

효율적인 비즈니스 프로세스 관리를 위한 J2EE 기반 B2Bi 협업 워크플로우 시스템 설계 및 구현

이 창 목[†] · 장 옥 배^{**}

요 약

본 논문은 비즈니스 프로세스와 업무 로직을 분리하여 비즈니스 프로세스를 손쉽게 모델링하고 이를 바탕으로 협업을 가능하게 하는 B2Bi 협업 워크플로우 모델링 시스템을 설계 및 구현하였다. 협업 워크플로우 모델링 시스템은 비즈니스 프로세스 모델링 도구, 실행 엔진, 모니터링 도구 3개의 컴포넌트로 구성되어 있다. 첫째, 비즈니스 프로세스 모델링 도구는 애플리케이션의 비즈니스 로직을 신속, 정확하게 반영하는 프로세스 맵을 구축하는데 사용된다. 둘째, 실행 엔진은 비즈니스 프로세스 인스턴스를 위한 실시간 실행 환경을 제공한다. 셋째, 모니터링 도구는 현재 진행되고 있는 비즈니스 프로세스에 대한 실시간 모니터링 기능을 제공한다. 또한 기존 시스템인 레거시 시스템과의 연동을 위해 XML 및 J2EE 기반으로 유연성, 확장성을 지원할 수 있도록 하였으며 새로운 기업 전략과 운용에 필요한 해결책을 제시한다.

키워드 : 비즈니스 프로세스 관리, 협업 워크플로우, 모델링 도구, 실행 엔진, 모니터링 도구

Design and Implementation of B2Bi Collaboration Workflow System for Efficient Business Process Management based on J2EE

Chang-Mog Lee[†] · Ok-Bae Chang^{**}

ABSTRACT

In this paper, the business process was easily modeled by distinguishing between the business process and work logic. Based on this model, B2Bi collaboration Workflow modeling tool, which facilitates collaboration, was designed and implemented. The collaboration workflow modeling tool consists of 3 components: business process modeling tool, execution engine and monitoring tool. First, a business process modeling tool is used to build a process map that reflects the business logic of an application in a quick and accurate manner. Second, an execution engine provides a real-time execution environment for business process instance. Third, a monitoring tool provides a real-time monitoring function for the business process that is in operation at the time. In addition to this, it supports flexibility and expandability based on XML and J2EE for the linkage with the legacy system that was used previously, and suggests a solution for a new corporate strategy and operation.

Key Words : Business Process Management, Collaboration Workflow, Modeling Tool, Execution Engine, Monitoring Tool

1. 서 론

비즈니스 환경의 변화는 새로운 비즈니스 모델의 등장에 따른 새로운 비즈니스 프로세스의 요구이다. 정보기술의 발달로 인하여, 기업 간 또는 기업 내 업무의 통합화가 가속화되면서, 새로운 프로세스의 필요성이 제기되고 있다. SCM(Supply Chain Management) 등이 그 대표적인 예로서, 하나의 제품을 생산하는데 관련된 모든 기업들의 물류를 하나로 보고 이를 최적화하고자 하는 것이다[1]. BPM(Business Process Management)은 기업 내외의 비즈니스

프로세스를 가시화하고, 비즈니스의 수행과 관련된 사람, 시스템을 프로세스에 맞게 실행/통제하며, 전체 비즈니스 프로세스를 효율적으로 관리하고 최적화할 수 있도록 지원하는 시스템이다. 기업의 활동은 날로 복잡성이 증가되고 있으며, 대내외적인 경영효율의 극대화가 기업 성공의 최대 관건으로 대두되고 있다. 이러한 현실 속에서 기업은 효율적인 경영을 위해 고객, 파트너, 임직원 및 정보 시스템과 비즈니스 프로세스 등이 효율적으로 운영/관리되어야 함에도 불구하고 서로 분리되고 통합적이지 못한 것이 현실이다. 이는 업무의 중복, 업무 시간의 증가, 데이터 중복 등의 손실을 초래하며, 특히 급격히 변화하는 시장 상황, 고객의 요구 및 내부 조직의 변경 등으로 인한 비즈니스 프로세스 변화에 신속히 대응하지 못하는 주요한 원인으로 작용하고

