

# BPML과 XPDL간의 상호 운용성 연구

임종선\*, 주경준\*\*, 주경수\*  
\*순천향대학교 정보기술공학부  
\*\*주) Golden Pond  
e-mail:ronmer@chol.com

## A Study on Interoperable between BPML and XPDL

JongSeon Lim\*, Kyung-Jun Joo\*\*, Joo Kyung-Soo\*  
\*Dept of Computer Science, Soonchunhyang  
\*\*(Co.) Golden Pond

### 요 약

프로세스 통합을 사용하면 이질적인 시스템과 조직에서 비즈니스 프로세스를 모델링하고 자동화할 수 있다. 프로세스 통합은 흔히 B2B 거래소 연결, 제품과 서비스의 온라인 구매, 고객 정보의 표준화, 또는 새로 구입한 애플리케이션 통합 따위의 많은 비즈니스와 기술 정책의 핵심이다. 통합 프로세스는 비즈니스 전략의 구현을 보다 용이하게 한다는 것이다. 현재에는 기업 내부의 프로세스 통합은 많이 이루어지고 있으며, 기업과 기업간의 데이터 전송을 위한 방법들이 많이 연구되고 있다. 그러나 기업과 기업간의 통합을 이루기 위해서는 데이터 통합 뿐만아니라 프로세스 통합을 이루어야 한다.

본 논문에서는 기업간 프로세스 통합을 위하여 BPML과 XPDL 상호간의 운용성을 연구하였으며, 차후에 이를 구현할 프로그램을 개발할 것이다.

### 1. 서론

전자상거래가 이루어지기 위한 기술적 요소 중 기업 통합 기술과 관련된 표준으로는 기업의 정보 및 서비스 표현과 기업간 거래 협약을 위한 표준, 기업과 기업간의 프로세스에 대한 통합 방법, 거래 시스템과 기업 내부 정보시스템간의 프로세스 통합 지침 등이 있다. 기업정보 표현 및 거래 협약에서 기업의 정보 및 서비스 표현과 기업간 거래 협약을 위한 표준을 위한 로드맵을 제시한다. 또한, 워크플로우에서 기업과 기업간 프로세스에 대한 통합 방법을 위한 로드맵을 제시하고, EAI(Enterprise Application Integration)에서는 거래시스템과 기업내부 정보시스템간의 프로세스 통합 지침을 위한 로드맵을 제시한다[5].

현재, 기업 내부의 프로세스 통합은 많이 이루어지고 있으며, 기업과 기업간의 데이터 전송을 위한 방법들이 많이 연구되고 있다. 그러나 기업과 기업간의 통합을 이루기 위해서는 데이터 통합 뿐만아니

라 프로세스 통합을 이루어야 한다.

본 논문에서는 기업간 프로세스 통합을 위하여 BPML과 XPDL 상호간의 운용성을 연구하였으며, 차후에 이를 구현할 프로그램을 개발할 것이다. 2장에서는 관련 연구 및 기술을 소개하며, 3장에서는 BPML과 XPDL간의 교환방법에 대하여 소개하며, 마지막으로 4장에서는 결론을 소개한다.

### 2. 관련연구 및 기술

#### 2.1 BPML

BPML이란 XML이 비즈니스 데이터용 메타언어인 것처럼, 비즈니스 프로세스(business process)용 메타언어이다. BPML은 e-Business의 공용 인터페이스로 이루어진 과정이며 되도록 많은 개별적 응용 도구들을 프로세스에 참여시키고자 한다. 이것이 가능한 것은 BPML의 공용 인터페이스 과정이 ebXML비즈니스 프로세스처럼, 혹은 RosettaNet처럼

럼 독자적으로 그 자체의 응용기술을 가졌기 때문이다. BPML은 통제흐름, 데이터 흐름, 이벤트 흐름에 관여하는 비즈니스 프로세스를 대표하는 동시에 비즈니스 규칙, 보안역할 그리고 거래의 맥락에서 결정적인 구성역할을 갖고 있다. BPML은 프로세스 지향적 기업전산화를 향한 현존 응용기술의 중심점을 매개한다. 또한 BPML은 동시적/비동시적 유통거래의 확실한 지지기반으로서 프로세스의 한 요소인 e-Business 프로세스내의 현존 응용기술을 포함하는 실행모델로서 사용될 수 있다.

BPML의 중요한 초점은 웹서비스를 정의하는 것이다. Activity 타입은 메시지 교환, 이벤트 핸들링, 보수(실패시), 지연에 대하여 명세하며, Attribute는 인스턴스의 상호관계, 메시지의 일부분을 실행, 위치 서비스 인스턴스를 지원하고, 트랜잭션, 블록 문법 구조의 지원, 예외 처리 및 수정을 지원한다[1][2].

