

웹 서비스 환경에서 확장 객체 기반 시소러스를 이용한 개념 기반 오케스트레이션

조호상, 양재동
전북대학교 컴퓨터공학과
e-mail : spooky0113@gmail.com, jdyang2000@naver.com

Concept-based Orchestration Using eXtended-Object Thesaurus(XOT) in Web Service Environment

Ho-Sang Jo, Jae-Dong Yang
Dept. of Computer Engineering, Chon-Buk National University

요 약

웹 서비스는 SOA 를 웹 상에서 구현한 기술로, 서비스들간의 느슨한 결합을 실현함으로써 재사용 가능한 서비스들의 활용을 극대화 한다. 본 논문에서는 확장 객체 기반 시소러스(eXtended Object-based Thesaurus, XOT)를 사용하여 적응 가능한 오케스트레이션 명세가 가능한 개념 기반 오케스트레이션 기법을 제안한다. XOT 는 재사용 가능한 서비스들이 속하는 범주들을 개념으로 표현하고 상호 합성 패턴 관계를 포함한 개념간 상관 관계를 명시한 일종의 지식베이스이다. 이 기법은 오케스트레이션에 참여하는 복합 서비스를 포함한 여러 다양한 서비스들을 구체적으로 명세하는 대신 XOT 내의 개념들로 변수화 하고, 이들을 오케스트레이션 시 적절한 재사용 가능한 서비스 또는 구성 서비스들의 조합으로 대체시킴으로써 서비스들의 재사용성을 획기적으로 개선할 수 있다.

키워드: 웹 서비스, 정보검색, 온톨로지

1. 서론

웹 서비스는 서비스 지향 아키텍처를 웹 상에서 구현한 연동 기술로, 서비스 간 느슨한 결합을 실현함으로써, 서비스들 간 합성과 대체를 용이하게 한다. 결과적으로 재사용 가능한 서비스들의 활용을 극대화 하기 때문에, 웹 서비스는 전반적인 소프트웨어 개발 및 유지 보수 비용을 절감시키는 효과가 있다[1].

웹 서비스를 발견하기 위한 UDDI 와 포털의 검색 엔진은 키워드 기반으로 검색을 수행한다. 그러나, 키워드 검색 방법은 사용자의 요구 사항을 정확하게 표현하지 못하고 WSDL 의 의미적 표현력 부족 때문에, 정확률과 재현율이 현격히 떨어지는 문제점이 있다.

이를 보완하기 위해, [2,3]에서는 OWL-S, WSMO 와 같은 온톨로지를 이용하여 웹 서비스를 기술함으로써, 시맨틱 질의와 개념 기반 정합이 가능하도록 하였다.

한편, 발견된 웹 서비스를 합성하는 연구들은 오케스트레이션(orchestration)과 코레오그래피(choreography)에 기반을 두는데, 이 연구들은 오케스트레이션 방식의 BPEL(Business Process Execution Language)을 주로 많이 사용한다[4]. 오케스트레이션은 개발자의 많은 시행착오와 오류를 발생시키므로, 이를 지원하는 방법들이 많이 연구되었다. 이 중에서, [5-7]은 목적 달성을 위한 복합 웹 서비스의 성공적 구현에 초점을 맞추었기 때문에, 웹 서비스 합성 시 구성 서비스들 간 입출력 호환성을 판별하는 작업에 있어서 융통성과 적응성이 떨어지는 문제점이 있다.

이러한 문제점을 [8,9]에서는 온톨로지에 명시된 포괄(subsumption), plug-in, exact 관계를 추론함으로써 한 구성 서비스의 출력과 이를 입력으로 받아들이는 다른 서비스 인터페이스 사이의 파라미터 호환성을 판별함으로써 해결하고자 하였다.

그러나, 이 연구들 또한, 이용 가능한 웹 서비스들을 사전에 범주화하지 않았기 때문에 복합 웹 서비스 합성 시, 결합 가능한 서비스들의 판별에 여전히 많은 소비시간이 요구되는 한계점을 지닌다.