† 준 회 원 : 전북대학교 전자정보 BK21 Post-Doctor
** 정 회 원 : 전북대학교 전자정보공학부 교수
논문접수 : 2006년 8월 11일, 심사완료 : 2007년 1월 15일

있다. 업무 생산성과 효율성을 높이기 위해서는 비즈니스 프로세스를 중심으로 이러한 단위 업무들을 연계하고 자동화 시킬 필요가 있으며 이를 가능케 하는 솔루션이 바로 BPM 이다.

1970년대 초 SQL의 등장으로 데이터 계층과 비즈니스로직 계층이 분리되어 시스템의 확장과 수정이 혁신적으로 용이해졌다. 또한 1990년대 웹 혁명은 거대한 클라이언트/서버 네트워크 기술을 통해 사용자 측의 다양한 표현과 업무로직을 분리시켜 업무로직의 변경을 용이하게 하였다. 이러한 기술을 바탕으로 현대의 비즈니스 애플리케이션은 다양한 플랫폼에서 운용되는 여러 애플리케이션들 간의 통합과 비즈니스 환경에 민첩하게 대응할 수 있는 프로세스 모델링 및 관리를 요구하기에 이르렀다. 이를 기반으로 기업은 레거시 시스템을 포기하지 않고 새로운 시스템 구축에 포함할 수 있어 비용을 절감할 수 있을 뿐 아니라 기업 간 협업을 통하여 새로운 기회를 만들어낼 수 있다.

본 논문에서는 비즈니스 프로세스와 업무로직을 분리하여 비즈니스 프로세스 관리 솔루션을 도입 운용하는 기업들이 급변하는 비즈니스 환경에 신속하게 대응할 수 있고 레거시 시스템과 연동하여 새로운 비즈니스 전략과 운용에 필요한 J2EE기반 B2Bi 협업 워크플로우 모델링 도구를 설계 및 구현하고자 한다.

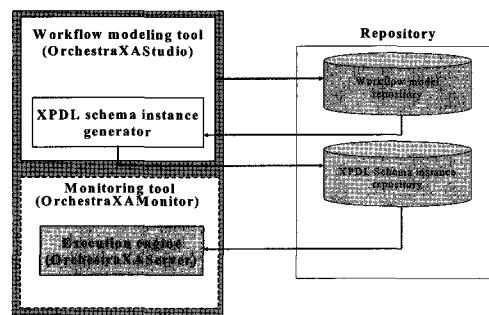
2. 관련 연구

워크플로우 관리 시스템은 다수의 상이한 정보 자원의 통합과 비즈니스 프로세스의 전반적인 수행 상황을 모니터링하기 위한 가장 효과적이고 현실적인 도구이다[2]. 초기에는 BPR(Business Process Reengineering)을 지원하기 위한 비즈니스 프로세스 자동화 도구로서 시작되었지만, groupware, ERP, EAI(Enterprise Application Integration), BPMS, ebXML, e-commerce 등의 분산 컴퓨팅의 협업을 위한 핵심 기술로서 자리 잡고 있다. 워크플로우가 많은 분야에 적용되고 있기 때문에 여러 표준이 제정되었고 앞으로 새로운 표준이 제정될 것이다. 워크플로우 관리 시스템이 중앙 집중식으로 구현되는 이유는 모니터링을 보다 용이하게 하기 위함이다. WfMC(Workflow Management Coalition)에서는 최초의 워크플로우 표준인 WfMC 참조 모델[3]을 만들었다. 이 모델에는 워크플로우 엔진, 주변 구성요소, 이들 간의 인터페이스 및 용어에 대한 표준안을 제정하였다. 웹의 확산으로 HTTP 기반의 워크플로우 표준으로 SWAP(Simple Workflow Access Protocol) [4] 이 발표되었다. 그 후 XML 표준인 Wf-XML로 통합되었다[5]. Wf-XML은 워크플로우 시스템간의 상호운용을 위한 XML 기반 프로토콜로서 상호운용을 위한 최소한의 메시지 집합을 제공한다. 이후 INCAs(INformation CArriers)[6]와 같은 에이전트 기반 워크플로우 모델이 제안되었다. 그러나 완전 분산 워크플로우 모델이기 때문에 매우 높은 확장성을 가지나 워크플로우에서 가장 중요한 기능인 모니터링과 관리 기

