

# 웹 서비스 컴포넌트를 이용한 소프트웨어 개발에서의 명세 확장 방법과 그 효과 분석

서광익<sup>o</sup> 이동근, 최은만  
 동국대학교

bradseo@dongguk.edu<sup>o</sup>, dongkun0606@hotmail.com, emchoi@dgu.ac.kr

## Specification Extension Method and Effect Analysis in Software Development Using Web Service Component

Kwangik Seo<sup>o</sup>, Dongkun Lee, Eunman Choi  
 Dongguk University

### 요 약

사용자는 웹 서비스 컴포넌트에 대한 발견뿐만 아니라 자신의 시스템에 맞게 동작할 수 있도록 명세를 참조하여야 구현 및 테스트 작업이 효과적으로 이루어진다. 따라서 컴포넌트 명세를 작성할 때 사용자 이해를 고려한 여러 요소가 포함되어야 하는데, 현재의 명세는 오퍼레이션 이름, 데이터 타입 등과 같이 API 수준의 정보를 제공하기 때문에 해당 컴포넌트에 대한 이해가 어려워 사용자의 요구에 적합한 것인지를 결정하기 힘들며, 구현 및 테스트 과정에서 제약, 선-후 조건, 품질과 같은 요소들이 빠져 있어 작업을 어렵게 한다. 따라서 본 연구에서는 앞서 제시한 문제들을 개선하고자 사용자 요구사항 기반의 명세 확장 방법을 제안하고, 이에 대한 효과를 명세 표준인 WSDL과 비교 분석함으로써 그 효용성을 보였다.

### 1. 서 론

웹 서비스 컴포넌트를 사용하기 위해 제공자는 명세를 작성하여 서비스 등록소에 등록(Publish)하고, 사용자는 그 컴포넌트를 찾는다. 이때 웹 서비스 컴포넌트에 대한 명세는 사용자가 찾고자하는 컴포넌트인지 확인하기 위한 중요한 참고자료가 되므로 충분한 내용을 담고 있어야 한다. 하지만 현재 웹 서비스 컴포넌트의 명세는 간단한 오퍼레이션 이름, 데이터 타입 등과 같이 API 수준의 정보만을 제공하는 동시에 기능적인 측면만을 명세하고 있어 사용자의 요구사항인 기능적·비기능적 요소와 일치하는 컴포넌트를 선택하기 어렵다[1]. 그리고 제공되는 기능적 요소 또한 추상적이고 비기능적 요소의 명세 정보가 미약해서 사용자가 원하는 웹 서비스 컴포넌트의 선택을 어렵게 한다[2]. 따라서 본 연구는 이러한 단점들을 개선하고자 사용자 입장을 고려한 요구사항 기반의 웹 서비스 컴포넌트에 대한 명세 확장 방법을 제안하고 이에 대한 효과를 분석하였다.

에서 시멘틱 웹 기술언어를 기반으로 웹 서비스 컴포넌트를 기술하기 위해 개발한 언어이며, 동시에 웹 서비스 컴포넌트의 발견, 실행 및 구성의 자동화를 목적으로 한다. DAML-S는 WSDL이 없는 의미적인 정보를 표현 및 포함하고 있지만, WSDL과 마찬가지로 기능 위주의 명세를 하고 있어 비기능적인 요소에 대한 기술이 없고 적합한 웹 서비스 컴포넌트의 선택 및 합성 등을 불가능하게 한다.

웹 서비스의 계약적 명세를 위해서는 사용자가 웹 서비스 컴포넌트를 정확히 이해하여 적합한 것을 선택하기 위한 기능, 품질, 하부 구조에 대한 정보를 필요로 한다. 그림 1은 WSDL과 DAML-S를 포함하여 기타 명세 표현이 제공해야 하는 요소들이 무엇인지를 구분하여 이를 정리했다[4].

Contract Category	Contract Type	Functionality				Quality		Infrastructure			
		Identify	Synthetic	Behavioral	Synchronic/Asynchronic	Compositional	QoS	Timing	Communication	Security	Management
Language											
WSDL		+	+								
BPEL4WS					+	+					
WS-CDL					+	+					
WS-Policy										+	
WSLA								+	+		+
WSOL					+			+	+		+
OWL-S		+	+	+	+						

그림 1 웹 서비스의 명세 언어 비교

웹 서비스 컴포넌트의 기능적인 검토사항으로는 컴포넌트의 제약사항들을 기술하는 구문과 오퍼레이션의 행위와 동기화 그리고 웹 서비스 컴포넌트들의 합성 단계에서 이루어지는 메시지 흐름이 있다. 그리고 비기능적인 부분

