

# 암시적/명시적 협업 프로세스 언어의 비교분석

조 명 현<sup>†</sup> · 박 정 업<sup>\*\*</sup> · 설 주 영<sup>\*\*\*</sup> · 백 문 홍<sup>\*\*\*\*</sup> · 손 진 현<sup>\*\*\*\*\*</sup>

## 요 약

지금까지 비즈니스 프로세스 관리에 대한 표준 및 다양한 연구 활동들이 수행되어 왔다. 하지만 아직 범용적인 협업 프로세스 언어 표준이 규정되어 있지 않은 상태이기 때문에, 관련 연구 활동들이 체계적이지 못한 면이 있다. 본 연구는 대표적인 협업 프로세스 언어들(BPEL4WS, BPML, WSCI, WS-CDL, BPSS 등)의 비교 분석을 통하여 협업 프로세스 언어를 올바르게 선택하고 활용하는 가이드라인을 제시하고자 한다.

이를 위해, 본 논문에서는 먼저 협업 방식에 따라 암시적 협업과 명시적 협업을 정의하고 세 단계의 순차적인 방법론에 따라 협업 프로세스 언어들의 특징을 비교 분석한 결과를 제시한다. 첫번째 단계로, Bernauer가 제안한 협업 환경의 프레임워크와 협업 프로세스 모델링 절차에 따라 협업 프로세스 언어가 가져야 할 필수 요소를 추출한다. 두 번째 단계로, 추출된 필수 요소를 기반으로 협업 프로세스 언어들의 특징을 분석한다. 마지막으로 협업 프로세스 언어들의 특징을 반영하여 실제 비즈니스 프로세스를 구성하는 완전한 예제를 제시한다.

키워드 : 협업 비즈니스 프로세스, BPEL4WS, BPML, WSCI, WS-CDL, BPSS

## Comparison and Analysis of Implicit and Explicit Collaboration Process Languages

Myung Hyun Jo<sup>†</sup> · Jung Up Park<sup>\*\*</sup> · Joo Young Sul<sup>\*\*\*</sup> · Moon Hong Baeg<sup>\*\*\*\*</sup> · Jin Hyun Son<sup>\*\*\*\*\*</sup>

## ABSTRACT

Until now, a variety of the standard and research activities are progressed in the business process management. However, since the common standard of the collaboration process language has not been determined, the research activities could not be well-systemized. We would present the guide-line to select and use the collaboration process language straightly through comparing different collaboration process languages (BPEL4WS, BPML, WSCI, WS-CDL, BPSS, etc).

In this regard, we define the implicit and the explicit collaboration as the collaboration method in advance and present the result acquired according to compare and analyze the features of the collaboration process languages. First, the necessary elements the collaboration process languages have are extracted through the framework of the inter-organizational workflow proposed by Bernauer and the collaboration process modeling procedure(CPMP). Second, we analyze the properties of the collaboration process language based the essential elements. Finally, we show the complete example that the collaboration business process really reflects the characteristics of the collaboration business process languages

Key Words : Collaboration Business Process, BPEL4WS, BPML, WSCI, WS-CDL, BPSS

### 1. 서 론

비즈니스의 세계화와 정보기술의 발전은 경영 혁신에 있어서 새로운 패러다임을 제시하고 있다. 과거에는 조직 혁신에 대한 노력이 내부적으로 강화하는 차원에서 강조되었다면, 이

제는 파트너십을 기반으로 전체 가치사슬에 포함되어 있는 많은 이해 관계자들에 대한 혁신으로 광범위해지게 되었다. BPM은 확장된 기업의 프로세스를 자동화하고 통합되고 최적화하기 위해 설계된 새로운 세대의 기술을 일컫는 말이다.

BPM에 대한 정의는 솔루션을 제공하고 있는 벤더나 관련 연구자에 따라 조금씩 다르기는 하나 큰 범주는 벗어나지 않는다. Terry[1]은 BPM을 ‘비즈니스 프로세스를 설계, 통합, 감시, 그리고 분석을 수행할 수 있는 능력으로서 기업들이 수행하고 있는 조직 변화 혁신과정과 그다지 차이가 나지 않는 것’으로 정의하고 있다. Mazumdar & AbuSafiyah[2]는 ‘정보시스템에 있어 가장 핵심적인 애플리케이션이며, 이는 위

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성·지원 사업(ITTA-2005-C1090-0502-0016)의 연구결과로 수행되었음  
 † 준 회원 : 코난테크놀로지 근무  
 \*\* 준 회원 : 한양대학교 컴퓨터공학과 석사과정  
 \*\*\* 정 회원 : LG CNS 근무  
 \*\*\*\* 정 회원 : 한국생산기술연구원 수석연구원  
 \*\*\*\*\* 종신회원 : 한양대학교 컴퓨터공학과 조교수  
 논문접수 : 2005년 11월 17일, 심사완료 : 2006년 8월 23일

크플로와 유사하게 비즈니스 프로세스의 자동화를 구현하기 위한 비즈니스 논리'로 정의하고 있다. Leymann[3]등은 웹을 활용하여 비즈니스 프로세스에 대한 설계 및 통제, 통합 등을 수행하는 방법론에 대해 제시하고, BPM을 '기업이 가치를 창출하기 위해 필요한 프로세스 접근법'으로 정의 내리고 있다.

이렇게 정의된 BPM은 포괄적으로 '프로세스 관점에서의 기업경영'으로 정의될 수 있다. 그러나 여기서 가장 핵심이 되는 BPM의 특성은 '애플리케이션으로부터 비즈니스 프로세스의 분리'라고 할 수 있다. 즉, 과거 데이터베이스 관리시스템이 애플리케이션으로부터 데이터를 독립시켰던 과정과 마찬가지로, BPM은 오늘날 애플리케이션으로부터 비즈니스 프로세스를 분리시킴으로써, 비즈니스 프로세스의 정의, 구현, 개선, 관리, 분석, 자동화를 용이하게 이루어 내고자 하는 것이다.

현재 BPM관련 표준은 그 접근 방식과 지원 영역에 따라 다양한 표준들이 제시되고 있다. 일부 표준화 노력들은 이미 90년대부터 워크플로우 및 BPR 아래에서 지속되어 왔으며, 최근의 표준화들은 주로 BPM이라는 좀 더 확장된 패러다임 하에 새로이 개발되어지고 있다. 국제적인 표준 기구로는 WfMC, BPML, OMG, OASIS, W3C등을 예로 들 수 있으며, 대표적인 비즈니스 프로세스 표준안으로는 BPEL4WS[4], BPML[5], WSCI[6], WS-CDL[7], BPSS[8]등을 예로 들 수 있다. 본 논문에서는 Bernauer[13]의 워크플로우 프레임워크와 협업 프로세스의 모델링 절차에 따라 협업 프로세스를 구성하기 위한 필수 요소를 추출한다. 필수 요소를 기반으로 5개의 협업 프로세스 언어들의 특성 및 지원 영역들을 상세히 비교분석한다. 분석된 결과를 바탕으로 두 가지 협업 비즈니스 환경의 특성은 발견한다.

본 논문의 구조는 다음과 같다. 제 2장에서는 관련 연구로써 기존 비즈니스 프로세스 언어 비교논문들을 분석한다. 제 3장에서는 애플리케이션을 기반으로 협업 프로세스 언어의 지원 영역을 정의하고, 각 언어들을 비교한다. 제 4장에서는 협업 프로세스의 모델링에 관련된 프레임워크 및 절차를 기반으로 협업 프로세스에 필수 요소를 추출한다. 제 5장에서는 추출한 비즈니스 필수 요소를 기반으로 5개의 표준안의 엘리먼트 비교를 하고, 표준안들의 특징을 기술한다. 제 6장에서는 하나의 완결된 예제를 통하여 표준안들의 장·단점 및 차이점을 분석하고, 제 7장에서 결론을 맺는다.

## 2. 관련 연구

[9]는 [10]을 기반으로 15개의 비즈니스 프로세스 언어에 대한 표준안들을 비교하여, 13개의 개념들을 추출하였다. 이것은 각 표준안에서 제공하는 모든 기능들을 종합시킨 것이다. [9]에서 비교한 15개의 표준안 중 본 논문에서 비교한 5개의 표준안도 포함되었지만, 비교한 내용이 단순히 기능의 유무만을 판단했을 뿐이다. 즉, 각 표준안에서 어떻게 프로세스를 구성하는지 판단할 수 없다.

[11]은 Rausch-Schott[12]에 의해 제안된 워크플로우 모델

링의 프레임워크를 기반으로 조직 간의 워크플로우를 구성하는 7개의 표준안(WSDL, WSFL, XLANG, BPML, WSCL, ebXML, WPDLL)을 비교하여, 그들의 장단점을 기술하였다. [11]은 각 표준안을 중심으로 서술적인 방식으로 기술하였을 뿐, 각 표준안들 사이의 자세한 엘리먼트 일대일 비교가 없다. [13]은 eCo 프레임워크와 Rausch-Schott가 제안한 워크플로우의 개념적 프레임워크를 기반으로 WSDL 기반의 B2B 프로토콜과 ebXML 기반의 B2B 프로토콜을 비교하였다. Bernauer[13]은 [11]과 유사하게 워크플로우 모델링의 프레임워크를 기반으로 두 가지 접근 방식을 기술적인 엘리먼트를 첨부하여 비교하였다. 하지만 B2B 프로토콜 간의 언어에 대한 세부 비교보다는 메시지 교환 방식(eg. one way, request/response, etc)과 같은 WSDL을 포함한 기능적 구성 방식에 더 치중하여 기술하였다.