능에 대한 고려가 전혀 없었다. WONDER 시스템[7]은 INCAs의 모델을 실제로 구현한 것인데 INCAs 모델에는 없는 모니터 기능을 추가하였다. 웹상에서의 워크플로우를 지원하기 위해 WfMC에서 WPDL(Workflow Process Description Language)를 제정하였고, 후에 XML을 적용하여 XPDL(XML Process Description Language)로 재개정되었다. XPDL은 워크플로우 정의 표준으로 되었지만 ebXML의 BPSS(Business Process Specification Schema)와 RosettaNet의 PIP(Partner Interface Process) 등의 독자적 프로세스 정의 표준들이 있다[8]. 웹 서비스의 출현과 급속한 표준화는 기본 기능의 표준화 뿐 아니라 워크플로우 기능 제공을 위한 표준으로 여러 단체 및 회사에서 다양한 웹 서비스를 위한 워크플로우 정의 표준을 제안하였다. 대표적인 예로는 IBM의 WSFL, Microsoft의 XLANG 등이 있고 최근에는 이 두 모델을 통합한 BPELWS(Business Process Execution Language for Web Services)로 통합되었다[8, 9].

3. B2Bi 협업 워크플로우 도구의 설계

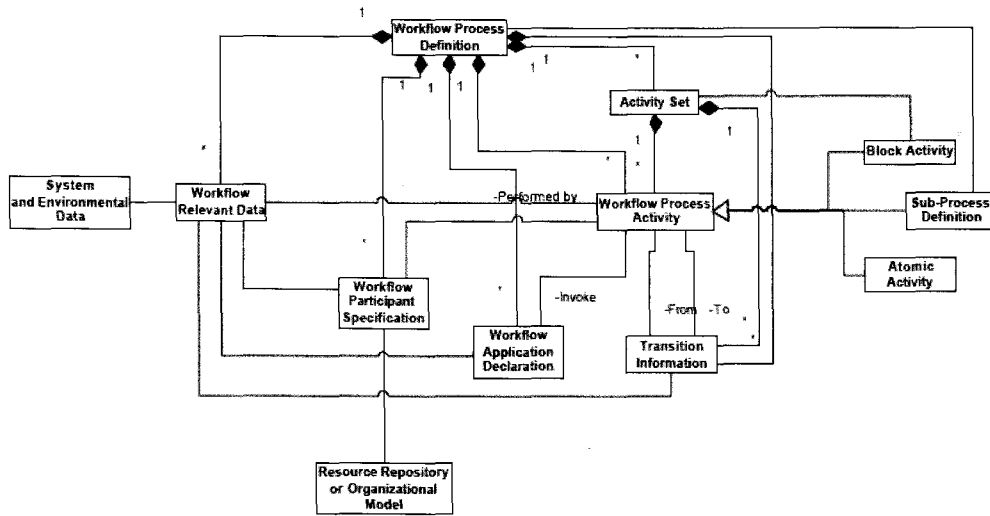
본 논문의 목표는 비즈니스 프로세스를 설계하고 실행하며 관리하기 위한 B2Bi용 협업 워크플로우 도구를 설계 및 구현하는 것이다. 이를 위해 비즈니스 프로세스를 모델링할 수 있는 자동화 도구인 모델러(OrchestraXASStudio)가 필요하다. 모델러에 의해 작성된 프로세스는 워크플로우 프로세스 정의 언어인 XPDL을 사용하여 정의되어야 한다. 따라서 XPDL 스키마 인스턴스 생성기가 필요하다. 또한 워크플로우 또는 프로세스 통합 애플리케이션의 신속한 배치와 유연한 실행을 위한 완전한 J2EE 기반의 프로세스 실행 엔진(OrchestraXAServer)이 요구된다. 마지막으로 이러한 실행 엔진에서 이루어지는 프로세스를 감시하고 운용할 수 있는 콘솔을 제공하는 모니터링 도구(OrchestraXAMonitor)가 필요하다. (그림 1)은 B2Bi 협업 워크플로우의 구조를 보여주고 있다.



