비스 구성 알고리즘에 대한 시간 복잡도를 $O(v^d(v'+e'))$ (단, v' 과 e' 은 각각 제안된 그래프 구조에서 정점과 간선의 수) 라고 계산하였다. 이는 복합 웹 서비스 구성 과정 전체에 대한 시간 복잡도가 아니라 그래프에서 복합 웹 서비스 구성을 위한 경로 탐색 과정에 대한 시간 복잡도이다. 그들은 탐색된 경로들을 조합하여 복합 웹 서비스를 구성하는 과정이 다항 시간 안에 계산될 수 있는 단순한 문제이므로 경로 탐색 과정이 전체 알고리즘의 시간 복잡도를 결정한다고 분석하였다. 그러나 사용자 요구사항에서 각각의 입력-출력 쌍에 대해 탐색된 유효한 경로가 두 개 이상일 경우, 가능한 복합 웹 서비스를 모두 찾기 위해서는 탐색된 경로들의 모든 조합이 고려되어야만 한다. 따라서 Hashemian과 Mavaddat의 복합 웹 서비스 구성 알고리즘에 대한 시간 복잡도는 $O(v^{d'})$ 이 되어야 한다.

Hashemian과 Mavaddat의 방법은 단순히 연결 가능한 모든 입력과 출력 조합을 고려하여 복합 웹 서비스를 구성하는 기존의 인공지능 기반 연구들에 비해, 사전에 가용한 서비스들의 입력과 출력을 연결한 그래프 형태의 자료구조를 사용함으로써 조금 더 나은 시간 복잡도, $O(v^{d'})$ 를 갖는다[6]. v' 과 e' 은 Hashemian과 Mavaddat가 제안한 의존 그래프에서 정점과 간선의 수를 의미한다. 이에 비해 제안된 방법은 $O(v^{i(l-1)})$ 의 시간 복잡도를 갖는다.

한편, Hashemian과 Mavaddat의 의존 그래프는 데이터 온톨로지에 포함된 개념의 수만큼의 정점 v' 을 갖는데 비해, 본 논문에서 제안된 행위 그래프는 행위 온톨로지에 포함된 개념의 수만큼의 정점 v 를 갖는다. 일반적으로 도메인 온톨로지는 그림 4에서와 같이 데이터 개념 보다 훨씬 적은 수의 행위 개념을 포함한다. 따라서 대부분의 경우에 $v > v'$ 가 되고, 이는 제안된 방법이 Hashemian과 Mavaddat의 방법보다 더욱 빠른 시간 복잡도를 가짐을 의미한다. 또한 이를 통해 제안된 방법이 입력과 출력의 단순 연결에 기반한 기존의 인공지능 기반 연구들에 비해 우수한 시간 복잡도를 가짐을 알 수 있다.

4. 실험 결과 및 분석

제안된 방법은 기존의 연구들과 마찬가지로 온톨

로지 상에서 같은 타입의 입출력 매개변수를 갖는 서비스들을 연결하여 복합 웹 서비스를 구성한다. 그러나 복합 웹 서비스 구성 과정에서 서비스의 입출력 정보 외에 기능적 의미를 추가적으로 고려하여 사용자가 요구하는 기능과 의미적 연관성이 없는 서비스는 탐색 대상에서 제외시킨다. 그러므로 전제조건과 효과가 존재하지 않는 정보 서비스에 대해 제안된 방법은 기존의 방법보다 빠르게 복합 웹 서비스를 구성할 수 있으며 구성된 복합 웹 서비스들은 평균적으로 더 높은 정확률을 갖게 된다. 이를 확인하기 위하여 본 논문에서는 제안된 방법과 기존 방법에 대해 복합 웹 서비스 구성에 걸리는 시간 및 생성된 복합 웹 서비스의 정확률을 실험을 통하여 비교 분석하였다.