[14]에서는 대표적인 내부 프로세스인 XPDL, BPML, BPEL4WS를 기술적으로 상세히 분석하였다. 이것은 각 표준안에서 제공하는 모든 비즈니스 엔터티들의 사용 방법과 제한 사항을 기술하여 본 논문과 밀접한 관계를 갖고 있다. 하지만 최근 연구의 초점이 되는 협업 프로세스에 대해서는 언급하지 않고, 비교되는 구문들이 어떤 특징들을 갖는지 기술하지 않았다.

대부분의 기존 연구들은 비즈니스 프로세스 언어들의 공통되는 여러 기술적인 요소들을 추출하여 단순 비교만 하였다. 이러한 단순 비교는 실제 엘리먼트에 의한 일대일 비교가 아니기 때문에, 추상적이며 다양한 비즈니스 환경에 어떤 엘리먼트가 적합한 요소인지 구체적으로 판별하지 못한다. 또한, 협업 프로세스를 구성하기 위한 모델링 절차가 구성되지 않아 어떤 순서로 프로세스를 모델링해야 하는지 명확하지 않다. 협업 프로세스는 상호 운용성(Interoperability)을 고려하여 모델링해야 하기 때문에, 모델링 비용과 시간을 감소시키기 위해서는 모델링 절차에 따라 구성해야만 한다.

## 3. 협업 프로세스의 분류 및 동기

협업 프로세스는 목적에 따라 두 가지 방식으로 애플리케이션을 구성할 수 있다[15, 16]. 조직 내부의 비즈니스 프로세스를 기준으로 외부 파티를 끌어당기는 방식과 조직 간의 동의를 통해 협업 프로세스를 구성하는 방식이 있다. 전자의 경우, *Bottom-up engineering*이라 하는데[15], 외부 조직과 프로세스를 연계하기 위해서는 추상(Abstract) 프로세스를 UDDI나 ebXML Registry와 같은 글로벌 저장소에 등록해야 한다. 본 논문에서는 이와 같은 협업을 암시적 협업(*Implicit Collaboration*)이라 정의한다. 후자는 *Top-down engineering*이라 불리는데[15], 둘 이상의 조직이 먼저 공통의 동의(Agreement)를 구성하는 인터페이스를 구성하고, 각 파티는 인터페이스를 기반으로 독립된 내부 프로세스를 연결하거나 새롭게 생성한다. 본 논문에서는 이것을 명시적 협업(*Explicit Collaboration*)이라 정의한다.

**[정의 1] 암시적 협업(Implicit Collaboration)**

외부 조직과 비즈니스를 연계하기 위해 한 조직 내의 실행 프로세스를 기준으로 내부 프로세스의 추상 프로세스를 이용하는 협업

**[정의 2] 명시적 협업(Explicit Collaboration)**

외부 조직과 비즈니스를 연계하기 위해 여러 조직 간의 공통의 동의를 기반으로 정의한 비즈니스 프로세스를 이용하는 협업

비즈니스 프로세스는 조직 내외의 특성에 따라 실행 가능한 표준안과 인터페이스로 구별된다. BPELWS, BPML과 같은 비즈니스 프로세스 언어들은 조직 안에서 발생할 수 있는 여러 작업들을 실행 프로세스로 구성한다. 그래서 실행 프로세스가 외부 조직과 비즈니스를 연계하기 위해서는 내부 프로세스를 기준으로 추상(Abstract or Public) 프로세스를 구성해야 한다. 즉, 개별적인 시각(Individual View)으로 한 조직의 추상 프로세스를 정의한다. 이렇게 정의된 추상 프로세스는 글로벌 저장소에 등록되며, 외부 조직은 글로벌 저장소를 통해 자신이 원하는 프로세스를 찾을 수 있다. 이때, 추상 프로세스를 이용하는 조직은 하나 이상이 존재할 수 있기 때문에, 추상 프로세스를 제공하는 조직은 서버 형태로 프로세스를 제공해야 한다. 즉, 정의 1로 구성된 협업은 클라이언트-서버 형태로 구성되어야 한다. WSCI는 BPML과 WSDL의 기능을 상속받아 추상 프로세스로서의 역할을 한다.

반면에, 조직 간을 기준으로 프로세스를 구성하는 WS-CDL, BPSS와 같은 표준안들은 실행 가능하지 않는 인터페이스이다. 이것은 여러 조직들의 공통의 동의를 필요하기 때문에, 글로벌 시각(Common View)으로 인터페이스를 정의해야 한다. 이렇게 정의된 인터페이스는 각 조직의 내부 프로세스를 연결함으로써, 하나의 가상 프로세스의 형태를 이룬다. 즉, 각 조직이 독립적인 피어로서, 조직 간의 네트워크가 P2P 형태로 구성되어야 한다. <표 1>은 구성 애플리케이션에 따라 협업의 차이를 구성한 표이다.

위에서 기술한 것처럼, 5개의 협업 프로세스 언어는 애플리케이션의 사용 목적에 따라 크게 두 가지 환경으로 협업 프로세스를 만든다. 그래서 동일한 틀로써 5개의 표준안을 비교하는 것이 적합하지 않을 수 있다. 하지만, 현재 협업 비즈니스 환경의 급속한 성장에 따라 상호 운용성(Interoperability)을 위해 적절한 표준안을 선택할 필요가 있다. 표준안을 선택하는 첫 번째 방법으로는 조직 내 환경에 적합한 기존 표준안을 조직 간의 환경에서도 적합할 수 있도록 구성하는 것이다. 사실 BPELWS, BPML의 실행 프로세스는 조

<표 1> 협업 프로세스 분류

	명시적 협업	암시적 협업
프로세스 구성 형태	P2P	클라이언트-서버
관점	글로벌 시각 (Common View)	개별적 시각 (Individual View)
비즈니스 프로세스 언어	WS-CDL, BPSS	BPELWS(abstract), BPML(abstract), WSCI

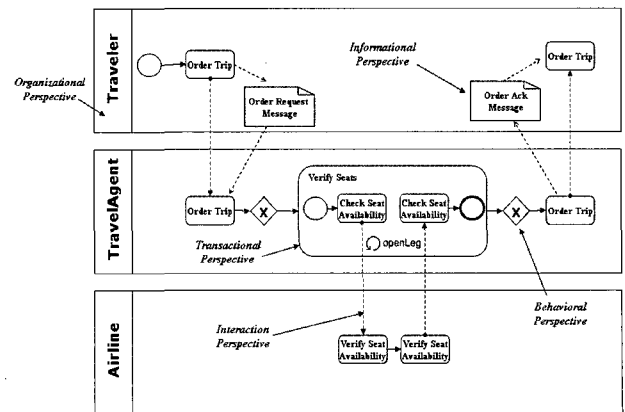
직 간의 환경도 고려하여 개발된 비즈니스 프로세스 언어이기 때문에, 이런 환경적 변화에 적응적으로 고려될 수 있다. 하지만 BPELWS(abstract), BPML(abstract)나 WSCI는 파티간의 동의 없이 한 조직의 입장에서 메시지 교환을 정의하기 때문에, 공유 데이터 타입을 정의하기 위해 추후에 비즈니스 프로세스의 변경을 요구할 수 있다. 이것은 내부 프로세스 또한 변경할 수 있어 심각한 비용 손실을 가져올 수 있다. 또, BPELWS나 BPML로 구성된 추상 프로세스와 웹 서비스를 외부 파티의 애플리케이션이 해석할 수 있어야 한다. 표준안을 선택하는 두 번째 방법으로는 조직 간 환경에 적합한 표준안을 기반으로 기존 내부 프로세스를 연결하는 방법이 있다. WS-CDL나 BPSS는 협업 파티간의 사전 동의를 기반으로 인터페이스를 정의하기 때문에, 내부 프로세스를 변경하지 않아도 된다. [16]처럼 협업 프로세스로부터 개별적인 내부 프로세스를 생성하기도 한다. 하지만 정확한 내부 프로세스의 연결을 통한 협업 프로세스의 완벽한 자동화를 이루기 위해서는 조직 간의 많은 기술 조율이 필요하다.

이런 구성의 환경적 차이는 [15, 16]과 같은 다양한 논문에서 기술되었다. 하지만 아직까지 협업 프로세스 표준안을 선택하기 위한 여러 표준안의 상세한 기술적 분석이 이루어지지 않았다. 그래서 본 논문은 비즈니스 프로세스 언어의 엘리트 일대일 내용 및 예제를 통한 분석을 통해, 표준 선택에 도움이 되고자 한다. 본 논문은 가정 1처럼 메시지 교환 프로토콜(ex. HTTP, SOAP)이나 구현에 관련된 하위 레벨은 고려하지 않는다.

**[가정 1] 5개의 비즈니스 프로세스 표준안 비교는 협업 비즈니스 환경만을 고려하며, 조직 내부의 환경인 구현 레벨은 고려하지 않는다. 오직 언어적 관점에서만 협업 프로세스 표준안들을 비교분석한다.**

**4. 협업 프로세스 모델링**

일반적으로 애플리케이션을 개발하기 위해서는 필요한 비용을 최소화하기 위해, 애플리케이션을 개발하기 위한 요소들을 여러 단계로 나누어 애플리케이션을 디자인한다. 유사하게,



(그림 1) 협업 프로세스의 5가지 관점

비즈니스 프로세스도 여러 요소로 구성되기 때문에, 디자인 시간과 비용을 최소화하기 위해 각 요소에 대한 순서를 정의할 필요가 있다. 본 장은 [13]에서 제안한 협업 프로세스의 5가지 요구 사항을 기반으로 협업 프로세스의 모델링 절차를 제안한다. 또, 각 단계에서 필요한 비즈니스 엔티티를 정의한다.