(그림 1) 협업 워크플로우 도구의 구조

3.1 비즈니스 프로세스 메타 모델

메타 모델은 프로세스 정의, 그들 사이의 관계 그리고 속성들 내에 포함된 최상위 수준의 개체들을 기술한다. 관련된 프로세스 모델로의 프로세스 정의를 그룹화하기 위한 다



(그림 2) 워크플로우 프로세스 정의의 메타 모델

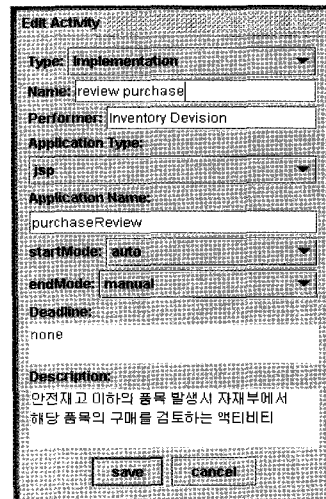
양한 규약과 서로 다른 프로세스 정의와 모델들에서 사용할 수 있는 공통 정의 자료를 정의한다. 각각의 개체들에 대하여 개체들의 특징을 기술하는 관련된 속성들의 집합이 있다.

3.1.1 워크플로우 프로세스 정의(Workflow Process Definition)

프로세스 정의 개체는 프로세스 내 다른 개체에 적용되는 문맥 정보를 제공한다. 프로세스 자체에 대해서는 컨테이너 역할을 하며 프로세스 정의 생성 날짜, 저자와 같은 프로세스 관리에 관련된 정보를 제공하거나 프로세스 실행 중에 사용되는 초기화 매개변수, 실행 우선순위, 검사되어야 할 시간제한, 통지 받을 사람, 모의실험 정보를 제공한다.

3.1.2 워크플로우 프로세스 액티비티(Workflow Process Activity)

프로세스 정의는 하나 이상의 액티비티로 구성된다. 각각의 액티비티는 논리적이면서 일체화된 작업 단위이다. 하나의 액티비티는 참여자의 할당으로 명세되는 자원과 응용 프로그램 할당으로 명세되는 컴퓨터 애플리케이션의 조합에 의해 수행되는 작업을 대표한다. 여기에는 워크플로우 관리 시스템이 자동으로 시작과 종료를 관리할 것인지에 대한 정보나 다른 액티비티와 경쟁적으로 관련될 수 있는 시스템 자원과 서비스에 대한 우선순위에 관련된 정보가 추가될 수 있다. 또한 워크플로우 관련 데이터 항목의 사용 내역이 명세될 수 있다. 그리고 하나의 액티비티는 서브프로우일 수 있는데, 이 경우 개별적으로 명세된 프로세스 정의 실행을 위한 컨테이너로서 사용된다. 서브 플로우에서 확인된 프로세스 정의는 자신의 액티비티 정의, 내부 전이, 자원 그리고 애플리케이션 할당을 포함한다. In- 그리고 Out- 매개변수는 호출하는 그리고 호출되는 프로세스 사이에 필요한 워크플로우 관련 데이터 교환을 허용한다. 하나의 액티비티는 액티비티 집합을 수행하는 블록 액티비티일 수 있다. 액티비티 집합 내 액티비티와 전이는 자신을 포함하는 프로세스의 네임 스페이스를 공유한다. 마지막으로 더미 액티비티는 아무런 자원 요구나 애플리케이션 처리를 수행하지 않지만