본 논문에서는 서비스들이 속하는 범주들 간의 상호 합성 패턴 관계를 포함한 다양한 관계성을 명시하는 확장 객체 기반 시소러스(eXtended Object-based Thesaurus, XOT)를 사용하여 동적으로 적응 가능한 오케스트레이션 명세가 가능한 개념 기반 오케스트레이션 기법을 제안한다. 이 기법은 오케스트레이션에 참여하는 복합 서비스를 포함한 여러 다양한 서비스들을 구체적으로 명세하는 대신 XOT 내의 개념들로 변수화 하고, 이들을 오케스트레이션 시 적절한 재사용 가능한 서비스들, 또는 구성 서비스들 간의 조합으로 대체시킴으로써 서비스들의 재사용성을 획기적으로 개선할 수 있다.

본 논문의 구성은 다음과 같다. 먼저, 2 절에서 웹 서비스를 범주화하고 개념간 관계성 정보를 가지는 XOT 에 대해서 설명하고 3 절에서는 XOT 를 이용한 개념 기반 오케스트레이션의 대해서 서술한다. 4 절에서는 본 논문이 제안한 시스템 구현에 대해 기술하고,

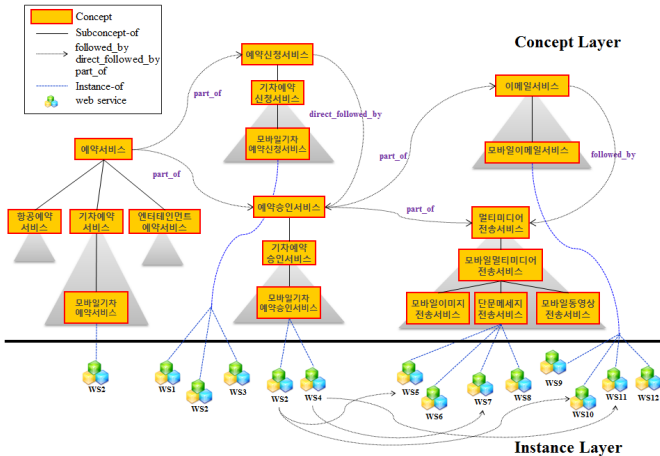
마지막으로 5 절에서는 결론 및 향후 연구에 대해서 제시한다.

2. 확장 객체 기반 시소러스 (XOT)

2.1 XOT 의 개념적 구조

XOT 는 개념 기반으로 웹 서비스들을 오케스트레이션 할 때, 사용되는 지식베이스이다[10]. XOT 는 구조화된 개념 집합으로 볼 수 있고, 하나의 개념은 웹 서비스의 범주를 표현한다. 시스템은 개념 사이에 설정된 다양한 관계성 링크들에 의해 서비스들 간 상관관계를 추적하고 분석함으로써, 사용자의 의도에 맞는 적절한 웹 서비스들을 발견하고, 오케스트레이션을 완성할 수 있도록 지원한다.

XOT 는 [그림 1]처럼 웹 서비스가 포함되는 범주들을 개념으로 표현하고 이를 일반화한 계층 구조로, 각 개념 간 합성 패턴 관계를 나타내는 *followed_by*, *direct_followed_by*, *part_of* 등을 포함하는 다양한 관계성들을 가질 수 있다. 각 개념은 해당 범주에 속하는 재사용 가능한 서비스 ID 들을 인스턴스로 갖는다.



(그림 1) 확장 객체 기반 시소러스(XOT) 개념 구조

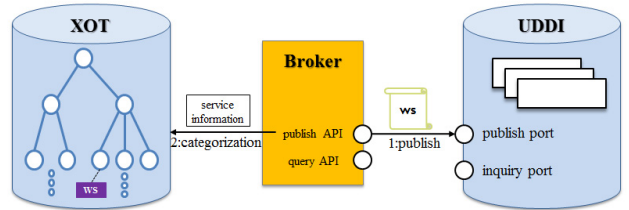
part_of 관계성은 2 개 이상의 서비스로 구성되는 복합 서비스 합성 패턴을 명시하기 위해 사용된다. 예를 들어, 모바일기차예약 서비스는 모바일기차예약신청 서비스와 모바일기차예약승인 서비스로 구성되고, 모바일기차예약승인 서비스는 다시 모바일메일 서비스와 모바일멀티미디어전송 서비스로 구성되는 합성 패턴을 표현하기 위해 *part_of* 가 사용될 수 있다. 여기서 하위 서비스는 상위 서비스의 *part_of* 관계성을 자동으로 상속받는다.

