웹서비스를 이용한 비즈니스 통합 플랫폼의 구현

Implementation of Business Integration Platform using Web Services

김민수(Minsoo Kim)*, 김훈태(Hoontae Kim)**†, 김동수(Dongsoo Kim)***

초 록

기업간 협업을 실현하기 위해서는 기업간 프로세스 상호연계(B2Bi)와 기업 애플리케이션 통합(EAI)이 함께 고려되어야 한다. 이미 많은 영역에서 이에 대한 연구와 개발이 이루어지고 있지만, 두 시스템이 각기 개별적인 플랫폼과 환경하에서 설계되고 운영됨으로써, 이 둘을 다시 통합해야 하는 문제가 발생하고 있다. 본 연구에서는 기업간 협업을 위해, B2Bi 환경의 변화에 유연하게 대응하면서도 B2Bi와 EAI 시스템을 단일의 통합 구조로 구현한 비즈니스 통합 플랫폼을 제공하고자 한다.

본 연구의 통합 플랫폼에서는 협업 프로세스가 비즈니스 통합의 주요 기술로 인식되고 있는 웹서비스(Web Services)를 이용하여 진행될 수 있도록 하였다. 특히, 다양한 비즈니스 통합 표준들이 수용될 수 있도록 이들 표준들을 비교 분석하여 기본적인 실행 요소들을 정의하였으며, 이를 처리할 수 있으면서도 향후 확장성을 가지는 프로세스 엔진을 개발하였다. 또한 이러한 비즈니스 통합 플랫폼의 검증을 위하여 웹 수주 시스템을 구현하였다. 본 연구의 결과는 기업간 협업을 위한 비즈니스 통합의 수준을 높이는데 기여할 수 있을 것이다.

ABSTRACT

Not only Enterprise Application Integration (EAI) within a company but also business-to-business integration (B2Bi) should be achieved for companies to collaborate seamlessly through the Internet. There have been great efforts in this area of EAI and B2Bi, resulting in lots of solutions in the market. However, EAI and B2Bi systems have been designed and operated in independent platforms and environments causing a serious problem of integrating EAI and B2Bi systems. In this paper we have proposed a business integration platform for connecting EAI and B2Bi systems in a uniform architecture and also coping with changes in business environments.

In the integration platform proposed in this work, collaboration processes between B2B companies can be performed using Web Services technology widely considered as B2B integration platform. Especially, we have defined basic business execution elements based on a thorough analysis of various business integration standards and developed an efficient and extensible business process engine for B2B collaboration process management. A Web-based order processing system is developed for validation of our integration platform. It is expected that the level of B2B collaboration can be enhanced using the integration platform and furthermore it will contribute to more collaborative B2B commerce.

키워드 : 기업간 협업, 웹서비스, 비즈니스 통합 플랫폼, B2Bi, EAI

B2B collaboration, Web Services, Business integration platform, B2Bi, EAI

* (주)큐빅씽크

** 대전대학교 산업시스템공학과 †교신저자

***가톨릭대학교 의료경영대학원

1. 서 론

오늘날 여러 산업 분야에서 기업간 협업(collaboration)이 강조되면서, 이에 대한 많은 연구가 진행되고 있다. 기업간 협업유형은 관점에 따라 여러 가지 분류가 가능한데, 적용된 정보통신기술의 수준에 따라 공급사슬형 기업, 확장형 기업, 가상 기업 등으로 분류될 수 있다. 여기에는, 장기적인 계약관계 위에서 기업간 정보 시스템의 긴밀한 연계를 형성하는 공급사슬형 기업에서부터 일시적인 만남과 해체를 반복하는 가상 기업에까지 여러 스펙트럼이 존재할 수 있다[4]. 따라서 정보통신기술의 측면에서 기업간의 다양한 협업 방식을 가능케 하기 위해서는 긴밀한 연계뿐만 아니라 유연한 연계 형태까지도 지원할 수 있어야 한다.

기업간 협업을 실현하기 위해서 기업간 프로세스 상호연계(B2Bi)와 기업 애플리케이션 통합(EAI)에 대한 연구와 개발이 이루어지고 있다. 그러나, B2Bi는 다양한 비즈니스 표준을 반영해야 하는데 반해, EAI는 솔루션 공급업체들의 고유방식에 따라 과거부터 개별적인 형태로 진행되어 왔다. 이것은 공용 프로세스(Public process)와 사설 프로세스(Private process)의 실행을 위해 각기 다른 시스템이 구축되어 운용되는 결과를 가져왔다. 점차 이 두 시스템에 대한 통합요구가 증가함에 따라, 기업들은 계속하여 변화해가는 B2Bi 프로세스를 지원하면서도 기존의 EAI 프로세스에 미치는 영향을 최소화할 수 있는 통합 방법을 필요로 하게 되었다. 이에 따라 비즈니스 표준이 도입될 때마다 새로운 B2Bi 시스템을 도입하고, 이를 다시 EAI 시스템과 개별적으로 통합하는 기존의 방식보다는 단일의 전사적 통합 프레임워크 상에서 B2Bi 프로세스의 확장을 지원할 수 있는 구조가 바람직하다고 하겠다.

본 연구에서는 이러한 구조를 비즈니스 통합 플랫폼으로 정의하였다. 특히 협업을 위한 다양한 연계 형태를 제공할 수 있도록, 데이터 및 트랜잭션 통합과 같은 긴밀한 연계 형태에서부터 웹서비스(Web Services)를 통한 서비스 지향의 유연한 연계 형태까지를 모두 지원하도록 비즈니스 통합 플랫폼을 구축하였다.

본 연구의 2장에서는 웹서비스 구성을 위한 표준들을 비교분석하고, 3장에서 표준들을 수용할 수 있는 실행요소로써 액티비티 콘트롤러(activity controller)와 서비스 구성요소를 정의하였다. 4장에서 비즈니스 통합 플랫폼의 프레임워크와 그 개발 결과를 제시하였다. 5장에서는 운영 시나리오를 제시하여 실행하였으며, 이를 통해 운용성을 검증하고, 기능적 요구사항이 만족되었는지를 평가하였다. 6장에서 결론을 제시하였다.

2. B2Bi를 위한 웹서비스 구성 표준

비즈니스 통합 플랫폼은 다양한 B2Bi 표준을 수용하면서도 확장되거나 새로이 추가되는 표준이 기업내부시스템에 영향을 미치지 않도록 해야 한다. 또한 다양한 협업유형이 가능하도록 웹서비스와 같은 유연한 연계 형

태까지를 지원해야 한다. 이를 위해서, B2Bi 표준들을 분석하고 B2Bi 표준의 액티비티(activity)들을 수용할 수 있는 구조를 정의하여야 한다. 이번 장에서는 B2Bi를 위한 웹서비스 구성 표준들을 비교·분석하였다.