4.1 복합 웹 서비스 구성에 소요되는 시간

복합 웹 서비스 자동 구성에 관한 대부분의 기존 연구들[6,7,11]은 복합 웹 서비스를 구성하기 위해 모든 서비스의 조합을 고려하기 때문에 지수승의 시간 복잡도를 갖는다. 이에 비해 Hashemian과 Mavaddat의 방법은 웹 서비스를 저장소에 등록하는 시점에 같은 타입의 입출력 매개변수를 갖는 서비스들을 연결하여 그래프 형태로 저장하고, 이를 통해 복합 웹 서비스 구성 과정에서 연결 가능한 서비스들만을 고려하기 때문에 보다 낮은 시간 복잡도를 갖는다.

한편 Sirin 등 [8]의 방법은 특정 도메인에서 미리 정의된 HTN을 사용하여 매우 빠른 시간 안에 정확한 복합 웹 서비스를 구성할 수 있다. 그러나 관련 연구에서 언급하였듯이 이 방법은 다수의 개별 웹 서비스를 조합하여 새로운 기능을 제공하는 일반적인 의미의 복합 웹 서비스 구성에 적용할 수 있는 방법이 아니다. 따라서 본 논문에서는 제안된 방법의 성능을 평가하기 위하여 일반적인 의미의 복합 웹 서비스 자동 구성 방법들 중에서 가장 빠른 시간 복잡도를 갖는 Hashemian과 Mavaddat의 방법과 복합 웹 서비스 구성 과정의 속도를 비교 실험하였다.

정확한 비교 실험을 위해서는 사용자의 요구사항으로부터 각각의 방법을 통해 복합 웹 서비스를 구성하는 모든 단계에 소요되는 시간을 비교해야만 한다. 그러나 Hashemian과 Mavaddat의 논문에서는 그래프 구조에서 탐색된 다수의 경로로부터 복합 웹 서비스를 조합하는 방법이 명확하게 제공되지 않는다. 한편 이전 절에서 시간 복잡도의 분석을 통해 확인할

수 있는 바와 같이 복합 웹 서비스 구성 과정의 시간 복잡도는 복합 웹 서비스 구성을 위해 탐색된 경로를 조합하는 과정에 의해 결정된다. 그러나 두 방법의 결정적인 차이는 경로를 탐색하는 과정이며 경로를 조합하는 과정은 같은 시간 복잡도를 갖는 유사한 방법으로 수행 가능하다. 따라서 본 논문에서는 각각의 방법에 대해 그래프에서 경로를 탐색하는 과정의 수행 속도를 측정하고 이를 통해 두 방법의 성능을 간접적으로 비교하였다.

그래프에서 필요한 경로를 탐색하는데 걸리는 시간은 그래프의 크기와 탐색해야 할 경로의 수에 의해 결정된다. 제안된 방법과 Hashemian과 Mavaddat의 방법은 서비스의 수와 온톨로지의 크기에 따라 탐색해야 할 그래프의 크기가 결정된다. 또한 사용자가 요구하는 입력과 출력의 수에 의해 그래프에서 탐색해야 할 경로의 수가 결정된다. 따라서 본 논문에서는 표 1에서와 같이 세 종류의 실험을 통해 제안된 방법과 Hashemian과 Mavaddat의 방법의 수행 속도를 비교하였다. 제안된 방법과 Hashemian과 Mavaddat의 방법을 제안된 알고리즘에 따라 Java를 이용하여 구현하였으며, 주어진 요구사항으로부터 복합 웹 서비스 구성을 위한 경로를 탐색하는데 걸리는 시간을 시스템 타임 스탬프를 통해 측정하였다.