모바일 기차예약신청 서비스 바로 다음에 모바일기차예약승인 서비스가 합성될 수 있음은 *direct_followed_by* 를 사용해 표현 가능하다. 같은 합성 순서를 나타내는 *followed_by* 관계성과 달리 이 관계성이 사용될 경우 오케스트레이션을 담당하는 브로커는 서비스 합성 시, *direct_followed_by* 로 연관된 두 구성 서비스들 간에 다른 어떤 서비스도 참여시킬 수 없게 된다. 이와 달리, *followed_by* 관계성을 사용하는, 모바일메일 서비스의 경우, 이서비스는 모바일멀티미디어전송 서비스와 바로 합성될 수도 있지만, 또

다른 서비스, 가령, 휴대폰진동 서비스와 먼저 합성될 수도 있다.

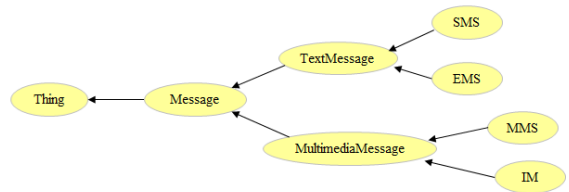
2.2 XOT 의 인스턴스 구축

서비스 제공자는 일반적으로 서비스를 등록할 때, 서비스를 WSDL 로 명세한 후, UDDI 레지스트리에 등록한다. 그러나, 본 시스템에서는 이와 동시에 [그림 2]처럼 서비스들을 범주화시켜 해당 XOT 개념의 인스턴스로 등록해야 하고, 개념 간 합성 패턴 관계성에 따라, 다른 개념 및 인스턴스들 간을 사전에 링크시키는 부수적인 작업이 필요하다.



(그림 2) 웹 서비스 등록 및 XOT 인스턴스 구축

앞서 언급한대로 웹 서비스들이 인스턴스로 등록된 XOT 의 한 개념은 다른 개념과 합성 패턴 관계성을 가질 수 있기 때문에, 두 개념 내의 각 인스턴스들 간에도 합성 가능한 관계성이 명시될 필요가 있다. 예를 들어, 개념 단문메시지전송에 새로운 인스턴스 ws5 가 등록되었을 때, 이 개념은 모바일메일 개념과 *followed_by* 관계성을 맺고 있으므로, 모바일메일 내의 인스턴스들 간과의 합성가능 여부를 판별해야 한다. 이 때, 어떤 서비스 인터페이스 출력이 텍스트 메시지 타입이고, 연계되는 서비스 인터페이스 입력이 메시지 타입이라면, 정확히 정합을 하였을 경우, 이 두 서비스들은 합성될 수 없다.



(그림 3) 파라미터 도메인 온톨로지

그러나, [그림 3]과 같은 파라미터 도메인 온톨로지의 포괄 관계 추론을 이용한 매칭 알고리즘을 통해, 텍스트 메시지의 상위에 메시지가 있는 것을 발견하고 서로 호환될 수 있음을 이끌어 낼 수 있다.

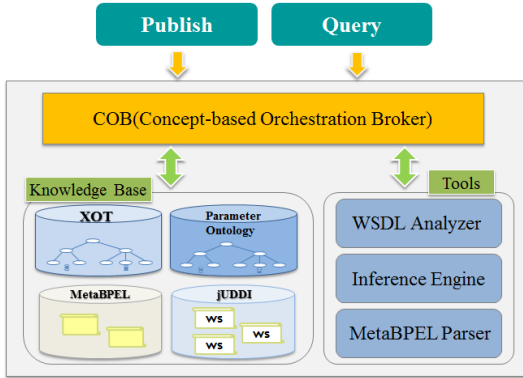
3. XOT 를 이용한 개념 기반 오케스트레이션

이 절에서는 XOT 를 이용하여, 오케스트레이션을 개념화함으로써, 웹 상에서 이미 개발된 서비스들의 재사용성을 극대화하는 방식을 기술한다. 오케스트레이션을 개념화 한다는 것은, XOT 의 합성 패턴 관계성에 기반하여 XOT 의 개념을 오케스트레이션에 참여하는 서비스 명으로 사용할 수 있다는 의미로, 개념화된 오케스트레이션은 MetaBPEL 로 표현한다.