### 2. 웹 서비스 명세 언어와 문제점

WSDL(Web Services Description Language)은 웹 서비스 컴포넌트를 사용하기 위한 정보를 기술하는 표준이며 서비스가 하는 일, 서비스로의 접근 방법, 서비스가 위치한 곳과 같은 3가지의 속성을 기술한다. 현재의 WSDL은 명세 범위가 기능 위주이고, 웹 서비스 컴포넌트를 사용하는데 필요한 기능적·비기능적 요소들에 대한 의미적인 표현, 제약, 선-후 조건 등과 같은 기능적·비기능적 요소에 대한 정보가 부족해 사용자가 해당 컴포넌트를 정확히 이해하지 못해 구현 및 테스트를 어렵게 한다[3]

DAML-S(DARPA Agent Markup Language)는 DARPA

으로는 품질과 가격 그리고 프로토콜과 보안이 있을 수 있다. 그림 1에서는 위와 같은 검토사항과 이를 명세할 수 있는 언어와의 상관관계를 보이고 있다. 그 중 WSDL은 제공되는 정보가 극히 적고, DAML-S는 품질과 하부 구조에 대한 정보를 명세하고 있지 않는 것을 볼 수 있다. 이는 앞서 지적한 WSDL과 DAML-S의 동일한 문제점을 지니고 있다.

3. 요구사항 기반의 웹 서비스 컴포넌트 명세

사용자는 요구분석을 통해 얻은 기능적·비기능적 요구사항과 웹 서비스 컴포넌트 제공자가 제공하는 명세를 비교하여 적합한 웹 서비스 컴포넌트를 선택한다[5]. 따라서 웹 서비스 컴포넌트 제공자는 사용자 입장을 고려하여 자신이 제공하는 웹 서비스 컴포넌트가 어떤 요구사항을 만족하는지에 대하여 명세할 필요가 있다.

표 1의 기능적 요구사항은 컴포넌트가 가지고 있는 오퍼레이션과 관련된 세부 항목과 각 오퍼레이션의 관계를 나타내는 항목으로 정의하였다.

표 2의 비기능적 요구사항은 웹 서비스 컴포넌트에 대한 목적 설명, 사용 자격, 환경 요소, 가격, QoS(Quality of Service)에 대한 항목으로 정의하였다

표 1 기능적 요구사항

항목	설명
FunctionRequirement	기능적 요구사항 요소의 집합
Operations	Operation의 집합
Operation	Operation을 의미
Name	Operation의 이름
Role	Operation의 역할
Condition	Operation 실행을 위한 조건의 집합
Pre-Condition	실행 조건의 의미
Post-Condition	실행 후 조건을 의미
InputData	입력 데이터의 집합
InputDataType	입력 데이터의 타입
InputDataConstraint	입력 데이터의 제약
OutputData	출력 데이터의 집합
OutputDataType	출력 데이터의 타입
OutputDataConstraint	출력 데이터의 제약
PossibleException	발생 가능한 예외
RelationshipOfOperations	Operation들의 관계
OperationSequence	Operation들의 실행 순서

표 2 비기능적 요구사항

항목	설명
NonFunctionRequirement	비기능적 요구사항 요소의 집합
Service	서비스 컴포넌트 정보
Name	컴포넌트의 이름
Role	컴포넌트의 역할
Qualification	컴포넌트의 사용 자격
Environment	컴포넌트를 사용하기 위한 환경
Hardware	하드웨어 정보
Network	네트워크 정보
Security	보안 정보
Price	컴포넌트 사용에 따른 가격
Money	일정 기간에 따른 가격
Period	사용 기간
QoS	컴포넌트의 품질 정보
Performance	요청에 대한 처리 성능 정보
ResponseTime	응답 시간
MaximumThroughout	최대 처리량
Stability	안정성
Availability	이용가능성
Reliability	신뢰성
Accessbility	접근 가능성
MeasurementMetric	품질 항목에 따른 측정 메트릭
AverageValue	메트릭을 통한 각 항목의 품질 평균치

4. 사례 연구 및 효과 분석

4.1. 명세 단계에서의 요구사항 이해

다른 사람에 의해 개발된 웹 서비스 컴포넌트를 이해하여 사용한다는 것은 쉬운 일이 아니다. 블랙 박스 형태로 만들어진 컴포넌트 내부를 사용자가 알 수 없기 때문에 사용자는 컴포넌트에 대한 정보를 상세히 제공받아 이것을 토대로 컴포넌트를 이해하여 사용 및 유지 보수를 할 수 있는 것이기 때문에 사용자의 이해를 돕기 위한 정보를 상세히 제공할 필요가 있으며, 본 연구에서는 이 정보들을 기능적·비기능적 요소들로 보고 이를 비교하며 방법은 다음과 같다.