실험을 위해서는 충분한 수의 가용한 웹 서비스와 다양한 크기의 온톨로지가 필요하다. 그러나 실제로 이러한 의미 있는 실험 데이터를 구하거나 구축하는 것은 매우 힘들다. 따라서 본 논문에서는 실험에 사용된 서비스와 온톨로지, 그리고 사용자 요구사항을 표 1의 조건을 만족하도록 임의로 생성하였다. 실험 1은 100개의 데이터 개념과 50개의 행위 개념을 가지는 온톨로지와 5개의 입출력을 갖는 20개의 사용자 요구사항에 대해 가용한 웹 서비스의 수를 50~250개로 늘려가면서 그래프 탐색에 걸리는 시간을 측정하였다. 같은 방법으로 실험 2에서는 온톨로지의 크

표 1. 그래프 탐색 과정의 수행 속도를 비교하기 위한 실험 구성

	서비스의 수	온톨로지의 크기		사용자 요구사항 I/O의 수
		데이터 개념의 수	행위 개념의 수	
실험데이터1	50~250	100	50	5
실험데이터2	150	25~225	13~118	3
실험데이터3	100	50	25	2~10

기를, 실험 3에서는 사용자 요구 사항의 입출력의 수를 변화 시켜가며 실험하였다. 실험 결과는 각각 그림 12, 그림 13, 그리고 그림 14와 같다.

그림 12는 가용한 웹 서비스의 수가 증가함에 따라 그래프 탐색 과정에 걸리는 시간의 증가를 보여준다. Hashemian과 Mavaddat의 방법은 레지스트리에 등록된 서비스와 서비스의 입출력 개념으로 의존 그래프를 구성하기 때문에 서비스의 수가 증가함에 따라

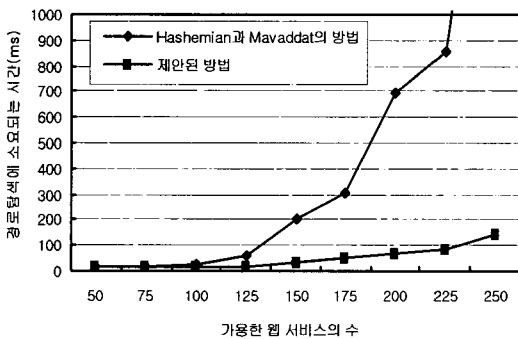


그림 12. 가용한 웹 서비스의 수에 따른 경로 탐색 속도

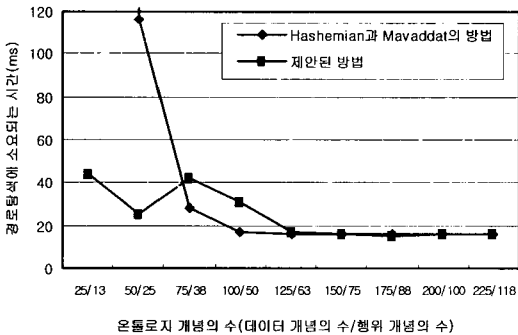


그림 13. 온톨로지의 크기에 따른 경로 탐색 속도

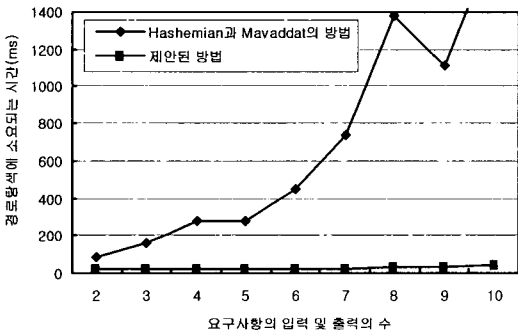


그림 14. 요구사항의 입출력 수에 따른 경로 탐색 속도

라 의존 그래프의 크기도 증가한다. 또한 의존 그래프 상에서 사용자가 제공하는 입력과 원하는 출력 간의 경로를 탐색하기 때문에 그래프의 크기가 커짐에 따라 탐색 속도가 급격하게 느려진다.

제안된 방법 역시 서비스의 수가 증가함에 따라 데이터 의존 그래프의 크기가 증가한다. 그러나 Hashemian과 Mavaddat의 방법과 달리 행위 개념의 수에 의해 크기가 결정되는 행위 연결 그래프에서 경로를 탐색한다. 따라서 서비스의 수가 증가하면 후보 서비스 집합의 크기가 커짐에 의해 행위 연결 그래프를 탐색하는 횟수가 늘어나지만, 상대적으로 매우 작은 크기의 행위 연결 그래프에서 모든 탐색이 이루어지므로 탐색 속도가 크게 증가하지는 않는다.