MetaBPEL 문서는 사용자가 Eclipse BPEL Designer 와 같은 시각화된 합성 도구를 이용하여 직접 생성할 수 있다[11].

3.1 시스템 구조

본 논문에서 제안하는 시스템은 [그림 4]와 같이 지식베이스, 개념 기반 오케스트레이션 브로커, 그리고 시맨틱 서비스 합성 지원도구 등 크게 세 부분으로 구성된다.



(그림 4) 시스템 구조

지식베이스는 XOT, 매칭 알고리즘에 사용되는 파라미터 온톨로지, 그리고 XOT 의 합성 패턴에 따라 생성된 MetaBPEL 저장소와 웹 서비스 정보를 저장하는 jUDDI 를 포함한다.

개념 기반 오케스트레이션 브로커는 지식베이스를 참조하고 시맨틱 서비스 합성 지원 도구를 활용하여 사용자의 요구에 부응하기 위해, 한 MetaBPEL 문서에 대응되는 BPEL 문서를 생성하고 오케스트레이션을 수행하는 일과 서비스 발견, 등록 등의 역할을 담당한다.

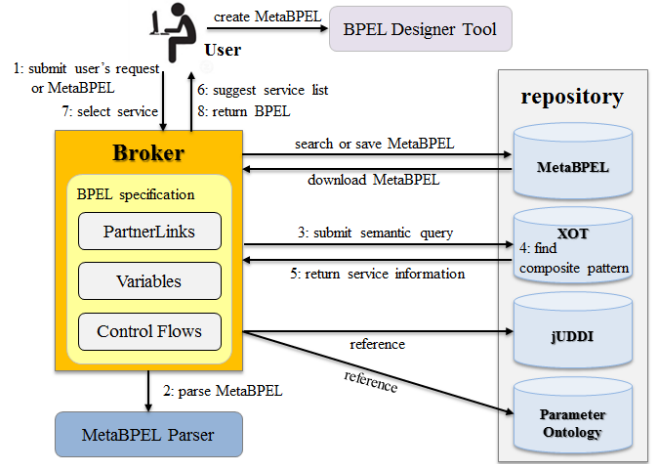
WSDL 분석기, 추론 엔진, MetaBPEL 파서로 이루어진 시맨틱 합성 지원도구는 브로커를 통한 웹 서비스의 등록 및 개념 기반 오케스트레이션을 지원하는 도구로써, 브로커와 상호 연동한다.

3.2 개념 기반 오케스트레이션

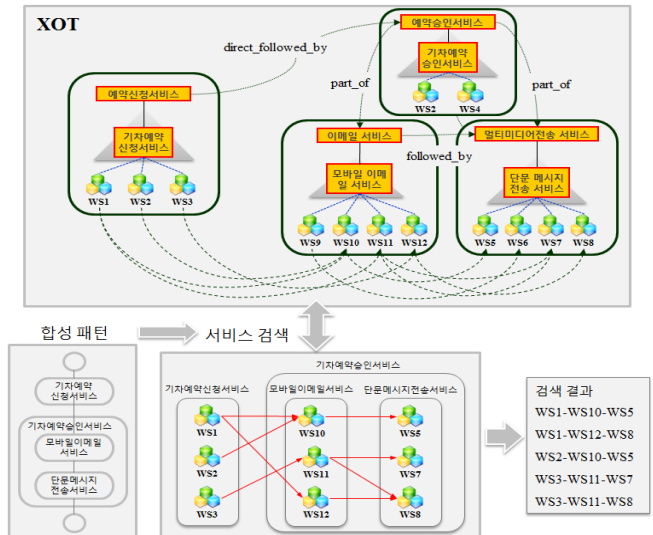
오케스트레이션을 위해 브로커는 먼저, 자연어로 기술된 사용자 요구를 받아들인 후, MetaBPEL 저장소에서 이 요구와 가장 유사한 MetaBPEL 문서들을 검색, 사용자에게 제시하고, 선택된 MetaBPEL 을 다운로드한다. 사용자가 만족하는 MetaBPEL 이 없다면 사용자는 직접 MetaBPEL 을 생성하여 브로커에게 제공할 수 있고, 이는 저장소에 저장되어 다른 사용자에게 참조될 수 있다.