기업들은 비즈니스 통합에서 개별 시스템들의 독립성을 위해 메시지 교환을 이용한 느슨한 통합 방식(loosely-coupled integration)을 선호한다. 이와 관련해 최근에 웹서비스가 각광 받고 있다. 웹서비스는 웹을 통해 접근할 수 있는 비즈니스 서비스나 애플리케이션, 또는 시스템 기능의 단위이다[6]. 웹서비스는 서비스 명세가 WSDL(Web Services Description Language)로 기술되어, UDDI(Universal Description, Discovery, and Integration) 레지스트리에 등록·검색될 수 있다. 사용자는 UDDI에 등록된 WSDL 정보를 바탕으로 SOAP(Simple Object Access Protocol) 메시지를 보내 해당 서비스에 접근할 수 있다. 웹서비스는 효과적인 비즈니스 수행을 위해서 WSCI(Web Services Choreography Interface)[1], BPML(Business Process Modeling Language)[2] 및 BPEL4WS(Business Process Execution Language for Web Services)[7]와 같은 여러 협업 표준들에서 활용되고 있다.

WSCI는 동적인 웹서비스 운용을 위해 제안되었다. 웹서비스는 WSDL을 사용하여 입출력 인자 및 호출 방식을 기술하지만, 정적인 애플리케이션 서비스를 제공한다. 이에 비해 WSCI는 복잡한 상호작용을 기술하기 위하여, 다수의 WSDL을 결합하여 동적인 서비스 명세를 가능하게 한다. 이외에도 임의의 웹서비스 참가자들이 기반 플랫폼과 프로그래밍 모델에 무관하게 상호운용 가능한 협업을 구성할 수 있도록 WS-CDL(Web Services Choreography Description Language)이 새로이 정의되고 있기도 하다[5].

BPML과 BPEL4WS는 분산된 웹서비스간의 비즈니스 논리를 표현할 수 있는 프로세스 정의 언어이다. 이 둘은 기업의 보편적인 비즈니스 프로세스를 포괄할 수 있는 추상화된 모델과 문법을 제시하며, 기업의 비즈니스 프로세스, 다자간 협업, 그리고 복잡한 웹서비스 구성을 정의하기 위해 사용된다. 특히, BPEL4WS는 흩어져 있는 웹서비스를 비즈니스 프로세스로 구성하기 위해, WS-Coordination, WS-Transaction과 함께 웹서비스 상호작용 모델을 확장하여 제공한다[3].

2.1 웹서비스와의 연계

웹서비스의 구성에 사용되는 모든 액티비티(activity)들은 단일 액티비티(simple activity)와 복합 액티비티(complex activity)로 크게 분류된다. 단일 액티비티는 더 이상 쪼개질 수 없는 원자적인 활동이며, 복합 액티비티는 다른 액티비티들로 구성된다. 비즈니스 구성언어들은 단일 액티비티를 통해 WSDL에서 정의된 'operation'들과 연계된다.

예를 들어, BPEL4WS의 'receive' 액티비티를 살펴보기로 하자. 'receive'는 메시지를 수신하는 액티비티로 'partnerLink', 'portType', 'operation' 및 'variable'이라는 네 가지 정보를 속성값으로 지정한다. 이때 'portType'과 'operation'은 WSDL 명세에서

제공하는 요소를 참조하고 있는 것으로, 이를 통해 웹서비스와 연동하게 된다. 위의 네 가지 정보는 비즈니스 프로세스에서 웹서비스를 호출할 때, 누가(portType) 어떤 역할(partnerLink)을 어떻게(operation) 그리고 어떤 정보(variable)를 주고받으며 하는지를 명시한다.

여러 B2Bi 표준에서 지원하는 단일 액티비티들이 〈표 1〉에 정리되어 있다.

BPML은 WSDL에 정의된 'portType'과 'operation'을 직접 참조하여 웹서비스와 통신한다. 이때 주고받는 데이터는 입력(input)과 출력(output)을 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 'portType'과 'operation'을 직접 참조하여 웹서비스와 연계된다. 그러나 BPML과 달리 입·출력을 사용하지 않는데, 이것은 WSCI가 BPML이나 BPEL4WS가 표방하는 것처럼 기업 프로세스의 실행을 지원할 목적은 아니기 때문이다. 웹서비스와의 연계 측면에서도 BPEL4WS는 'partnerLink', 'partnerLinkType' 및 'portType'으로 연결되는 흐름에서 다소 중복적으로 WSDL과 연관되는 요소를 보이기도 한다.

세 표준 모두 참여자들 간의 프로세스를 'message', 'portType', 'operation'과 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 그래프의 두 가지로 나눌 수 있다. 블록 구조의 프로세스는 프로그래밍 언어로 구현하여 작동하기 쉬운 반면, 방향성 그래프로 표현된 프로세스는 분석가나 설계자가 이해하고 다루기 쉬운 장점이 있다.

BPML과 WSCI는 블록 구조로 프로세스를 표현한다. 재귀적으로 구성될 수 있는 블록 구조는 프로세스의 선언, 정의, 수행에서 중요하게 사용된다. BPML과 WSCI는 이러한 블록간의 흐름통제를 위해 모두 7개의 복합 액티비티들을 정의하고 있다.

BPEL4WS 역시 블록 구조를 중심으로 하지만 방향성 그래프의 설계 방법을 적절히 가미하고 있다. 즉, 'flow' 액티비티를 통하여 병

〈표 1〉 B2Bi 표준들의 단일 액티비티

구 분	단일 액티비티
BPEL4WS	receive, reply, invoke, assign, throw, terminate, wait, empty, pick, compensate
BPML	action, assign, call, compensate, delay, empty, fault, raise, spawn, synch
WSCI	action, delay, empty, fault, call, spawn, join

렬적인 블록을 설계하면서, 내부 액티비티들 간의 의존 관계를 링크를 통해 방향성 그래프의 형태로 지정할 수 있게 하였다. 그러나 순환하는 것과 구조적 경계를 넘는 것을 제한하고 있다. BPEL4WS에서는 모두 5가지의 복합 액티비티가 있는데, 블록간에 특정한 문맥(context)을 공유하기 위한 'scope' 액티비티를 가지고 있는 것이 특기할 만하다.

〈표 2〉는 여러 B2Bi 표준들간의 복합 액티비티를 비교한 것이다.

2.3 비즈니스 프로세스의 실행 방법

BPML과 WSCI에서는 'call', 'compensate', 'spawn'과 같은 액티비티를 사용하거나 메시지를 수신함으로써 새로운 비즈니스 프로세스를 시작시킬 수 있다. 액티비티를 사용하는 경우는 주로 상위 프로세스에서 중첩되어 있는 하위 프로세스를 실행시키기 위해 이용되며, 정해진 입력 메시지의 수신을 통해 시작되는 경우는 주로 외부 기업의 요청에 의해서 일방향(one-way) 혹은 요청-응답(request-response)의 형태로 나타나게 된다.

이와 달리 BPEL4WS는 오로지 정해진 메시지의 송·수신을 통해서만 비즈니스 프로세스가 시작될 수 있다. 앞에서 기술한 단일 액티비티인 'receive'를 통해서 프로세스를 실행시킬 수 있으며, 'invoke' 액티비티의 경우에는 메시지를 송신하여 외부의 웹서비스를 호출하는데 사용될 수 있다.

2.4 예외 처리

비즈니스 프로세스는 종종 실행시간이 길고 비동기적인 메시지 통신을 사용하는 경우가 많다. 이러한 환경에서는 신뢰성 보장, 장기간에 걸친 잠금(lock)이나 격리(isolation)의 어려움과 같은 이유에 의해 트랜잭션의 ACID 속성이 제한된다. 그 결과 비즈니스 프로세스는 보상(compensation)이라는 개념을 에러 처리에 이용하고 있다. 보상이란 단념한 프로세스 내에서, 이미 수행한 트랜잭션의 영향을 되돌리는 일련의 액티비티이다.