그림 13은 온톨로지의 크기가 커짐에 따라 그래프에서 경로를 탐색하는데 걸리는 시간이 점점 감소하는 것을 보여준다. 이러한 결과는 실험에 사용된 가용한 서비스의 수가 150개로 고정되어 있기 때문이다. Hashemian과 Mavaddat의 의존 그래프는 온톨로지의 크기가 아니라 등록된 웹 서비스의 수와 웹 서비스가 사용하는 입출력 개념의 수에 의해 그래프의 크기가 결정된다. 따라서 온톨로지의 크기가 커진다고 해서 의존 그래프의 크기가 커지지 않는다. 제안된 방법 역시 데이터 의존 그래프의 크기는 온톨로지의 크기에 영향을 받지 않는다. 또한 행위 연결 그래프에서 정점의 수가 증가하기는 하지만 서비스의 수가 증가하지 않기 때문에 간선의 수는 증가하지 않는다. 오히려 참조할 수 있는 데이터 및 행위 개념의 수가 증가함에 따라 사용자가 요구하는 기능 및 입출력을 제공하는 복합 웹 서비스가 존재하지 않는 경우가 다수 발생한다. 이러한 경우 그래프 탐색 과정이 초기에 종료됨에 따라 경로 탐색에 걸리는 시간이 줄어드는 결과를 보이게 된다.

그림 14는 사용자가 요구하는 입출력의 개수에 따른 경로 탐색 속도의 변화를 보여준다. Hashemian과 Mavaddat은 의존 그래프 상에서 사용자가 제공하는 모든 입력으로부터 사용자가 요구하는 모든 출력까지의 경로를 탐색한다. 따라서 사용자 요구사항의 입출력 개념이 많아질수록 탐색해야하는 경로의 수가 증가함으로 인해 그래프 탐색 과정의 속도가 느려지게 된다. 제안된 방법 역시 Hashemian과 Mavaddat의 방법과 유사하게 탐색해야하는 경로의 수가 증가하기는 하지만 탐색의 대상이 되는 행위 연결 그래프

의 크기가 매우 작고 간선의 수가 적기 때문에 탐색 속도는 크게 증가하지 않는다.

4.2 구성된 복합 웹 서비스의 정확성

제안된 방법은 데이터 온톨로지와 더불어 웹 서비스의 기능을 정의하는데 사용되는 기능 온톨로지를 추가로 사용한다. 이를 통해 복합 웹 서비스 구성 과정에서 사용자가 원하는 기능과 관계없는 서비스들을 고려하지 않는다. 따라서 제안된 방법은 웹 서비스의 입출력 매개변수만을 연결하여 복합 웹 서비스를 생성하는 기존의 방법들에 비해 사용자가 요구하는 기능에 부합하는 더욱 정확한 복합 웹 서비스를 생성할 수 있다. 이를 증명하기 위해 본 논문에서는 기능 온톨로지를 사용하지 않고 입출력 매개변수 간의 매칭만을 통해 생성되는 복합 웹 서비스와 제안된 방법을 통해 생성되는 복합 웹 서비스의 정확률을 비교 및 분석하였다.

실험을 위해 표 2의 조건을 만족하도록 도메인 온톨로지, 가용한 웹 서비스 및 사용자 요구사항을 임의로 생성하였다. 임의로 생성된 하나의 온톨로지에 대해 50개부터 250개까지 25개씩 차등을 둔 총 9개의 가용한 서비스 집합을 임의로 생성하였다. 또한 기능 온톨로지를 사용하지 않고 사용자 요구사항으로부터 복합 웹 서비스를 구성하기 위해 Hashemian과 Mavaddat의 방법을 적용하여 연결 가능한 모든 서비스들의 경로를 생성하고, 이들 경로로부터 생성 가능한 모든 복합 웹 서비스의 조합을 수동으로 구성하였다. 이렇게 생성된 결과는 사용자가 제공하는 입력만을 사용하고 요구되는 출력을 생성하는 가능한 복합 웹 서비스의 모든 조합을 포함하며, 가용한 서비스들이 전제조건과 효과가 존재하지 않는 정보 서비스들인 경우 기존의 방법들 [6,9,11]이 생성하는 결과와 동일하다.