### 2.3 XPDL(XML Process Definition Language)

XPDL은 블록의 조작으로 첨가된 개념과 함께 그래프 구조의 언어처럼 작성되어지며, 범위는 패키지 와 프로세스 레벨에 관련된다. 라우팅은 동작사이의 전송의 명세에 따라서 조작된다. 프로세스내의 동작은 직접적인 그래프의 노드와 같게 될 지라도, 예지로 변환된다. 컨디션은 activity와 activities들의 다음에 실행되는 실행 시간에 의하여 전송의 결정이 연관된다.

XPDL의 중요한 이슈는 작업의 적절한 분산에 있다. Activity 어트리뷰트는 동작을 실행하는데 필요한 리소스를 명세하고, 이것은 리소스의 요구를 결정하는데 필요한 경우 실행 시간을 구하는 표현이다. Activity 어트리뷰트는 애플리케이션을 실행하는데 필요한 동작을 명세한다. 이 컨셉은 리소스 개념, 애플리케이션간의 합동, 동작의 실행을 통합하여 지원한다[3][4].

## 3. BPML과 XPDL간의 교환방법

### 3.1 Definition과 Recursive 블록 구조

BPML의 정의와 선언은 블록 구조 범위와 관계된 것을 사용하여 확장하여 작성하였다. 복잡한 동작은 activity 집합을 참조하였으며, 문법적인 연관을 가지고 있다. 문법에서는 특징들의 선언이나 재선언, 프로세스(nested 프로세스)의 선언 및 재선언, 예리 조작, 트랜잭션과 접속자(메시지 핸들링에 관한)들이 가능하다. 복잡한 activity가 activity 집합 내에 나타

날 수 있다. XPDL은 단지 최상위 레벨의 프로세스를 정의하는 것을 지원하므로 nested 프로세스가 아니다. 워크 플로우의 적절한 데이터는 최상위 레벨 이전에 선언되거나 프로세스 정의에서 선언되며, 두 개의 레벨로 제한된다[4].

BPML의 복잡한 activity들에 대한 특징의 대부분은 블록 activity에 의하여 XPDL에 표현될 것이다. 이 구조는 외부 집합을 가지지 않는 activity들의 집합에 대한 이름에 의하여 표현된다. 인스턴스에 의해, BPML의 모든 activity에 대한 생성 프로세스는 activity 집합의 첫 번째나 마지막 activity에서 간단하게 다루게된다. 첫 번째 activity는 andsplit이 되고, 마지막 activity는 andjoin이 될 것이다. 이것은 첫 번째 activity에서부터 각각 다른 activity들의 블록과 각각의 activity에서부터 마지막 activity로 변환될 수 있다. 복잡한 activity의 다른 타입은 적당한 제어 로직으로 구현되는 변환에 의해 재표현 된다[2].

### 3.2 전문화된 자동 activities

BPML은 자동 activity들의 수를 포함하며, 일부는 그들의 명확한 기능을 지원하는 특별한 어트리뷰트의 집합을 요구한다. XPDL에서는 sub-flow, 동기화, 비동기화에 의한 구현, 애플리케이션에 의한 구현, 라우팅 activity, 블럭 activity와 같은 기본적인 몇 개의 타입이 있다.

XPDL activity 내의 자동 BPML activity는 자연적인 맵이다. BPML은 애플리케이션의 개념을 가지고 있지 않다. BPML Call activity는 동기적인 sub-flow에 의해 XPDL activity 구현과 동일하다. BPML Spawn activity는 어떤 특별한 계산 관점을 제외한 XPDL 비동기 sub-flow와 동일하다. 다른점으로는 첨가된 어트리뷰트는 특수화된 BPML에 의하여 XPDL로 전송되는 필요한 정보를 반드시 사용될 것이다.