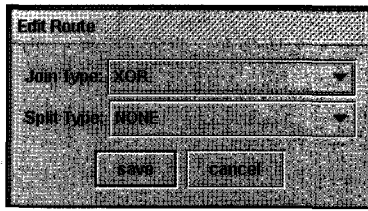
(그림 3) 액티비티 대화상자

내부 전이와 외부 전이 사이의 라우팅 결정을 지원한다. (그림 3)은 액티비티를 관리하는 대화상자를 나타낸다.

3.1.3 전이 정보(Transition Information)

액티비티들은 흐름 제어 정보를 통해 서로 간에 관련되어 진다. 각각의 개별 전이는 세 가지 기본 속성인 from-액티비티, to-액티비티 그리고 전이가 이루어지는 조건으로 이루어져 있다. 하나의 액티비티로부터 다른 액티비티로의 전이는 조건적이거나 무조건적일 수 있다. 하나의 프로세스 내 전이들은 프로세스 내 개별 액티비티들의 순차적이거나 병렬적 연산 결과를 나타낼 수 있다. 관련된 분기 또는 조인 조건에 대한 정보는 적절한 액티비티 내에 정의된다. 단순한 기본적인 전이와 from- 그리고 to- 액티비티와 관련된 분기 그리고 조인 함수를 사용하여 표현될 수 없는 좀 더 복잡한 전이들은 분기 그리고(또는) 조인 연산들의 추가적인 조합을 허용하는 실제 액티비티들 사이의 중간 단계로서 명세될 수 있는 더미 액티비티들을 사용하여 형성된다. 기본 전이 개체와

더미 액티비티를 사용하여 임의의 복잡한 라우팅 구조를 명세할 수 있다. (그림 4)는 라우팅 대화상자를 나타낸다.



(그림 4) 라우팅 대화상자

3.1.4 워크플로우 참여자 선언(Workflow Participant Specification)

워크플로우 참여자 선언은 프로세스 정의 안에서 다양한 액티비티들의 수행자로서 활동할 수 있는 자원들의 기술을 제공한다. 특정 액티비티를 수행하도록 할당될 수 있는 특정 자원들은 액티비티의 속성으로써 명세된다. 워크플로우 참여자 선언은 반드시 사람이 필요는 없다. 다시 말해서, 적절한 역량이나 책임을 가진 사람들의 집합이거나 자동화된 시스템일 수 있다. (그림 5)는 참여자를 관리할 수 있는 대화상자를 나타낸다.



(그림 5) 참여자 대화상자

3.1.5 워크플로우 자원 저장소(Workflow Resource Repository)

자원 저장소는 참여자가 사람, 프로그램 또는 시스템일 수 있다는 내용을 설명한다. 더 복잡한 시나리오에서는 참여자 선언이 인적 자원의 조직 모델일 수 있는 자원 저장소를 가리킬 수도 있다.

3.1.6 워크플로우 애플리케이션 선언(Workflow Application Declaration)

워크플로우 애플리케이션 선언은 IT 애플리케이션이나 각각의 액티비티와 연관된 프로세싱을 지원하거나 또는 전적으로 자동화하기 위한 워크플로우 서비스에 의해 호출될 수 있는 인터페이스 기술을 제공한다. 그러한 애플리케이션들은 일반적인 산업 도구, 특정 부서 또는 기업의 서비스, 또는 워크플로우 관리 시스템의 프레임워크 내에서 구현된 로컬 프로시저일 수 있다. 워크플로우 애플리케이션 정의는 어떠한 매개변수가 통과될 지를 포함하여 워크플로우 엔진과 애플리케이션 또는 인터페이스 사이의 인터페이스를 반

영한다.