#### 4.1 협업 프로세스 프레임워크

Bernauer[13]은 [12]에서 제안한 워크플로우 모델링을 조직 간의 프로세스에 맞춰 약간 변경한 프레임워크를 제안한다. 본 논문도 가정 1에 의해 협업 비즈니스 환경만을 가정하고 있기 때문에, 5개의 표준안을 비교하기 위해 [13]의 프레임워크를 수용한다. 하지만 본 논문은 가정 1에 의해 내부 프로세스의 구현은 고려하지 않는다. 그래서 [13]에서 제안한 7가지 관점 중 함수적 관점(Functional Perspective), 연산적 관점(Operational Perspective)을 제외한 5가지 관점만을 비교한다. [13]에서 제안한 협업 프로세스의 5가지 관점은 행동적 관점(Behavioral Perspective), 정보적 관점(Informational Perspective), 대화적 관점(Interaction Perspective), 조직 관점(Organizational Perspective), 트랜잭션 관점(Transactional Perspective)이다. (그림 1)에서 위 5가지 관점을 나타내는데, 이것은 여행 예약 협업 프로세스의 일부를 BPMN[17]으로 디자인한 것이다. 여행 예약 협업 프로세스는 6장에서 자세히 다룬다.

행동적 관점은 언제(*when*) 액티비티가 수행해야 하는지와 관련된다. [13]에서는 컨트롤 흐름(Control Flow), 시간적 제약 요소(Timing Constraints), 예외 처리(Exception Handling)를 고려하였다. 하지만 본 논문에서는 시간적 제약 요소는 예외 처리에 포함될 수 있으며, 예외 처리는 트랜잭션이 내에서만 고려되기 때문에, 이 부분은 트랜잭션 관점으로 고려한다. 그래서 본 논문은 행동적 관점에서 컨트롤 흐름만을 상세히 분석한다.

정보적 관점은 데이터 구조나 액티비티들 사이의 데이터 흐름을 고려한다. [13]에서는 정보적 관점으로 데이터 타입(Data Type), 재사용가능한 데이터 타입(Reusable Data Types), 데이터 플로우(Data Flow)를 고려하였다. 본 논문에서는 표준안의 재사용 유무에 따라 데이터 타입도 재사용 가능하기 때문에, 재사용가능한 데이터 타입은 따로 비교하지 않는다. 대신 데이터 타입을 이용하는 프로퍼티를 고려한다.

대화적 관점은 서로 다른 파티에 속한 액티비티들의 상호 메시지 교환을 고려한다. (그림 1)에서처럼 BPMN에서는 이런 대화적 관점을 메시지 플로우(Message Flow)라는 엘리먼트로 정의하여 사용한다[17]. [13]에서는 메시지 교환에 필요한 상위 레벨의 구성 요소보다는 교환 프로토콜이나 메시지 교환 정책과 같은 하위 레벨의 구성 요소를 고려하였다. 그래서 본 논문에서는 가정 1에 의해 대화적 관점에서는 메시지 교환에 필요한 구성 요소인 참여자(Participants), 연산(Operation), 시맨틱 일관성(Semantic Consistency)을 고려한다. 대화적 관점에서 교환되는 메시지는 정보적 관점에서 고려된다.

조직 관점은 협업 프로세스에 누가(*who*) 참여하는 지를 고려한다. [13]에서는 조직 관점으로 역할(Roles), 프로파일(Profiles), 동의(Agreements), 동적 참여(Dynamic Participation)를 고려하였다. 협업 프로세스는 여러 조직들 사이에 구성되기 때문에, 조직의 구별은 상당히 중요하다. 그래서 본 논문은 WS-CDL의 참여자 정의[7]을 기반으로 협업 프로세스의 참여자(Participant) 구분을 좀 더 세분화 한다. 프로파일은 각 파티의 메타데이터를 정의하는 것인데, 5개의 표준안 모두 고려하지 않기 때문에 비교에 참여시키지 않는다.

마지막으로, [13]에서 트랜잭션 관점은 조직 내외 모두를 고려하는데, 가정 1에 의해 조직 간의 트랜잭션만을 고려한다. 트랜잭션의 상세한 구분은 트랜잭션의 단위 및 트랜잭션의 단위에 따른 처리 방법을 비교한다.

#### 4.2 협업 프로세스 모델링 절차(Collaboration Process Modeling Procedure)

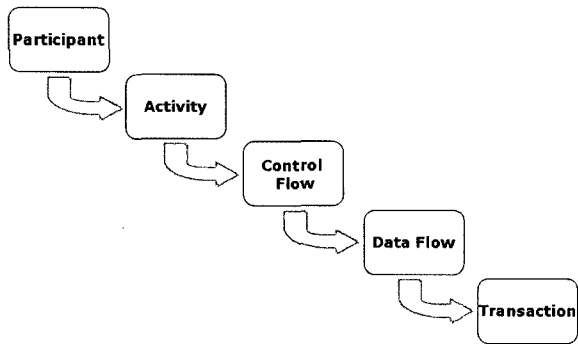
EDI(Electronic Data Interchange)와 같은 기존의 B2B 기술은 비즈니스 프로세스에 대한 고려가 없고 동일 시스템을 사용하는 회사 사이에서만 거래가 가능하다는 한계점을 가지고 있다. 이런 한계점을 극복하기 위해, 협업 프로세스에 대한 여러 표준안이 제안되고 있다. 하지만 아직까지 비즈니스 프로세스를 모델링하기 위한 절차는 제공하지 않고 있다. 그래서 본 논문에서는 (그림 2)와 같은 비즈니스 프로세스 모델링 절차를 제안한다.

먼저, 협업 프로세스는 조직 관점에서 프로세스의 부분을 책임져야 하는 참여자(Participant)를 구성해야 한다. 참여자는 조직 내에서는 조직 모델(Organization Model) [18]을 참조하여 부서(department)로 정의될 수 있으며, 조직 간에서는 프로세스에 참여하는 파티(Party)로 정의될 수 있다. 즉, 참여자는 프로세스를 구성하는 역할(Role)을 지칭한다. 예를 들어, 여행 예약에 관련된 협업 프로세스에서 참여자는 고객, 여행사, 항공사가 될 수 있다.

참여자가 정의되면, 대화적 관점에서 각 참여자가 메시지 교환에서 필요한 작업 또는 연산들을 정의할 필요가 있다. 이런 작업들은 일반적으로 비즈니스 프로세스에서 액티비티(Activity)라 불린다. 최근 비즈니스들이 대부분 웹서비스로 구성됨에 따라, 이런 행위들은 간단히 웹 서비스의 연산(Operation)으로 대체된다. 즉, 비즈니스 액티비티를 정의하기 위해서는 사전에 웹서비스를 정의하는 과정이 필요하다. 조직 내에서도 조직 내 애플리케이션을 통합하기 위해, 조직 간에서는 협업 프로세스를 구성하기 위해 웹 서비스를 이용한다. 본 논문에서 고려한 5개의 표준안 중 BPSS만을 제외한 모든 표준안이 웹 서비스를 이용한다.

비즈니스 프로세스는 아직까지 작업들의 관계성은 갖지 못한다. 각 행위들이 관계성을 갖기 위해서는 또는 각 작업들이 의미를 갖기 위해서는 행동적 관점에서 작업들의 컨트롤 흐름을 정의해야 한다. 이것은 모델링 표기법에서 화살표로 정의되며, 각 작업들을 연결하여 실행 순서를 결정한다.

앞서 정의한 컨트롤 흐름은 액티비티들을 단순히 연결하



(그림 2) 비즈니스 프로세스 모델링 절차 (CPMP)

여, 각 액티비티들의 실행 순서만을 정의하였다. 하지만 협업 프로세스는 정보적 관점에서 조직 간의 메시지를 송/수신하기 때문에, 송수신에 필요한 메시지를 정의할 필요가 있다. 즉, 액티비티들 사이에 데이터의 흐름을 정의할 필요가 있다. 예를 들어, 여행 예약의 협업 프로세스는 주문, 예약, 송장과 같은 메시지를 송수신할 필요가 있다. 송수신하는 메시지는 각 참여자의 지역 변수에 할당된다.

마지막 단계로 작업 수행에 관련된 트랜잭션 관점에서 트랜잭션을 정의할 필요가 있다. 트랜잭션은 특정 이벤트 및 예외를 감시하는 영역으로, 트랜잭션 내에서 발생하는 이벤트를 고려한다. 예를 들어, 여행 예약 시스템에서 고객의 주문 결제가 특정 시간 이상 지연된다면 여행사가 피해를 입을 수 있다. 이를 방지하기 위해 여행사는 예외 상황 발생을 위한 트랜잭션을 정의할 수 있다.

4.3 협업 프로세스 필수 요소

본 절은 4.1절과 4.2절을 기반으로 협업 프로세스를 구성하는 각 단계의 필수 구성 요소를 기술한다. 먼저, 참여자 정보는 조직 관점에서 다중 파티(Multi-Party)를 독립적으로 구성할 수 있어야 한다. 참여자 정의 단계는 세 부분으로 구성할 수 있는데, 참여자를 정의하는 단계, 참여자들의 동의(Agreement) 또는 관계성(Relationship)을 정의하는 단계, 동적으로 참여자들을 정의할 수 있는 단계로 나뉜다. 참여자는 구성 범위에 따라 파트너(Partner), 역할(Role), 행위(Behaviour)로 정의 된다. 즉, 파트너는 여러 역할을 포함할 수 있으며, 역할은 여러 행위들을 포함할 수 있다. 예를 들어, 중개 상인은 고객에게 공급자로서의 역할을 하지만, 제조업체에게는 소비자로서의 역할을 한다. 그래서 파트너는 대상 파트너에 따라 여러 역할을 가질 수 있다. 메시지 교환에 참여하는 두 조직의 역할은 관계성을 미리 정의할 수 있는데, 이것은 조직의 동의에 의해서만 가능하다. 예를 들어, 판매자와 중개자, 중개자와 구매자, 구매자와 배달 업체 등의 쌍이 파트너 사이의 관계로 정의된다. 동적 참여자 정의가 협업 프로세스의 필수 요소는 아니지만, 비즈니스의 복잡성이 증대됨에 따라 런-타임 시에 조직이 결정되는 기술을 요구한다.

