

# 비즈니스협업을 위한 웹서비스 표준에 관한 연구

김훈태\*, 임태수\*\*, 정재윤\*\*, 강석호\*\*

## A research on the Web service standards for business collaboration

Hoontae Kim, Taesoo Lim, Jae-yoon Jung, Suk-ho Kang

### 요 약

비즈니스 협업을 위한 방법들 중에서 메시지 교환을 통한 loosely coupled integration이 선호되고 있는 추세에서, 이를 위한 방법으로 최근에 각광받고 있는 기술이 웹서비스이다. 웹서비스를 확장한 협업 표준들이 제안되고 있는데, 본 연구에서는 동적인 웹서비스 운용을 위한 WSCI, BPEL4WS, BPML 등을 비교 분석하고자 한다.

핵심어: 웹서비스 구성법, WSCI, BPEL4WS, BPML

### Abstract

Business collaboration can be achieved by message exchange between companies. Such an exchange method enables loosely coupled integration between heterogeneous organization, and Web service has been a popular technique for accomplishing it. This research focuses on the choreography standards extending Web Service, specifically WSCI, BPEL4WS, and BPML. We compare and analyze the standards in terms of functionality, expressiveness and so on.

*Key Word* : Web Service Choreography, WSCI, BPEL4WS, BPML

---

\* 대전대학교 산업시스템공학과

\*\* 서울대학교 산업공학과

## 1. 서론

기업간 협업유형은 각각의 관점에 따라 여러 가지로 분류될 수 있다. 협업유형은 적용된 정보통신기술의 수준에 따라 공급사슬형 기업, 확장형 기업, 가상 기업 등으로 분류될 수 있다. 장기적인 계약관계를 토대로 한 정보 시스템의 긴밀한 연계를 특징으로 한 공급사슬형 기업에서부터 일시적인 만남과 해체를 특징으로하는 가상 기업형까지 여러 스펙트럼이 존재할 수 있다[5]. 즉, 기업간 협업 방식의 다양한 형태를 지원할 수 있기 위해서는 긴밀한 연계뿐만 아니라 유연한 연계 형태도 지원할 수 있어야 한다.

웹서비스는 메시지 교환을 통해 유연한 연계방식을 지원할 수 있는 표준으로 자리 잡고 있는 추세이다. 웹서비스는 기술적인 관점에서 매우 유연한 결합을 가능하게 해준다. 객체지향적인 관점에서 서비스에 대한 응용 인터페이스만 알면 구체적인 작동방식을 이해할 필요 없이도 이용할 수 있는 비동기적인 상호작용을 지원해 준다.

웹서비스는 이렇듯 기술적인 상호운용성을 제공하지만, 의미론적인 상호운용성에 대한 지원 미비로 진정한 비즈니스 협업 도구로서는 한계가 인식되어왔다. 비즈니스 모델관점에서 볼 때, 기업내 통합이나 패쇄적 비즈니스 모델 수준에서는 활용될 수 있으나, 사용자에 대한 인식이 어려워지면서 비즈니스 협의를 필요로 하는 반개방적 비즈니스나 개방적 비즈니스 모델에는 적합하지 않는 측면이 있다[1].

웹서비스에서 서비스를 기술하는 표준인 WSDL(Web Service Description Language)은 구조적으로 상태 정보를 포함하지 못한다. 교환되는 메시지의 상태 정보 부재는 기업

간 비즈니스 트랜잭션에서 흔히 발생하는 비동기적 메시지 교환이나, 장기 트랜잭션(long-lived transaction) 등을 처리하는데 효과적이지 못하다[7]. 웹서비스는 이외에도 여러 웹서비스들이 발생하는 순서정보, 트랜잭션 처리, 런타임시의 의사결정 등을 지원하지 못하는 단점 등이 있다[6].

이러한 단점들에 대한 보안을 통해 진정한 비즈니스 협업 도구로서 웹서비스를 확장하고자 여러 웹서비스 구성표준들이 도입되어 상호 경쟁하고 있는 추세이다. 본 논문은 그 중 가장 대표적인 세 개의 표준인 BPEL4WS(Business Process Execution Language for Web Service), BPML(Business Process Modeling Language), WSCI(Web Service Choreography Interface) 등에 대한 비교 분석을 통해 웹서비스 구성언어 표준들에 대한 최신 흐름을 소개함으로써 국내 표준화 작업에 도움을 주고자 수행되었다.