한편 생성된 각각의 서비스 집합에 대해 해당이

표 2. 생성되는 복합 웹 서비스의 정확성을 비교하기 위한 실험 구성

	가용한 서비스의 수	온톨로지의 크기		사용자 요구사항 I/O의 수
		데이터 개념의 수	행위 개념의 수	
실험 데이터	50~250	100	50	2~5

존재하는 15개의 사용자 요구사항을 작성하였으며, 해당 서비스 집합에 대해 제안된 방법과 기능 온톨로지를 사용하지 않는 방법을 적용하여 15개의 사용자 요구사항에 대한 복합 웹 서비스 집합을 생성하고 정확률의 평균을 계산하였다. 각각의 방법을 적용하여 생성되는 복합 웹 서비스들은 모두 사용자가 제공하는 입력만으로 실행이 가능하고 사용자가 요구하는 출력을 제공한다. 그러나 사용자가 제공하는 입력을 더 많이 사용하고, 사용자가 요구하는 기능과 유사한 웹 서비스를 포함하는 복합 웹 서비스가 더욱 사용자의 요구사항에 부합되는 서비스라 할 수 있다. 따라서 생성된 복합 웹 서비스의 정확률은 식 (5)를 사용하여 계산하였다. w_1 과 w_2 의 값으로는 각각 0.2, 0.8을 사용하였다. 실험 결과는 그림 15와 같다.

$$\text{복합 웹 서비스의 정확률} = (\text{사용자 입력 사용률}) \times w_1 + (\text{기능적 유사도}) \times w_2 \quad \text{식 (5)}$$

$$\text{사용자 입력 사용률} = \left(\frac{\text{복합 웹 서비스가 이용하는 사용자 입력의 수}}{\text{사용자가 제공하는 입력의 수}} \right) \quad \text{식 (6)}$$

$$\text{기능적 유사도} = \frac{\left\{ \begin{array}{l} \text{행위와 대상이 각각 행위사용자, 대상사용자인 웹 서비스들 포함 : 1} \\ \text{행위가 행위사용자의 자손 개념인 웹 서비스들 포함 : } \frac{0.8}{2^{\text{의미적거리}}} \end{array} \right.}{1} \quad \text{식 (7)}$$

- ※ 행위사용자, 대상사용자 : 사용자가 요구하는 행위와 대상
- ※ 의미적거리 : 복합 웹 서비스를 구성하는 요소 서비스의 행위들 중 행위사용자의 자손 개념인 행위들과 행위사용자와의 온톨로지 트리에서 깊이 차이 중 최소 값

그림 15는 가용한 웹 서비스의 수가 늘어남에 따라 기능 온톨로지를 사용하지 않는 복합 웹 서비스 구성 방법의 정확률이 대체적으로 감소함을 보여준다. 기능 온톨로지를 사용하지 않는 기존의 복합 웹 서비스 구성 방법은 사용자의 입력으로부터 사용자가 요구하는 출력을 제공하는 모든 복합 웹 서비스의 조합을 생성한다. 이때 사용자가 요구하는 기능이나 개별 웹 서비스의 기능을 고려할 수 없고, 단순히 같은 타입의 입출력 매개변수를 갖는 웹 서비스들을 연결하여 복합 웹 서비스를 구성한다. 따라서 생성된 복합 웹 서비스 집합이 사용자가 요구하는 기능과 전혀 다른 기능을 하는 복합 웹 서비스들을 다수 포

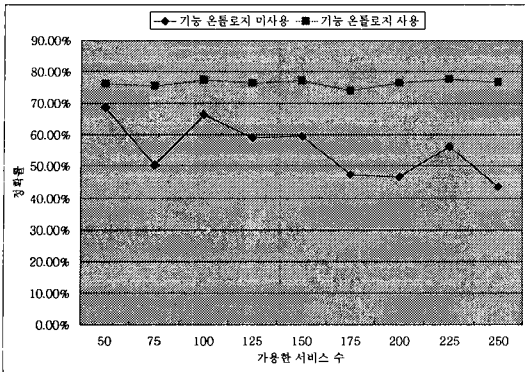


그림 15. 가용한 웹 서비스의 수에 따른 복합 웹 서비스의 정확률

합하게 되고 이로 인해 정확률이 낮아진다.