- a. 비교 대상 - WSDL, DAML-S, 제안된 명세
- b. 비교 구분 - 기능적 요소와 비기능적 요소
- c. 비교 세부항목
  - 기능적 요소 : 웹 서비스의 기능적 요구사항과 관련된 요구와 웹 서비스의 구현 및 테스트와 관련된 연구에서 공통적으로 제시하는 기능적 요소를 선정
  - 비기능적 요소 : 웹 서비스의 비기능적 요구사항과 관련된 연구와 웹 서비스 품질 및 테스트 관련 연구에서 공통적으로 제시하는 비기능적 요소를 선정

표 3 요구사항 명세 비교

대상 구분	세부 항목	WSDL	DAML-S	제안된 명세
		제공 여부	제공 여부	제공 여부
기능 요소	Operation Name	√	√	√
	Operation Role		√	√
	Pre Condition		√	√
	Post Condition			√
	Input/Output Data Type	√	√	√
	Input/Output Data Constraint			√
	Possible Exception			√
	Relationship of Operations		√	√
	Synchronization		√	
	Compositional		√	
비기능 요소	Service Name	√	√	√
	Service Role		√	√
	Qualification		√	√
	Environment			√
	Price			√
	QoS			√
결 과	제공 여부 수 / 전체 세부항목 수 *100	20%	66%	86%

표 3에서 현재 웹 서비스 컴포넌트 명세 표준인 WSDL은 기능적 요소와 비기능적 요소에 대한 부분이 DAML-S나 본 논문에서 제안한 명세보다는 훨씬 적은 항목들을 명세하고 있는 것을 볼 수 있다. 이것은 현재의 WSDL이 컴포넌트를 사용하기 위한 기능적·비기능적 요소들에 대한 명세가 부족하여 사용자의 이해를 어렵게 하는 것을 의미한다고 볼 수 있다. DAML-S는 기능적 측면에서는 많은 요소들을 제공하고 있지만, 제약사항이나 발생 가능한 예외 등에 대한 부분들이 빠져 있으며, 사용자로서 중요시 여기는 Price와 QoS에 대한 요소는 명세하고 있지 않은 것을 볼 수 있다. 이런 비기능적 요소들은 소프트웨어 컴포넌트 이해를 위한 모델 연구[6]에서도 지적하듯이 컴포넌트를 이해하는데 중요한 요소가 되지만 이에 대한 명세가 없어 비기능적인 측면에서의 이해는 떨어질 수밖에 없다. 마지막으로 본 논문에서 제안한 명세는 명세하고 있는 요소들이 사용자의 기능적·비기

능적 요구사항을 위해 정의된 항목들로 구성되어 있어 WSDL이나 DAML-S 보다 많은 요소들을 명세할 수 있어 컴포넌트에 대한 기능적 측면이나 비기능적 측면에 대한 이해가 높다고 볼 수 있다.

4.2. 인터페이스 구현 단계에서의 명세 이용

사용자가 웹 서비스 컴포넌트를 사용하기 위해서는 명세를 바탕으로 사용자의 다양한 플랫폼에 맞는 인터페이스를 구현하여 이를 통해 기능 및 품질 테스트 과정을 거치게 되는 것이다. 따라서 명세는 웹 서비스 컴포넌트를 사용하기 위한 인터페이스 구현을 위해 보다 쉽고, 많은 정보를 제공할 필요가 있다. 따라서 본 연구에서는 인터페이스 구현에 따른 WSDL과 제안한 명세를 비교하여 그 효과를 분석한다. 구현 과정에서 명세를 비교하기 위한 방법은 다음과 같다.

- a. 비교 대상 - WSDL, 제안된 명세
- b. 비교 범위 - 인터페이스 구현
- c. 비교 방법 - 배송 관리 웹 서비스 컴포넌트의 인터페이스를 WSDL과 본 논문에서 제안한 명세를 매핑하여 비교한다.

그림 2는 본 연구를 위한 실험 시스템의 웹 서비스 컴포넌트를 사용하기 위해 구현된 인터페이스와 WSDL과의 매핑을 나타내고 있다.