협업 프로세스에서 작업(Activity)은 대화적 관점에서 항상 두 파티 간의 메시지 교환으로 구성된다. 메시지 교환의

정의 단계에는 3 개의 구성 요소가 필요하다. 첫째, 메시지 교환은 정확히 두 파티들 사이에서 일어나는 연산이기 때문에, 교환에 참여하는 파티를 정의할 필요가 있다. 이것은 위에서 설명한 참여자 관계(Relationship)를 이용하여 정의할 수 있다. 둘째, 메시지 교환은 파티의 웹서비스 함수(Operation)를 호출함으로써 구성하기 때문에, 파티에서 제공하는 연산(Operation)을 정의할 필요가 있다. 마지막으로, 메시지 교환에서 가장 중요한 정의 단계로 시맨틱 일관성(Semantic Consistency)을 유지하기 위한 정의가 필요하다. 이것은 하나의 프로세스에 하나 이상의 인스턴스가 생성되면서, 이런 인스턴스들 사이의 정보 교환을 정확하게 구성하기 위한 요소이다. 일반적으로 메시지 상호 작용(Correlation)이라고도 불리는데, 메시지들이 올바른 인스턴스에 전달되기 위한 방법이다. 이를 위해서는 메시지가 어느 프로세스 인스턴스에 속한 것인지를 판별할 수 있는 프로세스 인스턴스 아이디를 정의해야 한다.

컨트롤 흐름은 행동적 관점에서 각 작업들 사이의 실행 순서를 결정하기 위한 요소로써, 두 가지 방법으로 정의된다. 먼저 작업들을 단위별로 블록화시켜 그들의 흐름을 구성하는 블록-구조화(Block Structured) 방법이 있다. 이것은 Java나 C와 같은 일반 프로그래밍 방법과 동일한 것으로 BPSS만을 제외한 4개의 표준안이 모두 이 방법을 통해 작업들의 컨트롤 흐름을 결정한다. BPSS는 작업들을 모두 정의한 후, 그들의 흐름을 그래프 형식으로 연결하는 방향성 그래프(Directed Graph) 방법을 이용한다. 이것은 사람이 이해하기 쉬운 베이직과 같은 프로그래밍 방법과 동일하다. BPELWS나 BPMN의 일부 요소가 연결형 그래프 방식을 이용하기도 한다. 컨트롤 흐름은 일반적으로 시퀀스, AND, XOR, 컨디션, 루프의 5 개 기능을 제공한다.

데이터 흐름은 정보적 관점에서 각 작업들 사이에서 또는 메시지 교환에서 발생하는 정보를 고려한다. 일단 메시지 교환에 필요한 XSD와 같은 데이터 타입을 정의하는 단계가 필요하다. 협업 프로세스는 데이터를 공유해서 사용하기 때문에, 협업 프로세스 표준안에서 직접적으로 데이터 타입을 정의하는 것보다 XSD와 같이 일반적으로 알려져 있는 데이터 타입을 이용하는 것이 더 바람직하다. 데이터 흐름은 변수를 이용해 사용되는데, 메시지 교환과 액티비티들 사이의 프로퍼티 할당에서 사용된다.

트랜잭션은 트랜잭션 관점에서 조직 간에 발생할 수 있는 예외 상황을 고려한다. 일단 트랜잭션의 단위를 정의할 수 있는 요소가 필요하다. 트랜잭션 단위는 데이터베이스의 ACID 단위인 Short-running 트랜잭션과 비즈니스를 오랜 시간동안 처리하기 위한 Long-running 트랜잭션으로 나뉜다 [19, 23]. 5개의 표준안은 두 단위를 모두 제공하는데, 전자가 사용되는 영역은 메시지 교환이 일어나는 부분이다. 후자의 경우, 트랜잭션의 조작(Manipulation) 시점에 의해 크게 두 부분으로 나뉠 수 있는데, 트랜잭션 내에서 특정 이벤트 또는 예외 상황이 발생했을 때 바로 처리하는 예외 처리(Exception Handling) 및 이벤트 처리(Event Handling)와 트

<표 2> 협업 프로세스의 관점 및 모델링 절차에 따른 비즈니스 필수 요소

	비즈니스 필수 요소
조직 관점	1. 참여자 : 파트너 (Partner) > 역할 (Role) > 행위 (Behaviour) 2. 참여자 동의 (Agreement) 4. 동적 참여자 정의 (Dynamic Participation)
대화적 관점	1. 참여자 (Participants) 2. 연산 (Operation) 3. 시맨틱 일관성 (Semantic Consistency)
행동적 관점	구성 방법 : 블록-구조화 (Block-Structured) 연결형 그래프 (Directed Graph)
	기능 : 시퀀스 (Sequence), AND, XOR, 조건 (Condition), 루프 (Loop)
정보적 관점	1. 데이터 타입 (Data Type) 2. 변수 (Variable) 3. 데이터 플로우 (Data Flow)
트랜잭션 관점	단위 : Short-running 트랜잭션 Long-running 트랜잭션
	조작 : 예외 처리 (Exception Handling), 이벤트 처리 (Event Handling), 보상 (Compensation)

랜잭션 이후에 이벤트가 발생해서 트랜잭션 내부로 돌아가는 보상 처리 (Compensation)가 있다. 지금까지 설명한 비즈니스 필수 요소들은 <표 2>와 같다.

### 5. 비즈니스 필수 요소에 대한 비즈니스 프로세스 표준안 비교

각 표준안들은 초점을 맞추는 비즈니스 도메인에 따라 그 구성 요소가 조금씩 차이가 있다. 본 장에서는 4장에서 정의한 비즈니스 필수 요소들을 기반으로 5개의 비즈니스 프로세스 표준안들의 구분 및 특징들을 비교한다.

#### 5.1 조직 관점 (Organizational Perspective)

협업 프로세스에서 파티는 비즈니스에 참여하는 역할로서 정의된다. 3장에서 정의한 것처럼, 프로세스는 구성 방법에 따라 두 가지로 구성되기 때문에, 프로세스를 바라보는 시각이 독립적이거나 의존적일 수 있다. 즉, 암시적 협업 표준안들은 조직 간의 통신에서 표준안에 주관적으로 비즈니스를 기술할 수밖에 없다. 반면에, 명시적 협업 표준안들은 조직 간의 통신에서 독립적으로 존재할 수 있기 때문에, 표준안에서 각 파티들을 객관적으로 기술할 수 있다. <표 3>은 5개의 표준안이 파티를 바라보는 시각과 파티를 정의하는 구문을 정리한 표이다.

<표 3>에서처럼 명시적 협업 표준안은 파트너 자체를 기술하는 반면, 암시적 협업 표준안은 파트너라는 요소는 정의하지 않는다. 사실 암시적 협업 표준안은 한 조직만을 기술하기 때문에 파트너라는 요소를 정의할 필요가 없다. 하지만 이렇게 여러 조직을 명세서에 기술하지 않으면 협업 프로세스가 한 조직 중심적 (Central)으로 구성되거나 조직을 구별하기 위한 다른 기술들을 필요로 할 것이다. BPSS는 CPA의 4단계 총 [20]에서 최 상위층만을 정의하기 때문에, 메시지 교

환과 실제 연산에 관련된 런-타임 (Run-time) 행위들은 어떤 것도 정의하지 않는다. 그래서 <Business Transaction Activity>라는 추상적 액티비티를 이용해, 메시지 교환의 대리자 (Delegator) 역할을 한다. 비즈니스 복잡성의 증대로 비즈니스에 참여하는 조직들 또는 역할들이 다양하게 되었다. 이것은 비즈니스에 참여하는 참여자 또는 역할들을 모두 명세서에 기술할 수 없음을 의미한다. 즉, 런-타임 시에 동적으로 참여자를 할당할 수 있는 기술이 명세서에 정의될 필요가 있게 되었다. BPML과 WSCI는 참여자의 동적 참여를 위해 <locate>, <locator>를 이용하였으며, WS-CDL은 채널을 조직들 간의 메시지 교환에서 전달함으로써 동적 참여를 가능하게 하였다. BPELWS는 자체적으로 동적 참여를 정의하지 않지만, WS-Addressing [21]의 프로퍼티를 이용하면 가능하다. <표 3>의 '~' 표시는 기능이 일부 적용된다는 것을 의미한다.