### 3.3 Activities and attributes for WSDL Message

BPML은 WSDL을 기반으로 만들어 졌으며, WSDL의 명확한 activity를 가지고 있다. 적당한 어트리뷰트에서 WSDL 메시지를 이용하는 어트리뷰트를 가진다. XPDL activity들은 전유 어트리뷰트가 첨가된 것과 같은 종류의 확장을 할 것이다.

| BPML   | XPDL  |
|--|---|
| <b>action</b> - 표준 WSDL 메시지 패턴을 구현.<br>One-way, request-response, solicit-response, notification   | 첨가된 어트리뷰트를 요구할 것이다. (어플리케이션/롤 그리고 라이브러리 기능들)                              |
| <b>correlation</b> - 프로세스의 올바른 실행에 매칭되는 메시지를 사용  | Workflow에 관련된 데이터는 명확한 인스턴스이다. 상호관계 어트리뷰트는 올바른 인스턴스를 식별하여 사용하는 것을 가질 것이다. |
| <b>locator</b> - 올바른 서비스를 찾는 것을 사용. WSCI 스펙을 적용.   |   |
| <b>call</b> - action은 activity들의 정해지지 않은 집합을 실행한다. 단지 만약 의미가 필요로 한다면, 그 activity들은 완벽한 action에 대한 명령에서 수행. WSDL request-response 연산의 수행을 명세. |   |
| <b>output, selector</b> - 특정 값으로부터 구조화된 메시지를 사용.   |   |
| <b>connector</b> - Global Model 서비스의 내부동작에 대한 명세를 사용.  |   |

<표 1>Activities and attributes for WSDL Message의 변환

### 3.4 Web Services 통합

비즈니스 프로세스는 두 가지 방법으로 기술 될 수 있다. 실행 가능한 비즈니스 프로세스 모델은 비즈니스 내부 동작 내에 참여하는 행동이다. 이에 대조하여 비즈니스 프로토콜은 프로세스 정의를 사용한다. 프로세스 정의는 나타나있는 내부행동이 없을 경우 프로토콜에 포함된 각 그룹의 명백한 메시지 교환 동작을 수동으로 열거해야 한다. BPEL4WS는 실행가능하고 추상적인 프로세스 각각의 행동 모델을 사용하는 것을 의미한다.

### 3.5 Specialized Complex Activities

복잡한 BPML activity들은 activity의 집합에 속하며 어떤 타입의 라우팅 로직이나 완전한 집합에 관한 flow를 포함하며, XPDL1.0은 activity 집합에 속하는 블록 activity를 포함하며, 블록 activity는 어트리뷰트가 될 것이다. activity 집합내의 activity들에 의한 flow logic을 명시하거나, flow logic은 집합내의 activity들 간의 변환에 의하여 생성되고 표현된 것이다.

context과 같은 첨가된 정보는 블록 activity의 어트리뷰트에 의하여 포함될 것이다. activity 집합은 block 이름과 블록 activity에 의하여 속하는 것과 함께 정의되며, 이것은 activity 집합의 정의를 재사용 하도록 만든 것이다. 매칭되는 BPML 범위의 다른 측면으로, activity set은 블록 activity 내에서 정의된다. 라우팅 로직을 사용한 이후에는 블록 activity를 완벽하게 명세할 수 있다.

### 3.6 Transaction and Exception handler

BPML은 트랜잭션에 대한 구조를 포함하고 다양한 타입의 예러나 예외를 핸들링 할 수 있다. 이것

은 ACID나 자동 트랜잭션이라 부른다. 구조는 다양한 어트리뷰트들, 롤의 범위와 여러 핸들링 로직을 포함한다. 그것들은 XPDL 어트리뷰트내의 맵이나 스페셜 라이브러리 기능을 사용할 수 있다. 이런 요구들은 세련된 문장을 요구하며, XPDL XML 스키마의 특별히 사용하는 확장에 대하여 조작할 수 있다. BPML에서는 모든 것은 잘 정의된 의미를 가진다.

| BPML                                     | XPDL   |
|--|--|
| <b>compensate</b><br>context내의 예외 처리로 연관 | 첨가된 어트리뷰트와 라이브러리 기능이 필요.   |
| <b>event handlers</b><br>time-out, fault | XPDL 1.0은 Deadlines의 명세를 포함한 예외 condition들과 deadline에 대한 예외 핸들러를 확장. |

<표 2> 트랜잭션과 예외 핸들러의 변환

### 3.7 XPDL Constructs with on BPML Analog

BPML은 프로퍼티의 인스턴스값을 기반으로 activity들의 관계자들의 명시나 정의의 표현의 방법은 제공하지 않는다. 그것은 리소스의 요구를 수행시키기 위한 activity를 명세한다. 그러나, 이것은 어떤 의미가 즉각적으로 제거되지 않을 것인가를 정의한다. BPML은 패키지에 대한 문법 정의에 애플리케이션의 개념을 가지고 있지 않다. 또한 activity의 구현도 마찬가지이다. BPML 프로세스와 activity를 수행하는 것과 애플리케이션을 가져오는 것 사이의 구별을 만들지 않았다. construct들 각각은 BPML action activity의 이용을 지원하게 될 것이다. WSDL Solicit-response를 사용하는 인스턴스에 대해서는 메시지는 표현을 수행하는 것을 포함할 것이