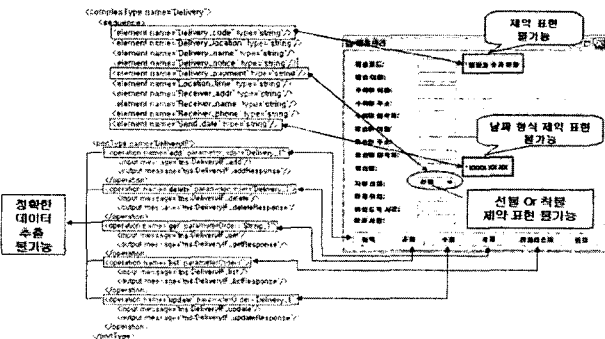


그림 2 인터페이스와 WSDL의 매핑

왼쪽은 웹 서비스 컴포넌트의 인터페이스정의와 관련된 portType과 types 부분이고, 오른쪽 이는 웹 서비스 컴포넌트를 사용하기 위한 인터페이스를 구현한 것이다. 사용자가 오른쪽과 같은 인터페이스를 오직 명세만을 참조하여 자신의 플랫폼에 맞게 만든다고 가정했을 때 이 둘을 비교해 보면 그림에서 지적하듯이 입력 데이터에 따른 제약사항들을 표현하지 않고, 각각의 오퍼레이션들에 대한 기능 및 선·후 조건등과 같은 데이터를 추출할 수 없는 것을 볼 수 있다. 따라서 사용자는 WSDL에서 제공하는 정보만을 가지고 인터페이스를 구현한다면 이러한 부분에 대한 구현은 불가능하다.

그림 3의 왼쪽은 본 논문에서 제안한 요구사항 기반의 명세 내용 중 배송 등록 오퍼레이션에 관련된 명세 결과이다. 앞서 살펴본 WSDL과는 달리 제약사항들에 대한 표현이나 입력 데이터에 대한 1:1 대응, 오퍼레이션에 대한 의미적 표현과 선·후 조건 기술로 인해 WSDL보다는 인터

페이스의 정의를 더 구체적으로 구현하고 있음을 보이고 있다.

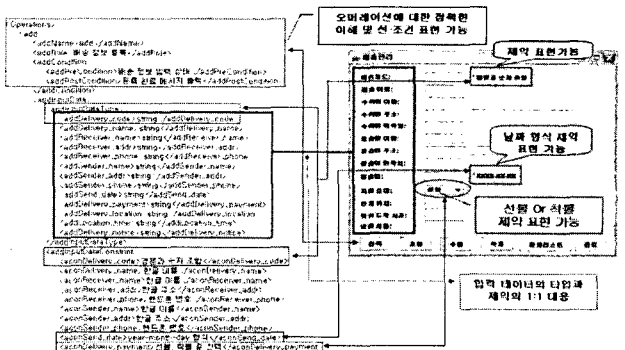


그림 3 제안한 명세와 인터페이스 매칭

5. 결론

사용자가 웹 서비스 컴포넌트를 효율적으로 사용하기 위한 명세가 무엇보다도 중요하다. 본 연구를 통해 현재 웹 서비스 컴포넌트의 명세들이 가지고 있는 문제점들을 지적하고, 이를 개선할 수 있는 방법으로 요구사항을 기반으로 하는 명세 확장 방법에 대해 연구하여 사용할 웹 서비스 컴포넌트에 대한 이해를 돕고, 기존 명세 보다 기능적·비기능적 요구사항들을 상세히 기술함으로써 적합 웹 서비스 선택을 유도함과 동시에 보다 쉽게 인터페이스를 구현하고 테스트함으로써 그 효과를 비교 분석하였다.

6. 참고문헌

- [1] Gold, N., Bennett, K., "Program Comprehension for Web Services", Proceedings of 12th IEEE International Workshop on Program Comprehension, pp.151-160, 2004.
- [2] Degwekar, S., Su, S.Y.W., Lam, H., "Constraint Specification and Processing in Web Services Publication and Discovery", Proceedings - IEEE International Conference on Web Services, pp.210-217, 2004.
- [3] Heckl, R., Lohmann, M., "Towards Contract-based Testing of Web Services", Electronic Notes in Theoretical Computer Science, Vol. 116, pp.145-156, 2005.
- [4] Tosic, V., Pagurek, B., "On Comprehensive Contractual Description of Web Services", The 2005 IEEE International Conference on 29 March-1 April, pp.444-449, 2005.
- [5] Simeon Simeonov, Building Web Services with Java: Marking Sense of XML, SOAP, WSDL and UDDI, SAMS, 2001.
- [6] Andrews, A., Ghosh, S., Choi, E., "A Model for Understanding Software Components," International Conference on Software Maintenance, pp.359-368, 2002.