## 2. 비즈니스 서비스 연동과 실행

BPEL4WS, BPML, WSCI 등의 각 표준 구조를 최상위계층에서 개괄적으로 살펴보면 각각 <표1>, <표2>, <표3>과 같다.

세 가지 표준은 각기 다른 형태를 취하고 있으면서 많은 부분에서 중복된 구조를 취하고 있다. BPMI.org에서는 BPEL4WS가 BPML의 부분집합이고, WSCI는 BPML의 보완적인 역할을 수행하고 있다고 언급한다.

2장에서는 세 표준이 WSDL과 연동하는 방식과 웹서비스 구성에 필요한 핵심 요소들을 살펴보기로 한다.

<표 1> BPEL4WS 개괄구조

```

<process>
  <partnerLinks>...</partnerLinks>
  <partners>...</partners>
  <variables>...</variables>
  <correlations>...</correlations>
  <faultHandlers>...</faultHandlers>
  <compensationHandlers>...
    </compensationHandlers>
  <eventhandlers>...</eventhandlers>
  {any activity}
</process>

```

<표 2> BPML 개괄구조

```

<package>
  <documentation>...</documentation>
  <feature>...</feature>
  <import>...</import>
  <identity>...</identity>
  <process>
    <event>...</event>
    <parameters>...</parameters>
    <context>...</context>
    {any activity}
    <compensation>...</compensation>
  </process>
  <property>...</property>
  <schedule>...</schedule>
</package>

```

<표 3> WSCI 개괄구조

```

<interface>
  <documentation>...</documentation>
  <process>
    <documentation>...</documentation>
    <context>...</context>
    {any activity}
  </process>
</interface>

```

### 2.1 웹서비스와의 연계

웹서비스 구성에 사용되는 활동(activity)는 크게 simple activity와 complex activity로 분류된다. Simple activity는 더 이상 조개질 수 없는 원자적인 단일 활동이며, complex activity는 다른 활동들로 구성되는 복합활동이다.

비즈니스 구성언어들은 simple activity를 통해 WSDL에서 정의된 operation들과 연계된다고 볼 수 있다.

BPEL4WS는 메시지를 수신하는 receive 활동이 가장 기본적인 활동이다. 이 액티비티를 사용할 때에는 크게 네 가지 정보(partnerLink, portType, operation, variable)를 속성값으로 지정한다. 이 중에서 portType과 operation은 WSDL 명세에서 제공하는 요소를 참조하고 있는 것으로 이를 통해 웹서비스와 연동하게 된다. 위의 네 가지 정보(partnerLink, portType, operation, variable)는 비즈니스 프로세스에서 웹 서비스를 호출하거나 호출받을 때, 어떤 역할(partnerLink)을 하는 누가(portType) 어떻게(operation) 어떤 정보(variable)를 주고받는 지, 그리고 물리적으로 어떻게 지원하는 지를 명시하도록 자원한다. 다른 simple activity는 <표 4>에 기술하였다.

BPML은 WSDL에 정의된 portType과 operation을 직접 참조하여 웹서비스와 통신한다. 이 때 주고받는 데이터는 input과 output을 이용하여 정의할 수 있다. 이 데이터들은 BPML 문서에서 정의된 특성값(property)들과 연결되어 있어서 다른 곳의 액티비티나 프로세스에게 값을 전해주는 역할을 한다.

WSCI도 BPML이나 BPEL4WS와 유사하게 WSDL에 정의된 portType과 operation을 직접 참조하여 웹서비스와 연계된다. 그러나, BPML과 달리 input/output을 사용하지 않는 데 이는 WSCI가 BPML이나 BPEL4WS가 표방하는 기업내부프로세스(executable business process)를 지원할 목적이 아니기 때

문이다. 그리고, 웹서비스와의 연계 측면에서 BPEL4WS는 partnerLink, partnerLinkType, portType으로 연결되는 흐름에서 다소 중복적으로 WSDL과 연계되는 측면이 발견된다. 각 표준에서 지원하고 있는 simple activity들은 <표 4>와 같다.