<표 3> 조직 관점의 필수 요소에 대한 표준안들의 구분 비교

	BPML WS	BPML	WSCI	WS-CDL	BPSS	
파티를 바라보는 시각	주관적	주관적	주관적	객관적	객관적	
참여자	파트너	X	X	X	<participant Type>	<Partner >
	역할	<role>	<role>	<role>	<roleType>	<Role>
행위		<port Type>	<port Type>	<port Type>	<interface>	X
	동의	X	X	X	<relationshipType>	<Role>
동적 참여자 정의	~	<locate>+ <locator>	<locate>+ <locator>	<channelType>	X	

#### 5.2 대화적 관점 (Interaction Perspective)

최근 대부분의 애플리케이션에서 통신은 바이너리 형태의 데이터보다는 XML로 구성된 다큐먼트 형태의 데이터를 사용한다. 예를 들어, SOAP 메시지를 교환하는 AXIS와 같은 웹 서비스 애플리케이션이 대표적이다. 비즈니스 프로세스의 데이터 통신도 이런 상황을 기반으로 두 조직 간의 데이터를 교환하기 위해 메시지 교환을 기본적으로 사용한다. 메시지를 교환하기 위해서는 두 조직을 정의할 필요가 있는데, 이 때 두 조직의 역할을 정확히 정의할 필요가 있다. 즉, 연산을 요청하는 요청 파티 (ex. <fromRole>)와 요청에 응답하는 응답 파티 (ex. <toRole>)를 명시해야만 한다. 각 표준안의 파티 정의는 대칭/비대칭으로 구분되는데, 대부분의 표준안이 메시지 교환을 위해 정확히 두 파티를 명시한다. 하지만 WSCI와 BPML의 경우 자신의 파티만을 정의하는데, 이것을 보완하기 위해 BPML은 <locate>라는 구문으로 제공되는 연산의 정확한 위치를 기술하고 WSCI는 글로벌 모델을 이용한다.

연산의 경우, 대부분이 웹 서비스의 함수를 이용하는데, 각 표준안의 특징에 따라 함수 정의에 차이가 있다. BPELWS와 BPML은 웹 서비스를 제공하는 조직과 이것을 이용하는 조직이 대칭적으로 동일한 함수를 명시한다. 하지만 WSCI는 자신이 제공하는 웹 서비스 함수를 정의하고, 글로벌 모델에

<표 4> 대화적 관점의 필수 요소에 대한 표준안들의 구문 비교

		BPELWS	BPML	WSCI	WS-CDL	BPSS
참여자 (Participants)	구문	<partner Link>	<role>	<role>	<relationship Type>	<fromRole>,<toRole>
	특징	대칭	비대칭	비대칭	대칭	대칭
연산 (Operation)	구문	<operation>	<operation>	<operation>	<operation>	X
	특징	WSDL/대칭	WSDL/대칭	WSDL/대칭	WSDL/비대칭	X
시맨틱 일관성 (Semantic Consistency)	구문	<correlation>	<correlate>	<correlate>	<channel Variable>	X
	특징	메시지 상호 작용 (Message Correlation)	메시지 상호 작용 (Message Correlation)	메시지 상호 작용 (Message Correlation)	채널 (Channel)	X

서 대칭이 되는 독립 함수들을 조인한다. WS-CDL는 하나의 글로벌 모델을 정의하기 때문에, 웹 서비스를 제공하는 함수 하나만을 비대칭적으로 협업 프로세스에 명시하면 된다. <표 4>는 대화적 관점의 필수 요소에 대한 표준안들의 구문 비교표이다.

비즈니스 프로세스의 메시지 교환은 동기적이거나 비동기적일 수 있다. 일반적으로 협업 프로세스의 경우, 동기적 메시지 교환보다는 비동기적 메시지 교환이 빈번히 발생한다. 예를 들어, 여행 예약 프로세스에서 호텔 예약에 대한 예약 완료의 메시지는 단순히 알려주기만 하면 된다. 또, 정확히 한번의 메시지 교환에서 정보를 완료할 필요도 없다. 여행자가 여행사에 여행을 예약을 할 경우, 여행사는 항공사에 비행기를 예약해야 하기 때문에 여행사는 동기적으로 사용자에게 예약을 응답할 수 없다. 후자의 예제의 경우, 비동기 메시지들 사이에 의미가 연결될 필요가 있는데, 이러한 것을 시맨틱 일관성이라 한다. 시맨틱 일관성은 크게 두 가지 방법으로 구성될 수 있다. 하나는 메시지 속에 메시지를 연결할 수 있는 매개체를 삽입하여 메시지들 사이의 의미적 일관성을 유지한다. 이것을 위해 암시적 협업 표준안은 <property>를 전달하는 메시지에 포함시킨다. 반면에 WS-CDL은 채널의 <identity>을 이용해 메시지가 통과하는 채널을 구별하는 방법을 사용한다. 이것은 메시지를 비동기적으로 메시지를 전달하더라도, 메시지에 채널의 아이디를 포함시킴으로써 응답할 때 돌아와야 하는 채널을 알 수 있다.

5.3 행동적 관점 (Behavioral Perspective)

<표 5>는 행동적 관점의 필수 요소인 컨트롤 흐름에 대해 표준안들을 비교한 표이다. 컨트롤 흐름은 런-타임 시 각 태스크들의 실행 순서를 결정짓는 요소이다. 4.3절에서 기술한 것처럼, 컨트롤 흐름을 결정하는 두 가지 구조는 블록-구조화와 연결형 그래프이다. BPML의 <raise>, <synch>와 BPELWS의 <link>는 연결형 그래프 특징을 갖는 요소가 있지만 이들은 실행 프로세스에서만 작동하는 요소이기 때문에, 본 논문에서 고려하지 않는다. 비즈니스 프로세스를 구성하는 5 가지 컨트롤 흐름으로는 시퀀스, AND, XOR, 루프,

컨디션이 있다. 시퀀스는 작업들을 순차적으로 수행하기 위한 요소이다. AND는 구문에 포함된 작업들을 동시에 수행시키기 위한 요소이다. 시퀀스와 AND는 가장 기본적인 컨트롤 흐름이기 때문에, 모든 표준안이 제공한다. XOR는 조건에 의한 데이터 기반 XOR과 이벤트 발생에 의한 이벤트 기반 XOR로 구별될 수 있다. 암시적 협업의 표준안들은 두 기능을 구별되게 모두 제공하는 반면, WS-CDL의 <choice>를 이용해 복합적으로 사용할 수 있다. BPSS에는 이벤트라는 개념이 표준안에 정착되지 않았기 때문에, 이벤트에 의한 XOR은 정의되지 않았다. 최근의 비즈니스 프로세스는 그 생명 주기가 상당히 길어짐에 따라(Long-Transaction)[19, 23], 데이터 기반 XOR보다는 이벤트 기반 XOR가 더 중요하게 다루어지고 있다. 루프는 작업의 반복적 수행을 가능하게 하는데, 블록-구조화 구조의 표준안은 프로그래밍 언어처럼 쉽게 정의할 수 있다. BPML, BPELWS, WSCI는 <while>과 <condition>이라는 엘리먼트 혹은 애트리뷰트를 통해 루프를 정의한다. 반면에 BPSS는 연결형 그래프 구조로 이러한 기능을 제공하지 못한다. 연결형 그래프가 이런 기능을 제공하기 위해서는 XPDL의 <ActivitySet>과 같은 블록 단위가 명세서에 따로 정의되어 있어야 한다[22]. 컨디션은 작업의 흐름을 결정짓는 역할을 하는데, 연결형 그래프 구조를 갖는 프로세스 언어에 꼭 필요한 기능이다. WS-CDL은 <guard>라는 컨디션 요소를 두어, <repeat>나 <workunit>과 같은 요소와 복합적으로 사용하게 함으로써, 비즈니스 분석가에게 협업 프로세스 디자인의 다양성을 확대시켜 주었다. 하지만 익숙하지 않는 사용자는 협업 프로세스의 데드락(Deadlock)과 같은 문제를 야기 시킬 수 있다.

<표 5> 행동적 관점의 필수 요소에 대한 표준안들의 구문 비교

		BPELWS	BPML	WSCI	WS-CDL	BPSS
구성 방법		블록-구조화	블록-구조화	블록-구조화	블록-구조화	연결형 그래프
시퀀스 (Sequence)		<sequence>	<sequence>	<sequence>	<sequence>	<Transition>
AND		<flow>	<all>	<all>	<parallel>	<Fork>, <Join>에서 type="AND"
XOR	데이터 기반	<switch>	<switch>	<switch>		<Fork>, <Join>에서 type="XOR"
	이벤트 기반	<pick>	<choice>	<choice>		
	모두				<choice>	
루프 (Loop)		<while>	<while>, <foreach>, <until>	<while>, <foreach>, <until>	<repeat>	X
컨디션 (Condition)		X	X	X	<guard>	<ConditionExpression>

5.4 정보적 관점 (Informational Perspective)

정보적 관점은 4.3절에서 기술한 것처럼 3가지로 구성된

<표 6> 정보적 관점의 필수 요소에 대한 표준인들의 구문 비교

		BPEL4WS	BPML	WSCI	WS-CDL	BPSS
데이터 타입		~	~	~	<informationType>	~
변수		<variable>	<property>	<property>	<variable>	X
데이터 플로우	변수 할당	<assign>	<assign>	X	<assign>	X
	메시지 교환	<variable>	<property>	X	<variable>	<businessDocument>

다. 첫째, 런-타임 시 구성될 수 있는 데이터들의 구체적인 포맷으로 데이터 타입을 정의해야 한다. 대부분의 협업 프로세스 표준안은 XSD를 이용하여 변수나 프로퍼티를 정의한다. 반면에, WS-CDL은 명세서에 새로운 타입을 기술할 수 있는데, 이것은 외부 명세서로의 확장 의 용이하게 해준다. BPSS는 메시지 교환에서 교환되는 메시지를 <businessDocument>으로 정의하는데, 이것은 ebXML의 다른 표준안을 이용하여 정의되기 때문에 본 논문의 영역 밖이다. 데이터의 흐름(Data Flow)은 액티비티들 사이 또는 외부 액티비티와의 메시지 교환에서 사용된다. 명시적 협업 표준안들은 액티비티들 사이보다는 메시지 교환에서 데이터 흐름을 더 중요하게 고려한다. 그래서 WS-CDL이나 BPSS는 메시지 교환에서 발생할 수 있는 문제에 대한 책임들도 정의한다. 예를 들어, WS-CDL은 메시지 교환 시, <record>를 두어서 교환되는 메시지를 변수에 기록하기도 한다. <표 6>은 정보적 관점의 필수 요소에 대한 표준인들의 구문 비교표이다.