BPEL4WS와 WSCI에서도 장기 수행 트랜잭션(LRTs; Long-Running Transactions)의 처리에서 오류 처리와 보상을 정의할 수 있는데, 'throw' 액티비티는 예외상황에서 오류를 발생시키며, 이것은 'compensate' 액티비티에

〈표 2〉 B2Bi 표준들의 복합 액티비티

구 분	BPML/WSCI	BPEL4WS
병렬구조	all	flow
직렬구조	sequence	sequence
분기구조	choice, switch	switch
반복구조	until, while, foreach	while
기 타		scope

의해 보상되도록 하였다.

BPML에서도 이와 유사하게 'exception'과 'compensation'이라는 특이한 프로세스를 사용하여 예외처리를 정의할 수 있는데, 이들도 다른 프로세스처럼 자신만의 실행문맥과 이벤트, 구성 액티비티들을 가지고 있으며, 예외나 보상이 필요할 때 실행될 수 있다.

2.5 트랜잭션 관리

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

BPEL4WS에서는 직접적으로 트랜잭션에 관해 언급하지 않으며, 벤더가 특정한 트랜잭션 처리 프로토콜을 의무적으로 지원하도록 요구하지 않는다. 대신에 보완적인 표준언어인 WS-Transaction에서 트랜잭션에 관한 자세한 사항을 다루고 있다[3].

3. 비즈니스 통합 플랫폼의 액티비티 콘트롤러와 서비스 구성 요소

2장에서 살펴본 것과 같이, B2Bi 표준마다 조금씩의 차이는 있지만 유사한 성격의 액티비티들을 제공하고 있다. 이러한 액티비티들은 새로운 표준에 의해 추가될 수도 있을 것이다. 따라서 기존 표준의 액티비티들을 만족시키면서도 향후 추가될 수 있는 액티비티가 지원될 수 있도록 하여, B2Bi 프로세스의 변화가 기업내부시스템에 파급되는 것을 차단할 수 있어야 한다. 본 연구의 비즈니스 통합 플랫폼에서는 이러한 요구사항을 만족시키기 위하여 다양한 액티비티 콘트롤러(activity controller)들과 시스템 서비스를 정의하여 제공한다.

액티비티 콘트롤러들은 앞서 살펴 보았던 단일 액티비티와 복합 액티비티들의 의미와 기능을 구현하기 위해, 비즈니스 통합 플랫폼에서 제공하는 구성 요소이다. 액티비티 콘트롤러들은 기존의 B2Bi 표준들에 의해 제안된 여러 액티비티들을 재구성하여 개발되었는데, 크게 시스템 제공 액티비티 콘트롤러와 사용자 확장 액티비티 콘트롤러로 구분할 수 있다. 특히 사용자 확장 액티비티 콘트롤러는 상태관리 서비스 및 런타임 객체저장소와 함께 새로운 액티비티를 정의할 수 있도록 하여, 기존 B2Bi 표준의 변경과 새로운 표준의 도입으로 인한 충격이 기업 후단의 어플리케이션 및 데이터 통합계층에 까지 파급되는 것을 막아준다.

3.1 시스템 제공 액티비티 콘트롤러

본 연구의 비즈니스 통합 플랫폼에서 제공하는 17개의 시스템 제공 액티비티 콘트롤러들은 의미상 1) 흐름제어를 위한 콘트롤러, 2) 조건 및 반복처리를 위한 콘트롤러, 3) 트랜

잭션 처리를 위한 콘트롤러, 4) 동시성 제어를 위한 콘트롤러, 5) 프로세스 중첩을 위한 콘트롤러, 6) 시점조절 및 예외처리를 위한 콘트롤러로 구분할 수 있다.

3.1.1 흐름제어를 위한 콘트롤러

여기에는 'Sequence', 'Escape', 'If' 및 'Switch'가 포함된다. 'Sequence'는 프로세스의 순차적인 흐름을 제어하는 것으로, 앞서 살펴본 B2Bi 표준들에서 직렬구조를 표현하는 'sequence' 복합 액티비티에 대응된다.

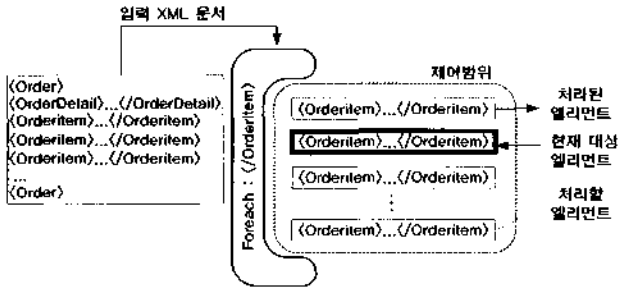
'Escape'는 전체 비즈니스 프로세스를 중단시키거나, 나중에 살펴보게 될 'While'이나 'Repeat' 등의 반복을 나타내는 콘트롤러로부터의 탈출을 위해 정의된다. 'Escape'가 전체 비즈니스 프로세스를 중단시키기 위해 사용될 때에는 조건에 따라 비즈니스 프로세스의 종료 상태를 성공과 실패로 결정할 수가 있다. 'Escape'는 BPEL4WS의 'throw', BPML의 'raise' 및 WSCI의 'fault'와 유사한 기능을 가지는데, 종료 상태를 지정할 수 있다는 장점을 가지고 있다.

'If'는 조건문의 평가에 의해 프로세스의 흐름을 둘 중에서 하나로 선택하는 반면, 'Switch'는 조건의 결과값에 따라 다양한 선택이 가능하다. 'If'와 'Switch'는 다른 B2Bi 표준에서 분기구조를 기술하는 'choice'와 'switch' 등의 복합 액티비티에 대응된다.

3.1.2 조건 및 반복처리를 위한 콘트롤러

여기에는 'While', 'Repeat' 및 'Foreach'가 포함된다. 'While'은 지정된 조건이 만족되는 동안 내포된 서비스나 콘트롤러들을 반복시키지만, 'Repeat'는 'While'과 달리 미리 정해진 변수나 상수의 값만큼을 반복한다. 이 두 콘트롤러는 다른 B2Bi 표준에서 반복구조를 나타내는 'until', 'while'과 같은 복합 액티비티에 대응된다.

'Foreach'는 입력 데이터에서 반복적으로 나타나는 동일위상의 자료구조를 처리하기 위한 것으로, 다른 B2Bi 표준의 'foreach' 복합 액티비티에 대응된다. 이 콘트롤러는 기업간 연계 프로세스에서 XML로 기술된 비즈니스 문서가 전달될 때, 반복되는 엘리먼트의 처리에 매우 유용하다. 특히 'Foreach'의 내부에서는 BPEL4WS의 'scope' 복합 액티비티와 같은 개념이 제공되기 때문에, 현재의



〈그림 1〉 'Foreach'를 통한 XML의 처리

반복 작업 대상이 되는 엘리먼트와 이전의 반복 작업 결과를 축적하고 있는 엘리먼트가 제어변수처럼 정의되어 콘트롤러 내부의 서비스들간에 공유될 수 있는 장점이 있다.