<표 4> simple activities

구분	Simple activities
BPEL4WS	Receive, reply, assign, throw, wait, empty, pick, compensate
BPML	Action, assign, call, compensate, delay, empty, fault, raise, spawn, synch
WSCI	Action, delay, empty, fault, call, spawn, join

세 표준 모두 참여자들 간의 프로세스를 message, portType, operation과 같은 논리적 요소를 통해 설계하고 있으며, 물리적인 프로토콜 바인딩이나 구체적인 서비스 방식 등은 WSDL에 위임하고 있다.

## 2.2 비즈니스 구성법

프로세스 표현 방법은 크게 블록 구조와 방향성 그래프 두 가지로 나눌 수 있다. 블록 구조의 프로세스는 프로그래밍 언어에서 구현하고 작동하기 쉬운 반면, 방향성 그래프로 표현된 프로세스는 분석가나 설계자가 이해하고 다루기 쉬운 장점이 있다.

BPML과 WSCI는 블록 구조로 프로세스를 표현한다. 계층적으로 구성되는 블록 구조는 프로세스의 선언, 정의, 수행에서 중요하게 사용된다. BPML로 작성된 프로세스의 흐름 통제는 모두 블록 구조에 의해 처리된

다. BPML에서는 블록의 흐름을 표현하기 위하여 복합활동 (complex activity) 이라고 부르는 7가지 활동(all, choice, foreach, sequence, switch, until, while)들을 지원한다. WSCI도 BPML과 동일한 7가지의 복합활동을 지원한다.

BPEL4WS는 블록 구조를 중심으로 설계하지만, 방향성 그래프의 설계 방법을 적절히 내포하고 있다. flow 구조를 통하여 블록 형태의 설계를 구축한 다음, 블록 구조 내에 포함된 활동간의 임의의 의존 관계를 지정하는 링크를 이용하여 방향성 그래프의 이점을 가미하고 있다. 그러나 순환하는 것을 배제하고 구조적 경계를 넘는 것을 제한하고 있다.

<표 5> complex activities

구분	BPML/WSCI	BPEL4WS
병렬구조	All	Flow
직렬구조	Sequence	Sequence
분기구조	Choice, Switch	Switch
반복구조	Until, While, Foreach	While
기타		Scope

BPEL4WS에서는 sequence, switch, while, flow, scope 등의 5가지 복합활동을 지원한다. sequence는 직렬 구조를 표현하고, switch는 제약값에 따른 OR 분기를 표현하며, while은 반복구조, flow는 AND 분기의 병렬 구조를 표현한다. 그리고 scope는 특정한 문맥 (context)을 공유하기 위한 블록을 표현한다. BPML/WSCI와 BPEL4WS의 복합활동을 비교하면 <표5>와 같다.

## 2.3 비즈니스 프로세스 초기화 방법

BPML이나 WSCI에서는 비즈니스 프로세스를 초기화하는데 활동을 사용하는 방법과 메시지 수신을 통한 방법을 둘 다 지원한다. 첫 번째로 call, compensate, spawn 활동을 사용하여 프로세스를 초기화시킬 수 있는데, 주로 긴밀히 결합되어 있는 동일 기업의 하위 프로세스를 초기화하는데 이용된다고 볼 수 있다. 두 번째로는 정해진 입력 메시지에 의해 프로세스가 시작될 수 있다. 이러한 경우는 다른 기업의 요청에 의해 시작될 때 주로 사용되는 것으로서 웹서비스가 수신으로 시작하는 one-way나 request-response 형태로 정의되어 있어야 사용이 가능하다.

BPML에서는 이와 달리 오로지 정해진 메시지를 수신함으로써만 비즈니스 프로세스가 초기화될 수 있다. 앞에서 기술한 simple activity인 receive 활동을 통해 프로세스를 초기화할 수 있다. receive 엘리먼트에는 프로세스를 초기화시키는 메시지가 어떤 역할(partnerLink)을 하는 누가(portType) 어떻게(operation) 보내며, 어떤 정보(variable)를 포함하고 있는지 정의되어 있다.

## 3. 프로세스 확장 요소

### 3.1 예외 처리

비즈니스 프로세스는 종종 장기간동안 수행되거나 비동기적 메시지(asynchronous message) 통신을 사용하는 경우가 많다. 이러한 환경에서는 신뢰적인 이유나, 장기간 락(lock)이나 격리(isolation)를 할 수 없다는 이유로 인하여 ACID 트랜잭션이 지역적으

로 제한된다. 그 결과 비즈니스 프로세스에서의 예외 처리는 보상(compensation)이라는 개념으로 처리하는 데 의존할 수 밖에 없다. 보상이란 단념한 커다란 활동에 속하는 이미 수행된 활동들의 영향을 회복시키는 활동이라고 할 수 있다.