또한 가용한 웹 서비스의 수가 많아지면 서로 연결 가능한 웹 서비스의 수가 증가하게 되고 생성 가능한 복합 웹 서비스 조합의 수 역시 증가한다. 그러나 사용자가 요구하는 기능과 유사한 기능의 웹 서비스보다는 상이한 기능을 제공하는 웹 서비스가 상대적으로 더욱 많이 증가하기 때문에 기존 방법에 의해 생성되는 복합 웹 서비스 집합은 사용자의 요구사항에 부합하지 않는 복합 웹 서비스를 더욱 많이 포함하게 된다. 결과적으로 가용한 웹 서비스의 수가 증가함에 따라 생성되는 복합 웹 서비스의 정확률은 점점 낮아지게 된다.

이에 비해 제안된 방법은 기능 온톨로지를 사용하여 사용자 요구사항 및 웹 서비스의 기능적 의미를 명시적으로 기술하고 이를 그래프 형태로 구조화하여 저장한다. 이렇게 저장된 그래프를 이용하여 복합 웹 서비스 구성 단계에서는 사용자가 요구하는 기능과 관계없는 서비스들로부터 이루어진 복합 웹 서비스는 그래프 탐색 단계에서 제외할 수 있게 된다. 따라서 제안된 방법에 의해 생성되는 복합 웹 서비스 집합은 사용자가 요구하는 기능과 유사한 기능을 하는 복합 웹 서비스만을 포함하게 되며 이로 인해 가용한 웹 서비스의 수와 관계없이 높은 수준의 정확률을 유지할 수 있다.

본 절에서는 제안된 방법의 우수성을 입증하기 위해 복합 웹 서비스 구성 과정에 걸리는 시간 및 구성된 복합 웹 서비스의 정확성에 대해 다양한 실험 데이터를 사용하여 비교 실험하였다. 실험에 사용된 서비스와 온톨로지 그리고 사용자 요구사항이 임의로

생성되었기 때문에 제안된 방법의 우월성을 완전히 보장하기는 어렵다. 그러나 실험 결과를 통해 온톨로지의 데이터 개념이 행위 개념의 수보다 적은 도메인에서 제안된 방법이 기존의 방법보다 높은 확장성과 빠른 수행속도를 보임을 알 수 있다. 또한 제안된 방법은 도메인 내에 가용한 웹 서비스의 수와 상관없이 기존의 방법에 비해 높은 수준의 정확률을 갖는 복합 웹 서비스를 생성함을 확인할 수 있다.

5. 결론 및 향후 연구

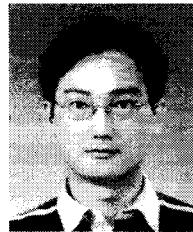
본 논문에서는 웹 서비스 및 사용자 요구사항의 기능적 의미를 명시적으로 기술하고 제안된 그래프 구조를 통해 구조화함으로써 사용자의 요구사항에 부합하는 복합 웹 서비스를 자동으로 구성하는 방법을 제안하였다. 제안된 방법은 전제조건과 효과를 갖지 않는 정보 서비스들로 구성되는 복합 웹 서비스에 대해 기존 연구들과 달리 기능적 의미의 정확성을 보장한다. 또한 복합 웹 서비스 구성 과정의 대부분이 상대적으로 크기가 작은 행위 연결 그래프 상에서 이루어지며, 사용자가 요구하는 기능과 의미적으로 연관성이 없는 서비스는 자연스럽게 탐색 대상에서 제외되기 때문에 기존 방법에 비해 작은 시간 복잡도를 갖는다.