| BPML  | XPDL   |
|---|--|
| <b>package</b>  | <b>Package</b>   |
| <b>process</b> - 실제의 이슈는 검토. 최상위 레벨의 프로세스는 특별. 프로세스는 실제로 도착할 것임. 프로세스는 재사용 가능한 유닛.  | <b>WorkflowProcess</b> - input/output 파라미터들에 첨가된 instantiation type과 scope를 포함하는 다른 요구된 어트리뷰트를 포함. 프로세스는 activitySet을 포함                 |
| <b>Nested process</b><br>문맥의 범위에서 정의. (최상위 프로세스와 같이)  | 프로세스 정의를 포함하는 정의를 허용하는 엘리먼트와 같은 block activity를 요구.  |
| <b>import</b>   | <b>ExternalPackage</b>   |
| <b>activitySet</b> -context는 activity 집합과 연관. 특성은 context의 인스턴스 내에서 공유. 범위 이슈는 명확.  | <b>activityset(XPDL 1.0에 포함)</b> - 단지 프로세스보다는 activity들의 그룹핑에 대한 구조. 범위 이슈가 잔존.  |
| <b>complex activity</b> - 하나나 그이상의 activity 집합으로 이루어져 있다. activity 집합이 재사용 가능하지 않음.<br>All, choice, foreach, sequence, switch, until, while | <b>block activity</b> -name에 의해서 activity 집합을 참조 제안 : 모든 BPML complex activity들은 XPDL block activity들의 activity 집합을 참조하거나 기술된 라우팅 로직을 사용 |
| <b>activity</b><br>여기에는 많은 종류의 자동 activity들과 complex activity들이 있다.   | <b>activity</b><br>사용되는 activity 어트리뷰트를 표현하는 activity의 다른 종류   |
| 자동 activity와 복잡한 activity들의 분할된 테이블   |  |

<표 3> PDL Constructs with on BPML Analog의 변환

며 서비스로 보내게 될 것이다. 참여자를 할당하는 것과 함께 워크 리스트 핸들러와 response는 구현될 것이다. 이것은 애플리케이션의 지시를 포함하여 실행하게 될 것이다.

XPDL은 시뮬레이션이나 그이상의 영역내에서 유용하게 사용할 수 있는 특별한 어트리뷰트의 개수를 포함한다. 그런 것들은 BPML에서는 쉽게 첨가할 수 있다.

#### 4. 결론

현재, 기업간 통합을 위한 비즈니스 프로세스 표준들이 속출하고 있다. 특히 앞에서 언급한 BPML을 포함하여 웹 서비스 방식으로 애플리케이션 서비스를 이용하기 위한 방안으로 하는 여러 가지 표준들, 즉 웹 서비스 구성 언어들이 대형 시스템 벤더들을 중심으로 제안되고 있다. 웹서비스를 통한 업무 프로세스를 독립적으로 사용하고 있던 IBM의 WSFL, Microsoft의 XLANG등도 BEA Systems와 함께 BPEL4WS로 통합하면서 기존의 독자적 표준을 대체하고 있다. 이처럼 웹 서비스는 대형 벤더의 표준 기술 제공에 힘입어 워크플로우를 비롯한 프로세스 통합의 중요한 대상으로 자리잡고 있다. 뿐만 아니라 비즈니스 프로세스 통합의 대상이 되고 있는 ebXML과 RosettaNet과 같은 전자상거래 표준 프레임 워크도 독자적인 프로세스 모델과 정의 방식을 포함하고 있다.

본 논문에서는 이러한 워크 플로우를 이용한 프로세스 통합을 하기 위하여, BPML과 XPDL의 통합 방안을 모색하였다. XPDL과 BPML의 여러 가지 특징들을 파악하고 각각의 특성들을 상호 교환을 위하여 분석하였다. 이를 이용하면 프로세스 통합을 좀 더 쉽고 효율적으로 할 수 있다.

#### 참고문헌

- [1] BPML working draft, 2002.6.
- [2] BPML working draft, 2002.4.
- [3] Workflow Process Definition Language, XML Process Definition Language Document Number WPMC-TC-10525 Document Status, Draft0.0.4a, 2001
- [4] Workflow Process Definition Language, XML Process Definition Language Document Number WPMC-TC-10525 Document Status, XODL 1.0 beta, 2002
- [5] 김훈태, 전자상거래 표준화 로드맵, 전자상거래 표준화 통합 포럼, see <http://www.ecif.or.kr/admin/morgue/Dataroom>