〈그림 1〉은 XML 문서에서 'Order' 엘리먼트 하위에 반복적으로 나타나는 'OrderItem' 엘리먼트들이 'Foreach' 콘트롤러에 의해 처리되는 예를 나타낸 것이다.

3.1.3 트랜잭션 처리를 위한 콘트롤러

일반적으로 트랜잭션은 단일의 리소스에 대해서 발생하는 지역 트랜잭션과 여러 분산된 리소스가 트랜잭션 관리자에 의해 조정되어 가며 진행되는 전역 트랜잭션(혹은 분산 트랜잭션)으로 구분된다. 지역 트랜잭션의 경우, 보통 대상 리소스가 되는 DBMS나 메시지 큐(Message Queue)에 대한 커넥션의 커밋(commit) 혹은 롤백(rollback)에 의해 쉽게 제어될 수 있기 때문에 처리가 비교적 단순하다. 그러나 전역 트랜잭션은 트랜잭션 관리자가 개개 리소스들의 커밋과 롤백을 2PC(2-Phase Commit)와 같은 프로토콜을 통해 모두 제어해야만 하는 복잡성을 가진다.

'StartTX'와 'EndTX'는 이러한 트랜잭션의 범위를 표현하기 위한 콘트롤러로, 내부에서 실행되는 트랜잭션 서비스들이 동일한 트랜잭션 관리자에 의해 제어되도록 해준다.

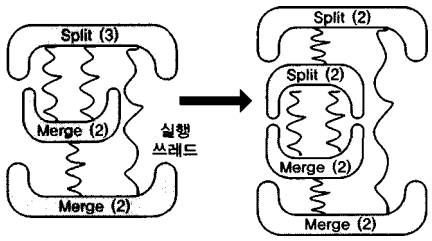
'CheckPoint'는 트랜잭션의 중간 결과가 커밋되어 유지될 수 있도록 해주는 콘트롤러로 'StartTX'와 'EndTX' 사이에서만 사용될 수 있다. 이것은 대용량의 트랜잭션을 처리하는 시스템에서 예기치 않은 시스템 실패가 발생할 경우 전체 트랜잭션을 다시 처음부터 수행하지 않고, 'CheckPoint' 이후부터 진행할 수 있도록 해준다. 이 세 콘트롤러는 앞서 살펴본 B2Bi 액티비티들 중에서 정확히 대응되는 것을 찾을 수 없다.

3.1.4 동시성 제어를 위한 콘트롤러

'Split'과 'Merge' 콘트롤러는 비즈니스 프로세스 내부의 여러 서비스들을 병렬적으로 실행시키는 것으로, 다른 B2Bi 표준에서 병렬 구조를 나타내는 'all', 'flow'와 같은 복합 액티비티에 대응된다.

본 논문의 비즈니스 통합 플랫폼에서는 'Split'과 'Merge'가 쌍으로 정의되고 동일한 개수의 분기를 갖도록 제약되어 있다. 그러나 'Split'과 'Merge'를 중첩시킴으로써 실제로는 다양한 분기 유형을 지원할 수 있기 때문에 비즈니스 모델의 표현 상에 문제를 발생시키지는 않는다.

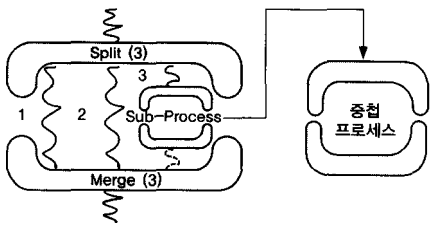
다음의 〈그림 2〉는 'Split'과 'Merge'를 중첩시켜 서로 다른 개수의 분기를 갖는 프로세스 구조를 같은 수의 분기를 가지는 구조로 정리하는 예를 나타낸 것이다.



〈그림 2〉 'Split'과 'Merge'의 중첩

3.1.5 프로세스 중첩을 위한 콘트롤러

'Sub-Process'는 한 프로세스 내부에서 새로운 프로세스를 동기적으로 호출하는 것으로, 내포된 프로세스가 종료되어 그 결과가 반환될 때까지 상위 프로세스의 순차적 처리를 중지시킨다. 그러나 'Sub-Process'가 처리되는 동안 상위 프로세스가 모두 중단되는 것은 아니다. 만일 'Split'에 의해 병렬적으로 진행되는 두 개 이상의 쓰레드가 상위 프로세스에 존재한다면, 'Sub-Process'를 사용한 쓰레드만 진행이 중지되며, 다른 쪽의 쓰레드는 계속 진행될 수 있다.



〈그림 3〉 'Sub-Process'로 인한 블로킹

〈그림 3〉에서는 3번 쓰레드만이 블록된 상태에서 나머지 1, 2번의 쓰레드는 계속 진행이 될 것이다. 물론 1과 2번 쓰레드도 'Merge' 콘트롤러에서 3번이 진행되어 올 때까지 블록되어 모두 동기화될 것이다.

'Chained-Process' 콘트롤러는 비동기적인 프로세스 호출을 위한 것으로, 상위 프로세스는 하위 프로세스의 성패 여부와 반환 결과에 대한 어떠한 정보도 필요로 하지 않으며, 계속 자체 프로세스를 진행하게 된다.

이 두 콘트롤러는 다른 B2Bi 표준에서 이야기 하는 'spawn', 'synch', 'join' 등의 단일 액티비티에 대응될 수 있다.

3.1.6 시점조절 및 예외처리 콘트롤러

비즈니스 통합 플랫폼 상에서 실행되는 모든 서비스에는 'Timeout'과 'Exception' 콘트롤러가 지정될 수 있다. 이것은 해당 서비스의 실행에 소요되는 시간이 프로세스의 성패에 영향을 줄 수 있도록 하며, 실패의 경우 보상이나 정리(cleanup)와 같은 예외처리 프로세스를 실행할 수 있게 한다.

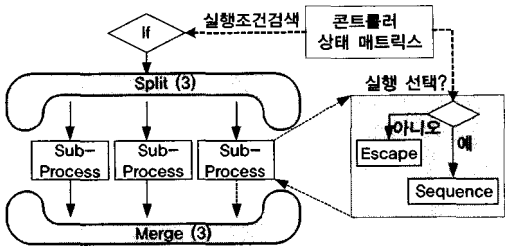
'Wait' 콘트롤러는 지정된 시간 동안만큼 해당 프로세스의 진행을 막는 것으로, 다른 프로세스의 처리 결과를 기다리거나 서로 다른 프로세스간의 시간차이를 조정하기 위해 사용된다.

이 콘트롤러들은 다른 B2Bi 표준에서의 'compensate', 'fault', 'wait' 액티비티들에 대응될 수 있을 것이다.

3.2 사용자 확장 액티비티 콘트롤러

비즈니스 통합 플랫폼 상에서 비즈니스 프로세스를 정의하여 실행하는 사용자는 시스템에서 기본적으로 제공하는 서비스와 액티비티 콘트롤러들을 조합하여 새로운 유형의 콘트롤러를 정의할 수 있는데, 이것을 사용자 확장 액티비티 콘트롤러라 한다. 이러한 콘트롤러의 확장성은 새로운 B2Bi 표준에 의해 추가되는 액티비티들을 쉽게 지원할 수 있도록 해주며, 새로운 어플리케이션 시스템이나 솔루션을 도입하지 않고도 기업간 프로세스 연계에 대응할 수 있게 해준다.