5.5 트랜잭션 관점 (Transaction Perspective)

DBMS는 데이터를 관리하는 시스템을 지칭한다. 유사하게 BPMS는 비즈니스 프로세스를 관리하는 시스템을 일컫는다. DBMS에서 가장 중요한 관리 대상 중의 하나는 다양한 사용자의 트랜잭션을 최대한 효과적으로 관리하는 것이다. 마찬가지로 BPMS도 다양한 조직 또는 일반 사용자에게 사용되는 비즈니스 프로세스의 트랜잭션을 효과적으로 관리해

<표 7> 트랜잭션 관점의 필수 요소에 대한 표준인들의 구문 비교

		BPEL4WS	BPML	WSCI	WS-CDL	BPSS
트랜잭션 단위	Long-running	<process> <scope>	<process>	<process> <context>	<choreography>	<BinaryCollaboration>
예외 처리	생성자	X	X	X	<causeException>	X
	핸들러	<faultHandler>	<faults>	<exception>	<exceptionBlock>	<Fail>
보상 처리	생성자	<compensate>	<compensate>	<compensate>	<finalize>	X
	핸들러	<compensationHandler>	<compensation>	<transaction>	<finalizeBlock>	X
이벤트 처리	생성자	<onAlarm>	<delay>	<onTimeout>	~	~
	핸들러	<eventHandler>	<exception>	<exception>	~	~

야 한다. 이렇게 DBMS와 BPMS가 많은 공통점을 갖지만, BPMS의 트랜잭션이 DBMS의 트랜잭션보다 오랫동안 지속되어야 한다는 차이점을 갖고 있다. 만일 오랫동안 지속되는 트랜잭션을 기존 DBMS처럼 계속해서 락킹(locking)하여 사용하면, 시스템은 상당히 비효율적일 것이다[19]. 그래서 BPMS는 트랜잭션의 단위를 작게 분할하여, 보상 처리나 예외 처리와 같은 기능을 추가시켰다. <표 7>은 트랜잭션 관점의 필수 요소에 대한 표준인들의 구문 비교표이다.

4.3절에서 기술한 것처럼 트랜잭션의 단위는 두 가지로 구분되는데, Short-running 트랜잭션에서 사용되는 엘리먼트로는 메시지 교환과 같은 심플 액티비티들을 예로 들 수 있다[5,6]. 일반적으로 Short-running 트랜잭션은 기존 DBMS의 기술로 충분히 포함할 수 있기 때문에, 본 논문에서 가정 1에 의해 Short-running 트랜잭션은 고려하지 않는다. 또, 최근의 비즈니스가 개인을 대상으로 하기보다는 조직 간의 협업을 요구하기 때문에, Short-running 트랜잭션에 관련된 엘리먼트들은 고려할 필요가 없다. Long-running 트랜잭션의 단위는 <표 7>과 같이 프로세스 단위이다. WS-CDL은 <choreography>, BPSS는 <BinaryCollaboration>이라 부른다.

협업 프로세스는 모든 것이 시스템에 의해 자동적으로 관리되지만, 시스템의 치명적 오류, 사용자의 취소, 프로세스 타임 지연 등과 같은 이벤트에 의해서 예외 상황이 발생할 수 있다. 이런 예외 상황은 조직 내/외의 프로세스의 오동작을 초래하여 심각한 비용 손실을 일으킬 수 있다. 모든 예외 상황을 막는 것은 불가능 하지만, 적어도 발생할 만한 예외 상황은 비즈니스 프로세스의 모델링 과정 중에 기술되어야만 한다. 대부분의 표준안은 예외를 자체적으로 정의하며, 예외 상황은 크게 이를 감지하는 영역 정의 부분과 핸들링 정의 부분으로 나뉜다. 일반적으로 프로세스 단위로 감지되기 때문에 자체적으로 따로 정의하지 않지만, WS-CDL은 메시지 교환 변수에 <causeException>을 두어 메시지 교환에서 발생할 수 있는 예외 상황만 처리한다. <표 7>에서와 같이 암시적 협업 표준안은 프로세스 자체의 예외를 처리하지만, 명시적 협업 표준안은 메시지 교환 영역에서의 예외도 처리한다. BPSS는 <Fail>이라는 구문을 통해 예외 상황을 잡아주는 하지만, 처리는 하지 못한다. 이것은 아마 CPA의 내부 협정에 의해 정의될 것이다[20].

트랜잭션의 보상 처리는 앞서 기술한 예에서와 같이 프로세스 간의 관계를 갖기 때문에, 협업 프로세스에서 상당히 중요한 역할을 차지한다. 그래서 <표 7>과 같이 대부분의 표준안들이 보상 처리를 고려한다. 예외 처리가 프로세스의 지속적인 운용을 위해 문제를 해결하려는 "Redo" 기능이라면, 보상 처리는 프로세스의 완벽한 초기화를 구성하려는 "Undo" 기능이다.

이벤트 처리는 시간에 대한 예외 상황만 기술하였는데, 이외에도 메시지 이벤트에 대한 이벤트도 처리할 수 있다. <표 7>에 알 수 있듯이, 암시적 협업 표준안은 모두 이벤트 처리를 정의하는 반면에, 명시적 협업 표준안은 일부만 정의한다. 예를 들어, 부록 A의 WS-CDL의 메시지 교환에서는 <timeout>이

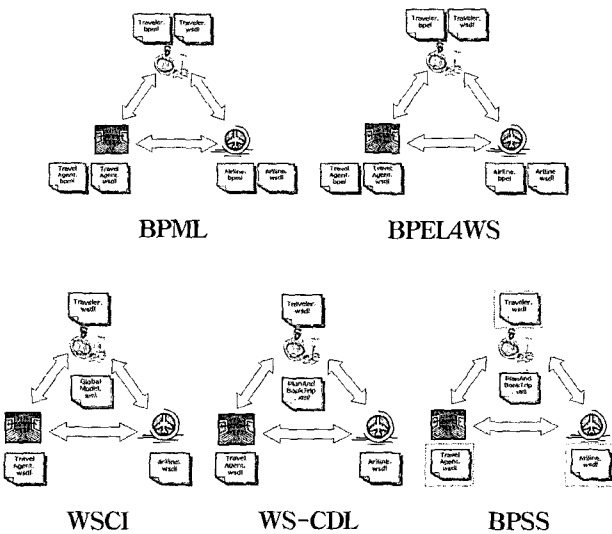


라는 엘리먼트는 메시지 교환에서 발생할 수 있는 시간적 예외를 처리할 수 있다. 하지만 이것은 프로세스단위의 이벤트가 아니기 때문에, Long-running 트랜잭션에서의 이벤트 처리라 보기 힘들다. BPSS도 <timetoPerform>이라는 엘리먼트를 두었지만 정확히 Long-running 트랜잭션에서의 이벤트 처리라 보기 힘들다.

### 6. 협업 프로세스 언어 분석

5장의 엘리먼트 비교를 통해 각 표준안의 차이점을 기술하였다. 하지만 각 엘리먼트가 어떤 식으로 구성되는지 알 수 없었다. 즉, 협업 프로세스가 동작하기 위해 각 조직들이 비즈니스 프로세스 및 웹 서비스를 어떻게 할당하는 지 알 수 없었다. 그래서 본 장에서 WSCI[9]에 기술된 여행 예약(Plan and Book Trip) 예제를 기반으로, 각 비즈니스 프로세스 언어의 구성 및 비교를 종합하여 기술한다. 여행 예약은 (그림 3)과 같이 고객(Traveler), 여행사(Travel Agent), 항공사(Airline)가 참여하는 협업 프로세스이다. 고객은 여행을 예약하고, 여행사는 여행에 따른 이용 가능한 비행 편을 항공사로부터 정보를 얻어 고객에게 추천한다. 항공사는 예약이 완료된 항공 티켓을 고객에게 전달한다.

(그림 3)은 5개의 표준안들이 여행 예약의 협업 프로세스를 구성하기 위해 각 조직이 갖추어야 하는 파일들을 명시하였다. 모든 표준안들이 작업(액티비티)을 위해 웹 서비스를 기반으로 두고 있어, 각 파티들은 다른 파티를 위해 웹 서비스를 제공해야 한다. 예를 들어, 고객은 여행사가 제공하는 'OrderTrip' 함수를 이용해 여행 예약을 시작할 수 있다. 5.1 절의 기술에 의해, BPSS의 액티비티가 웹 서비스로 실제 작업을 구성하는 지 알 수 없어 (그림 3)에 점선 박스로 표시하였다.



(그림 3) 여행 예약 시스템의 협업 프로세스 표준안들의 구성도

(그림 3)의 구성은 실제 부록 A의 예제처럼 구성될 수 있다. 부록 A는 (그림 1)에서 고객(Traveler)과 여행사(Travel Agent) 사이에 여행 예약에 관련된 메시지 교환을 실제 XML 문서로 표현한 예제이다. 부록 A에서 암시적 협업 표준안들은 각 파티마다 고유의 XML을 가져야 하는 반면에, 명시적 협업 표준안들은 하나의 XML을 각 파티가 공유하는 것을 알 수 있다. 부록 A의 메시지 교환은 고객이 여행사가 제공하는 'OrderTrip'이라는 함수를 접근하고 예약에 관련된 응답을 받는 예제이다. 즉, 동기적으로 메시지 교환이 동작한다.