두 번째로, MetaBPEL 내의 서비스명을 인스턴스로 대체하기 위해, 브로커는 파서를 이용해 MetaBPEL 로부터 서비스명들을 추출한다. 세 번째로, MetaBPEL 문서에 명시된 순서에 맞게 해당 서비스들에 해당되는 개념을 XOT 에서 검색하고 해당 인스턴스들을 단계별로 사용자에게 제시, 사용자의 선택을 유도한다. 마지막으로, 인스턴스들만으로 이루어진 오케스트레이션을 명세하는 최종 실행 가능한 BPEL 문서 결과

를 사용자에게 제공한다. [그림 5]는 위의 개념 기반 오케스트레이션 과정을 도식화해서 보여주고 있다.



(그림 5) 개념 기반 오케스트레이션



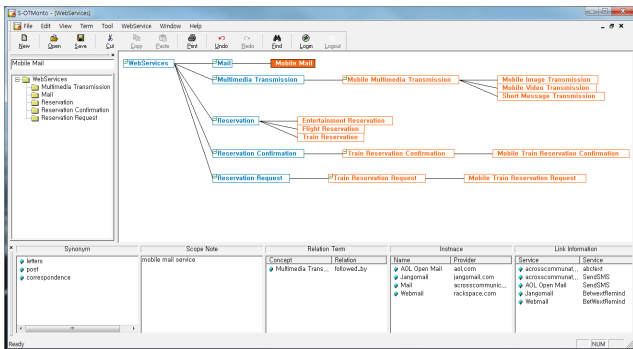
(그림 6) MetaBPEL 의 인스턴스화

[그림 6]은 인스턴스로 대체시킬 각 해당 서비스명을 XOT 에서 검색하여 사용자에게 보여주는 예이다. 먼저, 기차예약이 BPEL 문서에 포함되어 있을 경우, 브로커는 XOT 에서 개념 기차예약요청을 검색하여 인스턴스로 등록되어 있는 서비스 ws1, ws2, 그리고 ws3 를 추출한다. 다음으로 각각의 인스턴스들과 합성될 수 있는 개념 기차예약중인 내의 인스턴스들, 예를 들어, ws1 과 연결된 ws10 과 ws12 를 추출하여 사용자에게 보여준다. 계속해서, 대응되는 개념 모바일 메일과 단문메시지전송을 part_of 관계성을 이용하여 파악하고 각 인스턴스들을 followed_by 순서에 따라 사용자에게 제시한다. 사용자는 각 단계에 따라 브로커가 제안하는 인스턴스들을 선택하고, 단계별로 선택된 인스턴스들의 조합을 브로커는 최종적으로 해당 MetaBPEL 내 서비스명에 대체시킴으로써 오케스트레이션을 수행하고 실행 가능한 BPEL 문을 결과로 생성한다.

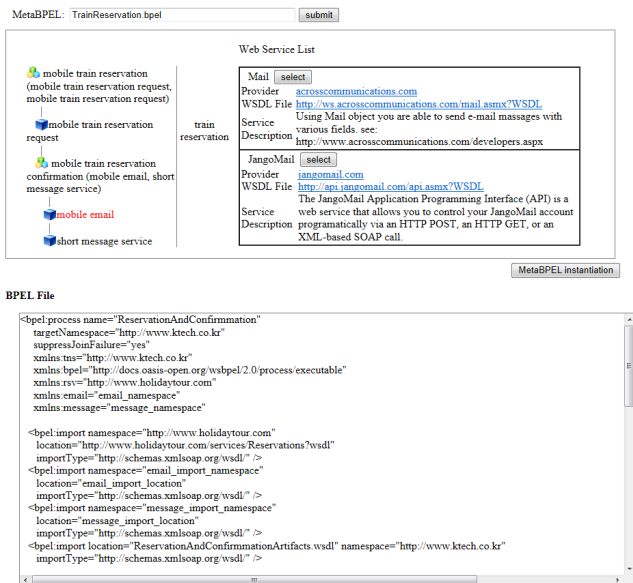
4. 구현

이 절에서는 XOT 를 사용하는 개념기반 오케스트레이션 시스템의 프로토타입을 구현하고 앞서 예들 들어 설명한 기차예약이라는 복합 서비스를 브로커가 처리하는 절차를 이 시스템을 통해 설명하도록 한다.