예를 들어, 'N-out-of-M', 'XOR'와 같은 콘트롤러들은 사용자 확장 콘트롤러로 정의할 수 있다. 〈그림 4〉는 '2-out-of-3' 콘트롤



〈그림 4〉 '2-out-of-3' 콘트롤러의 구성

러를 시스템 제공 액티비티 콘트롤러와 콘트롤러 상태 매트릭스를 통해 구성하는 예이다.

사용자 확장 액티비티 콘트롤러의 정의를 위해서는 구성 콘트롤러들의 개별 상태 정보를 종합적으로 관리하는 공통의 저장공간이 필요하다. 이를 위해서 비즈니스 통합 플랫폼에서는 상태 관리 서비스와 런타임 객체저장소와 같은 시스템 서비스를 제공한다.

3.3 시스템 서비스

상태 관리 서비스는 개별 콘트롤러와 비즈니스 프로세스 인스턴스들의 상태 정보를 제공하고 전이시킨다. 상태 정보는 상태 매트릭스에 의해 유지되는데, 현재 상태로부터 전이가 가능한 후속상태를 조건에 의해 선택할 수 있는 형태이다. 특히 다른 프로세스의 처리 결과를 이러한 전이 조건에 사용할 수 있다. 콘트롤러 상태 매트릭스와 변환 상태 매트릭스(Conversation State Matrix)가 각기 콘트롤러와 비즈니스 프로세스 인스턴스의 상태를 유지하기 위해서 사용되고 있다.

변환 상태 매트릭스는 기업 내·외부의 주체가 실행시킨 비즈니스 프로세스가 일련의 상호 작용을 진행시켜 나가기 위해, 현재의 상태 정보를 비즈니스 메시지 혹은 신호(Signal)의 송수신을 통해 전이시킬 수 있게 한다. 이러한 변환 상태 매트릭스는 B2Bi의 경우 일방향(One-Action) 혹은 양방향(Two-Action)과 같은 구성(Choreography)을 갖는 프로세스간의 대화(Conversation)에서 진행 상태를 전체적으로 유지하기 위해 사용된다.

상태 매트릭스가 콘트롤러나 비즈니스 인스턴스의 런타임 상태 정보를 제공한다면, 런타임 객체저장소는 여러 비즈니스 프로세스 간에 전달되거나 유지되어야 할 실제 데이터나 문서 객체를 보관한다. 즉, B2Bi 프로토콜을 통해 전달된 XML과 같은 비즈니스 문서가 파싱(Parse)되어 일단 메모리 객체로 구성되었다면, 이어서 실행되는 일련의 후속 프로세스들은 런타임 객체저장소를 통해 해당 객체를 참조하여 사용할 수 있다.

B2Bi와 EAI 플랫폼이 개별적으로 구성된 환경에서는 분리된 시스템간에 XML과 같은 비즈니스 문서를 전달하기 위해서 여러 번의 직렬화(Serialization)와 파싱을 반복해야 하겠지만, 본 논문에서 제시하는 비즈니스 통합 플랫폼에서는 런타임 객체저장소를 통해 쉽게 비즈니스 데이터가 전달될 수 있기 때문에 데이터의 불필요한 변환 작업이 줄어든다는 장점이 있다.

4. 비즈니스 통합 플랫폼의 구현

4.1 비즈니스 통합 플랫폼의 구조

본 연구로 개발된 비즈니스 통합 플랫폼의 전체 구조가 〈그림 5〉에 나타나 있다.

통신 프로토콜 및 어플리케이션 어댑터 들은 HTTP(S), FTP, SMTP 등의 대표적 프로토콜과 SAP/R3의 RFC와 같은 ERP 솔루션의 레거시(Legacy) 프로토콜을 통해 비즈니스 문서를 송수신함으로써 새로운 비즈니스 프로세스의 인스턴스를 생성하게 된다.

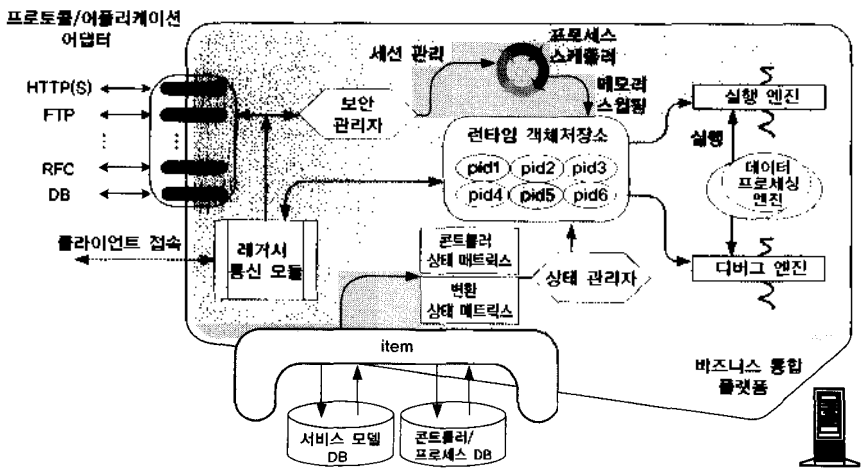
생성된 인스턴스는 자신이 사용하게 될 런타임 객체에 대한 참조(reference)를 가지고 비즈니스 모델의 상태 매트릭스에 따라 상태 로그(State Log)를 작성해 가면서 프로세스를 진행시키게 된다. 이에 대한 내용은 4.3절에서 다시 살펴보기로 한다.

실행 엔진(Execution Engine)은 프로세스 상태의 전이를 위한 조건문의 평가 및 데이터 변환(Conversion) 기능을 제공하며, 디버그 엔진은 프로세스 설계자가 프로세스를 배치(Deployment)하기 전에 검증을 위해 사용한다. 실행 엔진과 디버그 엔진의 처리 논리는 동일하지만, 분리되어 구현된 이유는 런타임 시에 실행 시스템이 디버그 시스템에 의해서 영향 받지 않기 위한 것이다.

리포지토리(Repository) 서비스 계층은 비즈니스 프로세스 모델과 액티비티 콘트롤러의 정의 및 상태 정보를 저장하고 있는 DB에 대한 접근을 조정·통제한다.

4.2 실행 엔진과 기본 처리

실행 엔진은 기본적으로 스택(Stack)에 기반하여 비즈니스 프로세스를 처리하게 되지만, 'Split'과 'Merge' 콘트롤러를 지원하기 위하여 트리 형태로 확장된다. 즉, 'Split'을 만나게 되면 병렬적인 분기의 개수만큼 자식 스



〈그림 5〉 비즈니스 통합 플랫폼의 구조

택이 생성되어 기존의 스택을 부모로 하는 트리를 형성하게 된다. 이때 추가된 각각의 자식 스택은 'Merge' 콘트롤러를 만나게 될 때까지 개별적인 쓰레드에 의해 실행된다. 실행엔진은 전체 스택 트리상에서 말단 노드(Node)만을 선택하여 실행시킴으로써 비즈니스 프로세스를 전개시켜 나가게 된다.