3장에서 기술한 것처럼, 협업 프로세스는 크게 명시적 협업과 암시적 협업으로 구성하는데, 둘의 차이는 (그림 3)과 부록 A에서 확연히 드러난다. (그림 3)과 부록 A에서 BPML, BPELWS, WSCI는 각 파티가 독립적인 자신의 협업 프로세스 문서를 가지는 반면, WS-CDL, BPSS는 공통의 협업 프로세스 문서 하나만을 각 파티가 공유한다. <표 8>을 보면 조직 관점에서 둘의 차이가 분명하다. 부록 A에서 WS-CDL의 <relationshipType>과 BPSS의 <fromRole>, <toRole>은 특정 상대방을 대상으로 역할을 구성하였지만, BPELWS, BPML WSCI의 역할은 그렇지 않다. 이것은 명시적 협업 표준안이 사전에 각 파티간의 동의를 있었다는 것을 의미한다. 즉, 명시적 협업 표준안이 암시적 협업 표준안보다 응집력(Cohesion)이 좋다고 할 수 있다.

<표 8> 협업 프로세스 언어의 필수 요소 비교 (O: 유, X: 무, ~: 일부)

		BPELWS	BPML	WSCI	WS-CDL	BPSS
협업 환경		명시적			암시적	
조직 관점	파트너	X	X	X	O	O
	역할	~	~	~	O	O
	행위	O	O	O	O	X
	동의	X	X	X	O	X
대화적 관점	동적 참여	~	O	O	O	X
	참여자	~	~	~	O	O
	연산	O	O	O	O	X
	시맨틱 일관성	O	O	O	O	X
행동적 관점	시퀀스	O	O	O	O	O
	AND	O	O	O	O	O
	XOR	O	O	O	O	~
	루프	O	O	O	O	X
정보적 관점	조건	X	X	X	O	O
	데이터 타입	~	~	~	O	~
	변수	O	O	O	O	X
	데이터 플로우	O	O	X	O	~
트랜잭션 관점	예외 처리	O	O	O	~	~
	보상 처리	O	O	O	O	X
	이벤트 처리	O	O	O	~	~

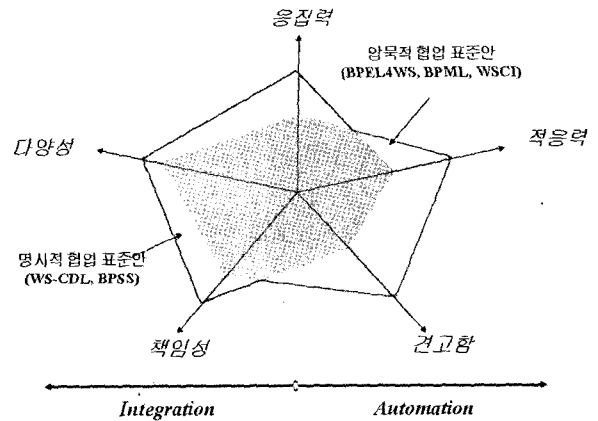
협업 프로세스 표준안은 XPDL과 같은 내부 프로세스 지향의 언어[22]와는 달리 기본적인 액티비티가 두 파티 간의 통신이기 때문에, 대화적 관점의 기능은 모든 표준안이 유사하다. 하지만 5.2절에서 기술한 것처럼, 명시적 협업이 암시적 협업보다 각 파티간의 책임성(Responsibility)을 고려한다. 부록의 B에서 WS-CDL의 <record>는 교환되는 메시지를 기록한다. 협업 비즈니스는 여러 조직에 의해 구성되기 때문에, 버그, 시스템 실패, 침입과 같은 예외 상황에 다수의 조직이 피해를 입는다. 이때, 조직들의 사이의 피해를 최소화시키기 위해 예외 상황에 대한 책임을 결정지를 필요가 있다. 예를 들어, 메시지 교환에서 잘못된 인수의 메시지로 인해 시스템이 오동작을 일으킨다면, 교환되는 메시지 기록을 통해 문제의 책임을 지을 수 있다. 만일 명세서 레벨에서 정의되지 않으면, 애플리케이션의 레벨에서 고려되어야 한다. 이것은 애플리케이션에 부담을 줄 뿐만 아니라, 이를 위해 새로운 기술을 설정하고 파티와 협정을 맺어야 한다. 만일 다른 파티가 다른 기술을 요구할 때는 애플리케이션의 비중이 더욱 커진다.

행동적 관점은 액티비티의 실행 순서를 결정하는 것이기 때문에, <표 8>과 같이 대부분의 표준안이 기본적인 기능들을 제공한다. 5.3절에서 기술한 것처럼, 명시적 협업이 암시적 협업보다 다양성(Diversity) 플로우를 구성할 수 있다. 하지만 이것은 머신이 이해하기 어렵기 때문에, 협업 프로세스를 계산하는 속도 면에서 암시적 협업이 더 좋은 성능을 가질 것이다. 또한 디자인된 프로세스의 유효성 검사(Validation Check)에서도 암시적 협업이 더 쉽게 적용될 수 있다. 즉, 암시적 협업 표준안이 명시적 협업 표준안보다 머신에 대한 적응력(Adaptivity)이 좋다고 할 수 있다.

부록 A에서 메시지 교환에 고려되는 데이터들을 자세히 확인할 수 있다. WSCI를 제외한 모든 표준안이 메시지 교환에서 데이터를 명세서에 기술한다. 하지만 전달받은 데이터를 프로세스에서 다시 이용할 수 있기 때문에, BPSS의 문서 단위인 <businessDocument>보다는 프로퍼티 단위인 <variable>이 더 조작하기 쉽다. 데이터의 흐름은 프로세스의 흐름을 결정 짓기 때문에 다른 표준들과 상호운용하기 위해서는 XSD와 같은 기존 포맷을 이용하는 것이 적합하기 때문에, BPSS를 제외한 모든 표준안이 XSD를 이용한다.

협업 프로세스의 증가로 트랜잭션의 개념을 중요시하게 되었다. 그래서 <표 8>에서처럼 모든 표준안이 예외 처리, 보상 처리와 이벤트 처리를 제공한다. 하지만 각 표준안에 따라 집중하는 영역과 기능에 차이가 있다. 5.5절에서 기술한 것처럼, 명시적 협업은 메시지 교환단위에 발생할 수 있는 예외나 이벤트를 처리하는 반면에, 암시적 협업은 프로세스 단위에서 발생할 수 있는 예외나 이벤트를 처리하였다. 메시지 교환에서 발생할 수 있는 예외나 이벤트는 모두 프로세스 단위에서 처리할 수 있기 때문에, 암시적 협업 표준안이 명시적 협업 표준안보다 트랜잭션 관점에서 더 견고(Robustness)하다고 할 수 있다.

<표 8>은 5장의 엘리먼트 비교를 총괄한 표로, 각 표준안



(그림 4) 명시적 협업과 암시적 협업 표준안의 특성

이 5가지 관점에 대해 서로 조금씩 차이를 갖는 것을 알 수 있다. 하지만 그 차이는 기능상의 차이이기 보다는 본 논문에서 정의한 협업의 접근 방식의 차이이다. 사실 어떤 환경이 협업 비즈니스 환경에 더 적합한지는 알 수 없다. 하지만 본 논문의 비교를 통해, 대화적 관점이나 조직 관점에서는 WS-CDL과 BPSS가 책임성, 응집력이 더 좋은 것을 알 수 있다. 반면에 행동적 관점이나 트랜잭션 관점에서는 BPML, BPELWS, WSCI가 적응력이나 견고함이 더 좋다는 것을 알 수 있다. 결론적으로 (그림 4)와 같이 명시적 협업은 이기종의 프로세스 통합 측면에서 뛰어나며, 암시적 협업은 빠른 자동화 측면에서 뛰어나다는 것을 알 수 있다.

### 7. 결론

본 논문은 협업 비즈니스 환경에서 협업 프로세스를 구성하는 여러 비즈니스 프로세스 언어들(BPML, BPELWS, WSCI, WS-CDL, BPSS)을 비교하였다. 이들은 협업 비즈니스 환경에 적합하도록 구성된 표준안으로써, 각각 애플리케이션의 사용 목적에 따라 조금씩 차이를 가지고 있다. 이러한 차이는 협업 비즈니스 환경의 상호 운용성을 악화시키며, 협업 비즈니스의 비용적 손실을 초래할 수 있다. 그래서 본 논문은 협업 프로세스 환경의 적합한 표준안 선택에 도움이 되고자, 5개의 대표적인 협업 프로세스 표준안을 비교분석한다.

본 논문은 Bernauer가 제안한 프레임워크를 기반으로 협업 프로세스 모델링 절차에 따른 협업 프로세스 필수 요소들을 추출하였다. 이들을 기반으로 5개 표준안들의 엘리먼트 일대일 비교를 하였다. 이러한 비교는 각 표준안이 제공하는 기능들을 구체적으로 구별해 줄 뿐만 아니라 표준안들의 장/단점 및 제한점을 찾게 해주었다. 마지막으로, 완결된 하나의 예제를 통하여 명시적 협업과 암시적 협업 표준안의 특성을 발견하였다.

### 참고 문헌

[1] Bill Terry, "Implementing Business Process Management,"

Business Integration Journal, pp.1-4, May, 2004.