XOT 를 구축하고 관리하기 위해서 [그림 7]과 같은 XOT 관리기를 구현하였으며, 웹 서비스들의 WSDL 을 저장하고 인스턴스 웹 서비스들을 실제 추출하기 위해 XOT 와 연동되는 jUDDI 레지스트리를 구축하였다.



(그림 7) XOT 구현



(그림 8) 개념 기반 오케스트레이션 프로토타입 구현

이제, 새로운 복합 웹 서비스명 기차예약이 MetaBPEL 파서의 도움으로 MetaBPEL 에서 발견되었다고 하자. [그림 8]은 XOT 를 참조하여 브로커가 MetaBPEL 문을 순차적으로 처리한 후, 단계별로 사용자가 인스턴스 서비스들을 사용해 오케스트레이션하는 과정을 보여준다.

사용자가 마지막 웹 서비스까지 모두 선택을 끝내면, 최종적으로 브로커는 입력 MetaBPEL 문서에 대응되는 BPEL 문서를 생성하고 이를 사용자에게 제공한다.

5. 결론 및 향후 연구

본 논문에서는 XOT 를 이용한 개념 기반 오케스트레이션이라는 기법을 제안하였으며 이 기법의 타당성

을 보이기 위한 시스템 프로토타입을 구현하였다. 제안 기법은 구체적인 서비스명 대신 개념을 사용함으로써, 개념에 정합되기만 한다면 오케스트레이션 과정에서 실제 서비스들이 동적으로 대체될 수 있기 때문에, 앞서 언급한 대로 기 개발된 서비스들의 재사용성을 획기적으로 높일 수 있을 것으로 기대한다.

향후 연구로, 현재 수작업으로 구축한 XOT 각 개념 사이의 합성 패턴 관계성 명세를 반자동화하는 연구를 들 수 있다. 또 다른 연구로는 새로운 웹 서비스 합성을 할 때, 발생할 수 있는 다양한 합성 패턴을 XOT 내에 체계적으로 표현하는 작업도 필요하다.

참고문헌

- [1] Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design," Prentice Hall PTR, 2005.
- [2] D. L. Martin et al., "Bringing Semantics to Web Services : The OWL-S Approach," First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC), San Diego, CA, 2004.
- [3] D. Roman et al., "Web Service Modeling Ontology," Applied Ontology, vol. 1(1), Jan. 2005, pp. 77-106.
- [4] D. Kovac and D. Trcek, "A Survey of Web services Orchestration and Choreography with Formal Models," http://www.softec.si/pdf/kovac_damjan.survey.pdf, 2011.
- [5] M. Viroli. "A core calculus for correlation in orchestration languages," Journal of Logic and Algebraic Programming, vol. 70(1), 2007, pp. 74-95
- [6] R. Hamadi and B. Benatallah, "A Petri net-based model for web service composition," the 14th Australasian database conference, vol.17, 2003, pp. 191-200,
- [7] P. Bertoli, M. Pistore, and P. Traverso, "Automated composition of Web services via planning in asynchronous domains," Artificial Intelligence, vol.174(3-4), 2010, pp. 316-361
- [8] F. Lécué, E. Silva and L. F. Pires, "A framework for dynamic web services composition," 2nd ECOWS Workshop on Emerging Web Services Technology (WEWST07), Halle. Germany: CEUR Workshop Proceedings, Nov. 2007.
- [9] N. Srinivasan, M. Paolucci and K. Sycara, "An Efficient Algorithm for OWL-S Based Semantic Search in UDDI," First International Workshop on Semantic Web Services and Web Process Composition, 2004, pp. 96-110.
- [10] J. G. Bae, J. D. Yang, M. Y. Lee, "Design and Implementation of Expert Recommending System with Extended Object-Based Thesauri on Social Network Services," International Conference on Information Science and Applications (ICISA), April. 2011, pp. 1-7.
- [11] M. Popovici, M. Muraru, A. Agache, L. Negreanu, C. Giumale, C. Dobre, "An Ontology-Based Dynamic Service Composition Framework for Intelligent Houses," 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS), Mar. 2011, pp. 177-184.