스택은 'Sub-Process' 콘트롤러에 의해서도 확장될 수 있는데, 두 개의 스택간에 부모자식관계가 형성되어서 부모 스택은 'Sub-Process'에 의해 실행되는 중첩 프로세스가 완료될 때까지 처리가 블록된다. 이러한 블로킹을 통해 중첩 프로세스의 동기적 호출이 가능해 진다. 반면 'Chained-Process'의 경우에는 스택에서 부모자식 관계가 형성되지만, 부모 스택의 쓰레드가 계속 진행되기 때문에 두 프로세스가 개별적으로 처리된다.

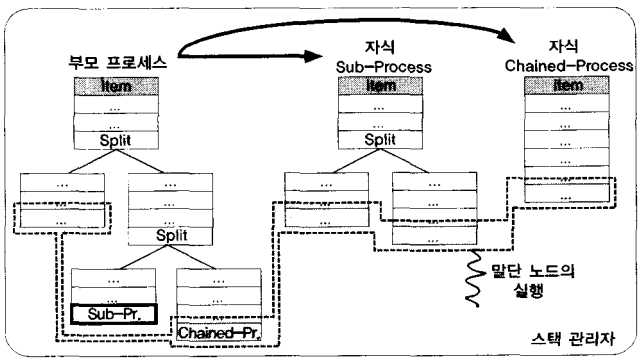
〈그림 6〉은 이러한 트리 형태의 스택 구조에서 프로세스 처리를 나타낸 것이다. 실행엔진이 전체 말단 노드를 수집할 때 부모 프로세스 스택 중에서 'Sub-Process' 콘트롤러가 포함되어 있는 스택을 실행에서 배제하고 있는 것을 볼 수 있다.

4.3 액티비티 콘트롤러의 처리

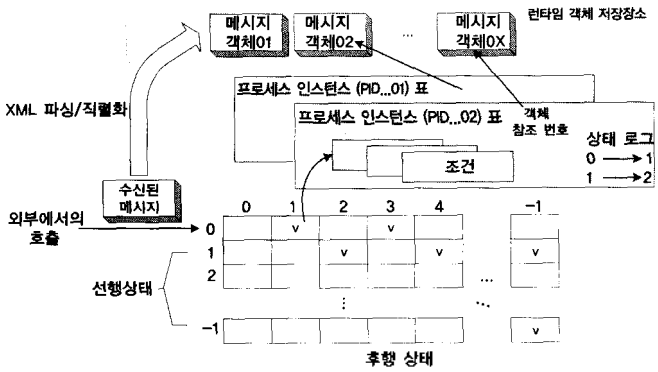
본 연구의 비즈니스 통합 플랫폼에 의해 정의된 콘트롤러들과 비즈니스 프로세스 모델들은 프로세스 인스턴스 테이블과 상태 매트릭스 및 런타임 객체저장소에 의해서 상태 관리와 입출력 데이터 연계를 진행하게 된다. 외부의 비즈니스 주체에 의해 발생된 비즈니스 메시지는 통합 플랫폼의 프로토콜 어댑터들에 의해 감지되어 비즈니스 문서와 함께 새로운 프로세스 인스턴스를 생성하게 된다.

비즈니스 문서는 리포지토리 서비스 계층을 통해 데이터베이스에 기록됨과 동시에 런타임 객체저장소에 전달된다. 데이터베이스에 기록하는 것은 해당 인스턴스가 급작스런 시스템의 장애 등으로 인해 유실되는 것을 막으며, 수신부인(Non-repudiation)을 방지하기 위한 것이다. 대부분의 B2Bi 표준들은 이러한 수신부인의 문제를 양사간의 협약의 형태로 강제하고 있다.

생성된 프로세스 인스턴스는 고유의 프로세스 번호(ID), 자신의 런타임 객체에 대한 참조 정보를 가지고 프로세스 인스턴스 테이



〈그림 6〉 트리 형태의 스택과 프로세스



〈그림 7〉 비즈니스 프로세스의 실행 과정

블에 저장된다. 이후 프로세스 인스턴스는 자신의 변환 상태 매트릭스와 콘트롤러 상태 매트릭스를 통해 실행 상태를 전이시켜 가면서 프로세스를 진행시키게 된다. 이러한 실행 이력은 상태 로그에 기록된다.

상태 매트릭스 상의 상태 전이를 위해서는 사전에 정의된 조건이 평가되어야 하는데, 새로운 프로세스 인스턴스의 호출 결과도 조건값으로 가능하다. 이러한 조건 역시 프로세스 인스턴스 테이블에서 유지된다.

〈그림 7〉은 이러한 비즈니스 프로세스의 실행 과정을 다양한 시스템 서비스와 함께 간략히 나타낸 것이다.

5. 적용 사례 및 평가

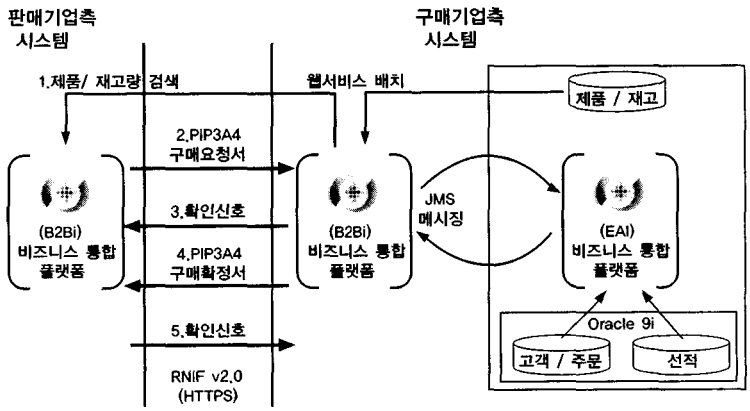
본 연구를 통해 개발된 비즈니스 통합 플랫폼을 활용하여 웹 수주 시스템의 시나리오를 구현하였다. 이번 장에서는 웹 수주 시스템의 성공적인 실행을 통해 비즈니스 통합 플랫폼의 유용성을 검증하였다. 특히 비즈니스 통합 플랫폼이 1) 협업의 다양한 연계 유형을 제공하고 2) 새로운 액티비티의 도입을 지원하며, 3) B2Bi와 EAI 기능을 통합적으로 지원할 수 있는지 라는 3가지 요구사항의 만족여부를 점검하여 전체적인 평가를 내리도록 하겠다.

5.1 웹 수주 시스템에의 적용 사례

기업간 협업에서 높은 비중을 차지하는 비즈니스 프로세스로 주문 처리 프로세스를 뽑을 수 있다. 본 연구에서는 주문 관리를 위해 로제타넷(RosettaNet) 표준에서 정의하고 있는 PIP3A4(Partner Interface Process 3A4)를 구현하여 웹 수주 시스템을 개발하였다[8]. 이때, 파트너 회사의 제품 목록과 재고량을 파악하는 프로세스를 웹서비스의 형태로 추가하여 협업 과정에 유연한 연계 형태도 포함되도록 하였다.

5.1.1 웹 수주 시스템의 구성

웹 수주 시스템은 PIP3A4에 정의된 대로 판매기업(Seller)과 구매기업(Buyer)의 역할을 맡는 두 B2Bi 시스템과 판매기업 측의 후단 통합을 맡는 EAI 시스템으로 구성되었다.