BPML과 WSCI에서는 위와 같은 장기 수행 트랜잭션(LRTs; Long-Running (Business) Transactions)을 처리하기 위하여 오류 처리(fault handling)과 보상(compensation)을 정의할 수 있다. throw 활동을 통해 예외상황시 오류를 발생시킬 수 있으며, compensate 활동을 통해 이미 완료된 특정 구역을 보상할 수 있도록 지원한다.

BPML에서도 유사하게 exception과 compensation이라는 특이한 프로세스를 사용하여 예외처리를 지원한다. 즉 이들은 일반적인 프로세스나 중첩 프로세스와 같이 자신만의 컨텍스트나 이벤트, 구성 액티비티들을 가지고 있으며, 예외나 보상이 필요할 때 exception이나 compensation 프로세스를 처리하도록 한다.

### 3.2 트랜잭션 관리

BPML과 WSCI는 단일 트랜잭션과 장기 트랜잭션을 모두 포괄적으로 지원한다. 보상 액티비티는 조정 트랜잭션과 확장 트랜잭션 모두와 연결될 수 있어서, 트랜잭션이 취소되면 동일한 컨텍스트 내의 모든 보상 액티비티가 역방향으로 수행되며 보상을 진행한다. 또한, 중첩 트랜잭션을 허용하여 하나의 중첩 트랜잭션은 내부에 여러 개의 단일 트랜잭션이나 확장 트랜잭션을 포함할 수 있다.

BPML4WS 명세서는 직접적으로 트랜잭션에 대해 언급하지 않으며, 벤더가 특정한 트랜잭션 처리 프로토콜을 의무적으로 지원하도록 요구하지 않는다. 대신에 보완적인 표준언어인 WS-Transaction에서 트랜잭션에 관한 자세한 사항을 다루고 있다.

#### 4. 결론

웹서비스 구성 방식은 조정(orchestration), 구성(choreography), 협업(collaboration) 등으로 분류되기도 한다[4]. 조정은 기업내부의 프로세스 관리를 위한 것이라면, 구성은 기업간 상호 작용을 염두에 둔 것이다. 조정과 구성이 웹서비스의 하부 구조에서 출발한 개념이라면 협업은 비즈니스 상부에서 출발했다고 볼 수 있다. BPML과 BPML4WS는 다소 조정에 초점을 맞춘 표준인데 반해, WSCI는 구성에만 초점을 맞춘 표준이라고 볼 수 있다. BPML과 BPML4WS는 거래 당사자의 쌍방이 하나의 정의된 프로세스를 같이 공유하고 활용하는 관점이라면, WSCI는 하나의 거래 당사자의 인터페이스 관점에서 웹서비스 구성을 기술한다. 전체적인 연결은 global 모델을 통해 따로 제공한다.

BPML과 WSCI는 상보성을 표방하는 만큼 많은 부분에서 동일한 모습을 보여 준다. 그리고, BPML/WSCI는 블록구조로 중첩구조를 표현하고자 하는데서 구현상의 복잡함을 야기하고 있는 반면, BPML4WS는 중첩구조를 허용하지 않는 측면에서 구현이 용이하나 프로세스의 재활용 측면에서 한계를 드러낸다.

#### 참고 문헌

- [1] 정보통신정책, 제14권 15호, 2002.
- [2] Business Process Execution Language for Web Services 1.1, May, 2003.
- [3] Business Process Modeling Language, November, 2002.
- [4] McDonald, C., Orchestration, Choreography, and Collaboration, Sun Microsystems, 2003.
- [5] Jagdev, H. S., and Thoben, K.-D., Anatomy of enterprise collaborations, Production Planning and Control, 12 (5), 437-451, 2001.
- [6] Web Service Choreography Interface 1.0, August, 2002.
- [7] Wil van der Alast, Don't go with the flow: Web services composition standards exposed. Web Services - Been there done that?, Jan/Feb , IEEE intelligent systems, 72-76, 2003.