한편 사용자가 요구하는 기능이 항상 대상과 행위의 1:1 관계로 명확하게 표현되지 않을 수 있다. 사용자가 두 개 이상의 행위나 대상을 조합하여야만 자신이 원하는 기능을 정확하게 표현할 수 있는 경우가 존재한다. 향후에는 다수의 행위와 기능 그리고 연산자를 통해 사용자가 원하는 기능을 정교하게 기술하고 이로부터 복합 웹 서비스를 자동으로 구성하는 방법을 연구할 것이다. 또한 복합 웹 서비스 구성 과정에서 서비스의 질을 고려하고 이를 기반으로 생성된 복합 웹 서비스들을 정렬하여 제공하는 순위 시스템 또한 연구할 것이다.

참고 문헌

[1] S.C. Oh, D.W. Lee, and S.R.T. Kumara, "A Comparative Illustration of AI Planning-based Web Services Composition," ACM SIGecom Exchanges, Vol.5, No.5, pp. 1-10,

- 2005.
- [2] Web Ontology Language (OWL), W3C Recommendation 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-features-20040210>.
- [3] Semantic Markup for Web Services (OWL-S), W3C Member Submission 22 November 2004, <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122>.
- [4] Web Service Modeling Ontology (WSMO), W3C Member Submission 3 June 2005, <http://www.w3.org/Submission/2005/SUBM-WSMO-20050603>.
- [5] Semantic Annotations for WSDL and XML Schema (SAWSDL), W3C Recommendation 29 August 2007, <http://www.w3.org/TR/2007/REC-sawsdl-20070828>.
- [6] R. Akkiraju, A. Ivan, R. Goodwin, B. Srivastava, and T. Syeda-Mahmood, "Semantic Matching to Achieve Web Service Discovery and Composition," Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06), pp. 70, 2006.
- [7] V. Agarwal, K. Dasgupta, N. Karnik, and A. Kumar, "A Service Creation Environment Based on End to End Composition of Web Services," Proceedings of the 14th International Conference on World Wide Web, pp. 128-137, 2005.
- [8] E Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau "HTN Planning for Web Service Composition Using SHOP2," Journal of Web Semantics, Vol.1, No.4, pp. 377-396, 2004.
- [9] S.V. Hashemian and F. Mavaddat, "A Graph-based Framework for Composition of Stateless Web Services," Proceedings of the European Conference On Web Services (ECOWS'06), pp. 75-86, 2006.
- [10] L.A.G. da Costa, P.F.Pires, and M. Mattoso, "Automatic Composition of Web Services with Contingency Plans," Proceedings of the IEEE International Conference on Web Services(ICWS'04), pp. 454-461, 2004.
- [11] J. Rao, P. Kungas, and M. Matskin, "Logic-based Web Services Composition: from Service Description to Process Model," Proceedings of the IEEE International Conference on Web Services(ICWS'04), pp. 446-453, 2004.
- [12] L. Ye and B. Zhang, "Discovering Web Services Based on Functional Semantics," Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing(APSCC'06), pp. 348-355, 2006.



신 동 훈

2003년 2월 연세대학교 컴퓨터과
학과 졸업(학사)
2005년 2월 연세대학교 컴퓨터과
학과 졸업(석사)
2005년 3월~현재 연세대학교 컴
퓨터과학과 박사과정

관심분야 : 인터넷 컴퓨팅, 서비스 지향 컴퓨팅, 웹 서비
스 조합



이 경 호

1995년 2월 연세대학교 전산과학
과 졸업(학사)
1997년 2월 연세대학교 컴퓨터과
학과 졸업(석사)
2001년 2월 연세대학교 컴퓨터과
학과 졸업(박사)

2001년 3월~2002년 8월 National Institute of Standard
and Technology(NIST) 객원연구원
2002년 9월~현재 연세대학교 컴퓨터과학과 부교수
관심분야 : 인터넷 컴퓨팅, 서비스 지향 컴퓨팅, 멀티미디어
문서 처리