〈그림 8〉 웹 수주 시스템의 구성

이를 위해 모두 3개의 비즈니스 통합 플랫폼이 설치되었는데, 두 개의 플랫폼 인스턴스는 각 기업의 PIP3A4 공용 프로세스를 구현하기 위해 사용되었으며, 판매기업측에서 레거시 시스템과의 통합을 위해 한 개의 플랫폼 인스턴스가 추가로 설치되었다.

〈그림 8〉은 이러한 웹 수주 시스템의 구성을 프로세스와 함께 나타낸 것이다.

구매기업측의 통합 플랫폼에서는 판매기업측의 서버에 노출되어 있는 웹서비스를 통해 현재 가용한 제품의 목록과 수량을 파악한 후, 이를 토대로 로제타넷 표준의 구매요청서(POR: Purchase Order Request)를 생성하여 판매기업에 전달한다. 이때 사용된 PIP 3A4는 02.00 버전으로 HTTPS상에서 운영되는 RNIF 2.0 버전의 통신 메커니즘을 사용하였다[9]. 〈그림 8〉에 PIP3A4에서 실제로 발생하는 양방향 액티비티 다이어그램(Two-Action Activity Diagram)이 정리되어 있다.

판매기업측의 서버에서는 구매요청서에 대한 확인신호(Acknowledgement Signal)를 구매기업측에게 전달한 후, 구매확정서(POC: Purchase Order Confirmation)를 작성하기 위해, 새로운 프로세스 인스턴스를 생성하여 전달된 구매요청서와 함께 후단의 레거시 시스템에 전달하게 된다.

판매기업측 후단에서 레거시 시스템과의 통합을 받고 있는 서버는 비즈니스 프로세스 내에서 고객(Customer)에 대한 마스터 정보와 제품(Product Item) 마스터 정보가 각기 저장된 이종의 DBMS를 'StartTX' 및 'EndTX' 액티비티 콘트롤러를 통해 전역 트랜잭션의 형태로 묶어 사용하게 된다. 이때, 고객 마스터와 주문(Order) 정보는 Oracle 9i에서 관리되고, 재고(Inventory) 및 제품 마스터 정보는 MSSQL Server 2000에서 관리되고 있다. 이것은 EAI 시스템에서 일상적으로 발생하는 분산 트랜잭션을 구성하기 위해 의도적으로 설정된 환경이다.

주문의 생성과 재고량에 대한 갱신이 성공적으로 커밋할 경우, 판매기업측 후단의 EAI 서버는 B2Bi 서버에 그 처리결과를 전달하게

된다. 이 처리결과는 판매기업측의 B2Bi 서버를 통해 로제타넷 구매확정서로 만들어져 구매기업측의 서버에 전달된다.

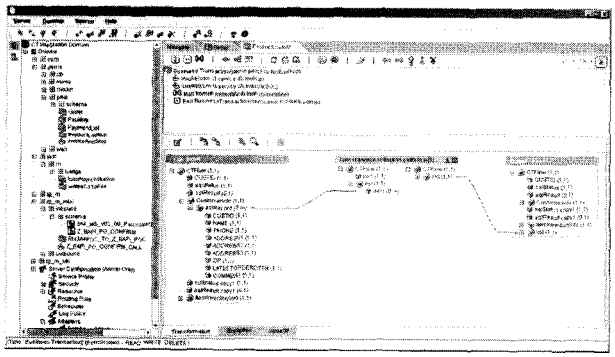
구매기업측의 서버는 구매확정서에 대한 확인신호를 보내줌으로써 웹 수주 시스템의 전체 협업 프로세스가 종료된다.

5.1.2 프로세스 모델과 실행 결과

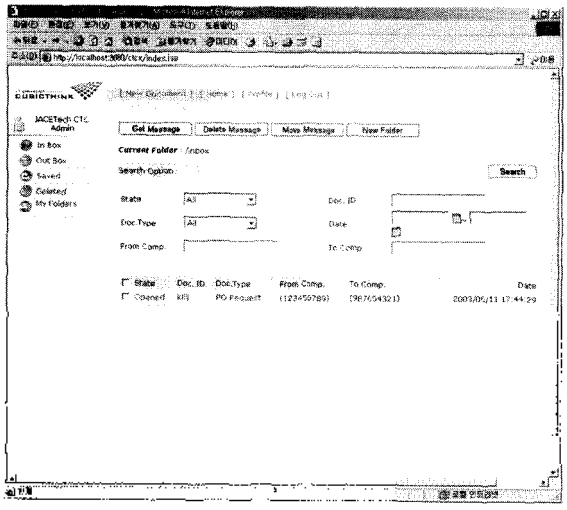
〈그림 9〉는 웹 수주 시스템의 시나리오에서 제품 마스터 정보와 재고 정보를 검색하기 위해 사용되는 'ProductLookup' 프로세스이다. 이 프로세스는 판매기업측의 B2Bi 플랫폼을 통해 구매기업측의 플랫폼에 웹서비스의 형태로 배치된 것이다.

〈그림 10〉은 구매기업측에서 생성하여 판매기업측에 보낸 로제타넷 구매요청서를 웹을 통해 조회해 볼 수 있도록 한 화면이다.

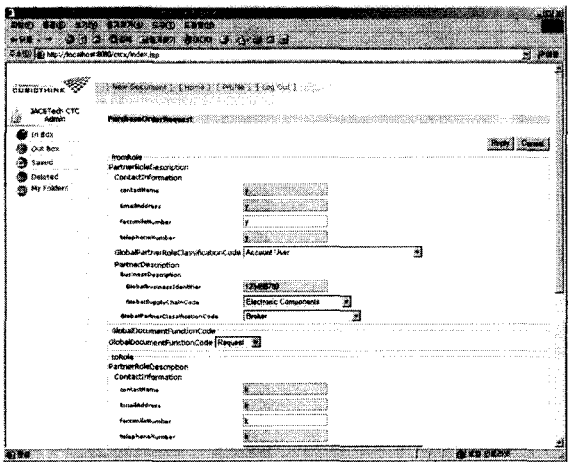
〈그림 11〉은 판매기업측의 플랫폼에서 구매기업측에 생성하여 보낸 로제타넷 구매확정서의 조회 화면이다.



〈그림 9〉 웹서비스로 배치되는 프로세스



〈그림 10〉 로제타넷 구매요청서의 화면



〈그림 11〉 로제타넷 구매확정서의 화면

5.2 비즈니스 통합 플랫폼에 대한 평가

웹 수주 시스템을 구축하고 운용함으로써, 본 연구에서 제기하였던 요구사항의 만족 여부를 살펴보고, 이를 통해 비즈니스 통합 플랫폼에 대한 평가를 내려보기로 하겠다.