[2] Mazumdar and M. AbuSafiya, "Document-Centric Approach to Business Process Management," Document Engineering Proceedings of the 2004 ACM Symposium on Document Engineering, 2004.

[3] Leymann, D. Roller and M. T. Schmidt, "Web Services and Business Process Management," IBM Systems Journal, Vol. 41, No.2, pp.198-211, 2002.

[4] T. Andrews, F. et al, "Business Process Execution Language for Web Services (BPEL4WS) Version 1.1," May, 2003.

[5] A. Arkin, "Business Process Modeling Language," Business Process Management Initiative proposed recommendation, 2003.

[6] Arkin, A. et al, "Web Service Choreography Interface (WSCl) Version 1.0," W3C Note 8, Aug., 2002.

[7] N. Kavantzias and D. et al, "Web Service Choreography Description Language (WS-CDL) Version 1.0," W3C Working Draft, Oct., 2004.

[8] J. Clark and C. et al, "eXML Business Process Specification Schema (BPSS) Version 1.01," UN/CEFACT and OASIS Specification, May, 2001.

[9] J. Mendling, G. Neumann and M. Nüttgens, "A Comparison of XML Interchange Formats for Business Process Modelling," Lecture Notes in Informatics (LNI), Vol.56, pp.129-140, Oct., 2004.

[10] Jan Mendling, "A Survey on Design Criteria for Interchange Formats," Technical Report JM-2004-06-02. Vienna University of Economics and Business Administration, 2004.

[11] Martin Bernauer, Geri Kappel and Gerhard Karmler, "Comparing WSDL-based and ebXML-based Approaches for B2B Protocol Specification," Proceedings of the 1st International Conference on Service-Oriented Computing (ICSOC-03), pp.225-240, Dec., 2003.

[12] S. Rausch-Schott. "TriGSflow - Workflow Management Based on Active Object-Oriented Database Systems and Extended Transaction Mechanism," PhD thesis, University at Linz, 1997.

[13] M. Bernauer, G. Kappel, G. Kramler and W. Retschitzegger, "Specification of Interorganizational Workflows-A Comparison of Approaches," Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003), pp.30-36, July, 2003.

[14] Robert Shapiro, "A Technical Comparison of XPDL, BPML and BPEL4WS," 2002.

[15] Michael Stollberg, Dumitru Roman and Juan Miguel Gomez, "A Mediated Approach towards Web Service Choreography," Nov., 2004.

[16] Jan Mendling and Michael Hafner, "From Inter-Organizational Workflows to Process Execution: Generating

BPEL from WS-CDL," Proceedings of OTM 2005 Workshops. Lecture Notes in Computer Science 3762, Agia Napa, Cyprus, pp.506-515, No., 2005.

[17] BPML.org, "Business Process Modeling Notation (BPMN) Version 1.0," May, 2004.

[18] Victor F. A. Santander, Jaelson Castro, "Deriving Use Cases from Organizational Modeling," Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering, pp.32-42, Sep., 2002.

[19] Benchaphon Limthanmaphon and Yanchun Zhang, "Web service composition transaction management," ACM International Conference Proceeding Series, Vol.52, pp.171-179, 2004.

[20] ebXML Trading-Partners Team, "Collaboration-Protocol Profile and Agreement Specification. Version 1.0," UN/CEFACT and OASIS, May, 2001.

[21] Martin Gudgin and Marc Hadley, "Web Service Addressing Version 1.0," W3C Candidate Recommendation, Aug., 2005.

[22] WfMC, "Workflow Process Definition Language-XML Process Definition Language," Document Number WFMC-TC-1025, Documentation Status - 1.0 Final Draft, October 25, 2002 Version 1.0.

[23] Guy Pardon and Atomikos, "Business Transactions, Compensation and the Try-Cancel/Confirm (TCC) Approach for Web services," 2004.

<부록 A> 그림 1에서 고객과 여행사의 메시지 교환에 따른 표준 안 비교

	Traveler	Travel Agent
BPEL 4WS	<pre>&lt;invoke partnerLink="OrderTripPL" portType="ta:OrderTripPT" operation="OrderTrip" inputVariable="tripOrderReq" outputVariable="tripOrderAck" /&gt;</pre>	<pre>&lt;receive partnerLink="OrderTripPL" portType="tns:OrderTripPT" operation="OrderTrip" variable="tripOrderRequest" /&gt; &lt;reply partnerLink="OrderTripPL" portType="tns:OrderTripPT" operation="OrderTrip" variable="tripOrderAck" /&gt;</pre>
BPML	<pre>&lt;action name="RequestOrderTrip" portType="wsdl:TAtoTraveler" operation="OrderTrip"&gt;   &lt;output element="defs:tripOrderReq"&gt;     &lt;source property="tripOrderRequest" /&gt;   &lt;/output&gt;   &lt;input element="defs:tripOrderAck" property="tripOrderAck" /&gt; &lt;/action&gt;</pre>	<pre>&lt;action name="ReceiveTripOrderTrip" portType="wsdl:TAtoTraveler" operation="OrderTrip"&gt;   &lt;input element="defs:tripOrderReq" property="tripOrderRequest" /&gt;   &lt;output element="defs:tripOrderAck"&gt;     &lt;source property="tripOrderAck" /&gt;   &lt;/output&gt; &lt;/action&gt;</pre>
WSCl	<pre>&lt;action name="RequestOrderTrip" role="tns:Traveler" operation="tns:TravelerToTA/O</pre>	<pre>&lt;action name="ReceiveOrderTrip" role="tns:TravelAgent" operation="tns:TAtoTraveler/Or</pre>

	rderTrip"/>	derTrip"/>
WS-CDL	<pre> &lt;interaction name ="OrderTrip" channelVariable="tns:Traveler2TAC" operation="tns:TAToTraveler/OrderTrip" initiate="true"&gt; &lt;participate relationshipType ="tns:TravelerTravelAgentRelationship" fromRole="tns:Traveler" toRole="tns:TravelAgent" /&gt; &lt;exchange name ="TripOrderEx" informationType ="tns:orderReqT" action="request"&gt; &lt;send variable="cdl:getVariable(tns:OrderReq tns:Traveler)" /&gt; &lt;receivevariable= "cdl:getVariable(tns:OrderReq,tns:TravelAgent)" /&gt; &lt;/exchange&gt; &lt;exchange name="TripOrderEx" informationType ="tns:orderAckT" action="response"&gt; &lt;send variable= "cdl:getVariable(tns:OrderAck, tns:TravelAgent)" /&gt; &lt;receive variable= "cdl:getVariable(tns:OrderAck, tns:Traveler)" /&gt; &lt;/exchange&gt; &lt;timeout time-to-complete="cdl:getVariable(tns:CompleteTime)" fromRoleRecordReference="Traveler" toRoleRecordReference="TravelAgent" /&gt; &lt;record name="record-chanel-info" when ="after" causeException="true"&gt; &lt;source variable= "cdl:getVariable(tns:RecordReq tns:Traveler)" /&gt; &lt;target variable ="cdl:getVariable(tns:RecordAck tns:Traveler)" /&gt; &lt;/record&gt; &lt;/interaction&gt; </pre>	
BPSS	<pre> &lt;BusinessTransactionActivity name= "OrderTrip" nameID="BTA1" businessTransaction="OrderTrip" businessTransactionIDRef="BT1" fromRole="TravelerToTravelAgent" fromRoleIDRef="R1" toRole="TravelAgentToTraveler" toRoleIDRef="R3" /&gt; </pre>	



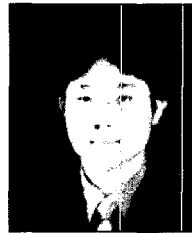
**조 명 현**

e-mail : mhjo@cse.hanyang.ac.kr  
 2004년 한양대학교 전자컴퓨터공학부  
 학사  
 2006년 한양대학교 컴퓨터공학과 석사  
 2006년~현재 코난테크놀로지 근무  
 관심분야: 그리드 컴퓨팅, 센서 네트워크,  
 시멘틱 웹, e-비즈니스



**박 정 업**

e-mail : jupark@cse.hanyang.ac.kr  
 2005년 한양대학교 전자컴퓨터공학부 학사  
 2005년~현재 한양대학교 컴퓨터공학과  
 석사과정  
 관심분야: e-비즈니스, 그리드 컴퓨팅,  
 센서 네트워크



**설 주 영**

e-mail : jysul@cse.hanyang.ac.kr  
 2004년 세명대학교 소프트웨어공학과 학사  
 2006년 한양대학교 컴퓨터공학과 석사  
 2006년~현재 LG CNS 근무  
 관심분야: e-비즈니스, 시멘틱 웹, 센서  
 네트워크



**백 문 홍**

e-mail : mhbaek@kitech.re.kr  
 1982년 서울대학교 제어계측학과 학사  
 1984년 서울대학교 제어계측학과 학사  
 1995년 동경대학 전기공학과 박사  
 1995년~1996년 한국과학기술연구원  
 박사후 연구원  
 1996년~현재 한국생산기술연구원 수석연구원  
 관심분야: 지능로봇, 3D 영상해석



**손 진 현**

e-mail : jhson@cse.hanyang.ac.kr  
 1996년 서강대학교 전산학과 학사  
 1998년 한국과학기술원 전산학과 석사  
 2001년 한국과학기술원 전자전산학과 박사  
 2001년 9월~2002년 8월 한국과학기술원  
 전자전산학과 박사후 연구원  
 2002년 9월~현재 한양대학교 컴퓨터공학과 조교수  
 관심분야: 데이터베이스, e-비즈니스, 유비쿼터스 컴퓨팅,  
 임베디드 시스템