5.2.1 다양한 연계 유형의 지원

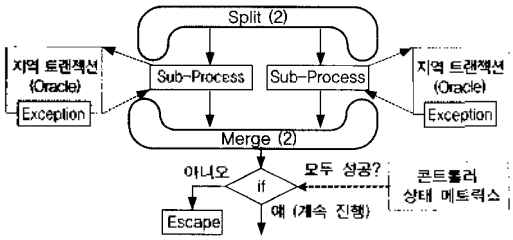
초기에 언급하였듯이 기업간 협업을 위해서는 데이터와 트랜잭션 중심의 긴밀한 통합 유형에서부터 메시징과 웹서비스와 같은 유연한 형태의 통합까지를 지원해야 한다. 비즈니스 통합 플랫폼은 웹 수주 시스템의 구축에에서와 같이 1) 이종 DBMS에 대한 데이터 및 트랜잭션 위주의 연계 형태, 2) 자바 메시지 서버(JMS: Java Message Server)를 통한 메시지 형태의 연계, 3) 웹서비스 호출과 같은 연계 형태, 4) 로제타넷 RNIF와 같은 B2Bi 구성(Choreography) 형태의 연계를 모두 지원한다. 이를 통해 비즈니스 통합 플랫폼이 다양한 연계 유형을 지원할 수 있음을 확인할 수 있다.

5.2.2. 새로운 액티비티 콘트롤러의 도입

웹 수주 시스템의 경우 시스템 제공 액티비티 콘트롤러만으로도 전체 프로세스를 실행시킬 수 있었지만, 사용자 확장 액티비티 콘트롤러가 사용되도록 새로이 '2-AND' 콘트롤러를 정의하였다. '2-AND' 콘트롤러는 앞서 시나리오에서 'StartTX'와 'EndTX'가 담당하였던 두 DBMS간의 전역 트랜잭션을 대체하였다.

〈그림 12〉와 같이 각 DBMS에 대한 지역 트랜잭션을 실행하는 두 개의 'Sub-Process'를 'Split'과 'Merge'를 통해 호출하는 형태로 구성되었다.

그 결과, '2-AND' 콘트롤러는 비즈니스 프로세스를 성공적으로 실행시켰으며, 이를 통



〈그림 12〉 '2-AND' 콘트롤러의 구성

해 사용자 확장 액티비티 콘트롤러가 후속 프로세스의 변경 없이 지원됨을 보였다.

5.2.3. B2Bi와 EAI 기능의 통합 지원

비즈니스 통합 플랫폼은 B2Bi와 EAI 기능을 단일 프레임워크 속에서 지원하도록 설계되었다. 이를 확인하기 위하여, 판매기업측에 B2Bi 기능과 EAI 기능을 각각 담당하도록 구성된 두 개의 시스템을 하나의 시스템으로 통합하였다. 즉, EAI 기능을 담당하는 비즈니스 통합 플랫폼의 프로세스 모델들을 모두 B2Bi 기능을 담당하는 비즈니스 통합 플랫폼에 옮긴 다음 웹 수주 시스템을 운용하여 아무런 프로세스의 변경 없이 시스템이 작동함을 확인하였다. 이것은 비즈니스 통합 플랫폼이 단일 구조 속에서 B2Bi적 요소와 EAI적 요소를 모두 지원하고 있음을 보인다고 하겠다.

이상의 결과를 통해, 본 연구의 비즈니스 통합 플랫폼이 앞서 언급하였던 3가지의 요구사항을 모두 만족시킴을 볼 수 있었다.

6. 결 론

기업간 협업의 효과적인 구현은 기존의 B2Bi 시스템과 EAI 시스템의 유기적인 재구성을 필요로 한다. B2Bi 및 EAI와 같이 통합을 위한 시스템들이 또다시 통합의 대상이 되는 악순환을 피하기 위해서는, 초기부터 B2Bi 시스템과 EAI 시스템의 기능적 요구를 단일화된 통합 플랫폼 상에서 정의하여 종합적으로 제공할 필요가 있다. 이 과정에서, 현재 기업 정보 시스템에서 변화를 빈발시키고 있는 B2Bi 프로세스가 기업 후단의 EAI 프로세스에 미칠 영향이 최소화되거나 차단될 수 있는 기능이 중요해진다.

본 논문에서 제시하는 액티비티 콘트롤러들은 변화의 가능성이 높은 B2Bi 표준으로부터 레거시 시스템을 보호할 수 있으며, 단일의 플랫폼을 통해 후단 시스템과의 반복되는 재통합의 요구를 피할 수 있게 한다.

본 논문에서 개발한 비즈니스 통합 플랫폼은 웹서비스를 포함해서, JMS 메시징, 전역 트랜잭션과 로제타넷 트랜잭션의 수행과 같은 다양한 서비스 및 연계 방식을 가능케 한다. 향후 이러한 비즈니스 통합 플랫폼의 개념과 관련한 지속적인 연구가 필요할 것으로 기대된다.

참 고 문 헌

[1] A. Arkin et.al., Web Services Choreography Interface 1.0, 〈http://www.w3.org/TR/wsci/〉 (8 Aug. 2002).

[2] Assaf Arkin, Business Process Modeling Language, 〈http://www.bpmi.org〉 (13 Nov. 2002).

[3] F. Cabrera et al., Web Services Transaction, 〈http://www.ibm.com/developerworks/library/ws-transpec/〉 (9 Aug. 2002).

[4] Jagdev, H. S., and Thoben, K.-D., Anatomy of enterprise collaborations, Production Planning and Control, 12 (5), 437-451, 2001.

[5] Kavantzas, Burdett, and Greg Ritzinger. Web Services Choreography Description Language Version 1.0,〈http://www.w3.org/〈TR/2004/WD-ws-cdl-10-20040427/〉 (27 Apr. 2004).

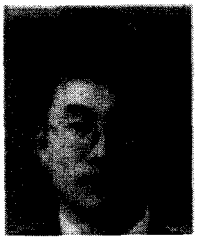
[6] R. Lay, J2EE Platform Web Services, Prentice Hall, 2003.

[7] T. Andrews et al., BPEL for Web Services 1.1, 〈http://www.ibm.com/developerworks/library/ws-bpel/〉 (5 May 2003).

[8] PIP3A4: Request Purchase Order. V02.00. 〈http://www.rosettanet.org/〉 (6 Dec. 2001).

[9] RosettaNet Implementation Framework: Core Spec. V02.00,〈http://www.rosettanet.org〉 (13 Jul. 2001).

저 자 소 개



김민수 (E-mail : minsky@cubicthink.com)
1994. 2 서울대학교 산업공학과 졸업(학사)
1996. 2 서울대학교 산업공학과(석사)
2002. 8 서울대학교 산업공학과(박사)
2002. 3 ~ 현재 ㈜큐빅씽크 기술연구소 개발팀장
관심 분야 워크플로우, e-비즈니스 프레임워크, B2Bi, EAI

김훈태 (E-mail : hoontae@daejin.ac.kr)
1988. 2 서울대학교 산업공학과(학사)
1990. 2 서울대학교 산업공학과(석사)
1997. 2 서울대학교 산업공학과(박사)
1997. 3 ~ 현재 대진대학교 산업시스템공학과 부교수
관심 분야 전자거래, 프로세스 분석 및 통합, 공급망관리

김동수 (E-mail : dskim@catholic.ac.kr)
1994. 2 서울대학교 산업공학과 졸업(학사)
1996. 2 서울대학교 산업공학과(석사)
2001. 2 서울대학교 산업공학과(박사)
2001. 1 ~ 2003. 3 한국전산원 전자거래연구부 e-biz 표준팀장
2003. 4 ~ 현재 가톨릭대학교 의료경영대학원 전임강사
관심 분야 BPM, 의료정보시스템, e-비즈니스 프레임워크