3.1.7 워크플로우 관련 데이터(Workflow Relevant Data)

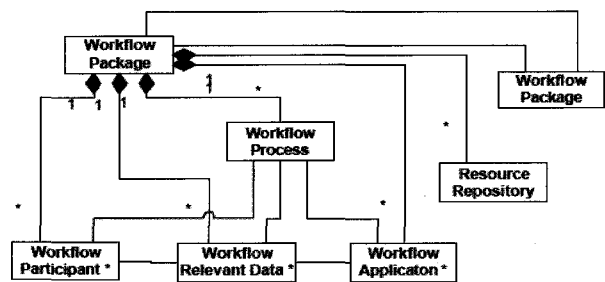
워크플로우 관련 데이터는 프로세스 실행 동안 각각의 프로세스 인스턴스 안에서 생성되고 사용되는 데이터를 정의한다. 이 데이터들은 워크플로우가 실행되는 동안 액티비티 또는 애플리케이션들에 유용하게 사용되며 전이나 참여자 할당에서처럼 조건 표현식의 평가를 위한 활동 사이의 임시 결과 또는 영구 저장 정보로써 사용된다. 애플리케이션 그리고(또는) 전이 조건에 의해 호출된 액티비티들은 워크플로우 프로세스 관련 데이터를 참조할 수 있다.

3.1.8 시스템 및 환경 데이터(System & Environment Data)

시스템 및 환경 데이터는 워크플로우 관리 시스템이나 로컬 시스템 환경에 의해 유지되지만 워크플로우 액티비티들에 의해 접근될 수 있으며 워크플로우 관련 데이터와 같은 방식으로 조건식 평가에 있어 워크플로우 관리 시스템에 의해 사용될 수 있다.

3.2 패키지 메타 모델

패키지 메타 모델은 프로세스 모델의 교환 또는 저장을 위한 개체와 속성을 규정한다. 이것은 개별 프로세스 정의를 참여자 명세를 위한 개체 정의, 애플리케이션 선언 그리고 개별 프로세스 정의 수준보다 패키지 수준에서 정의될 수 있는 워크플로우 관련 데이터를 연관시키는 다양한 상속 규칙을 정의한다. 패키지 정의는 패키지 내에 포함된 모든 개별 프로세스 정의들에 적용될 수 있는 다수의 공통 프로세스 정의 속성 명세를 허용한다. 패키지 메타 모델은 워크플로우 프로세스 정의, 워크플로우 참여자 명세, 워크플로우 애플리케이션 선언, 워크플로우 관련 데이터 등의 개체들을 포함한다. (그림 6)은 패키지 정의 메타 모델을 보여준다.



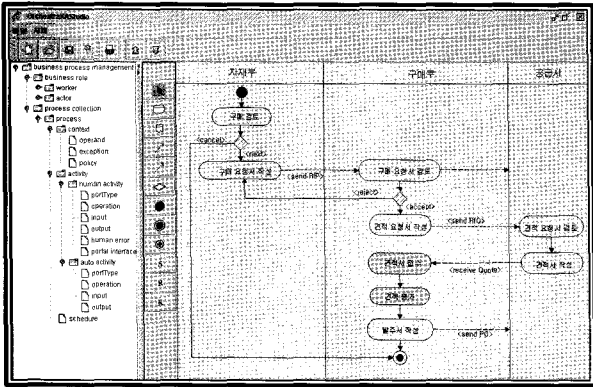
(그림 6) 패키지 정의 메타 모델

4. B2Bi 협업 워크플로우 도구의 구현

4.1 비즈니스 프로세스 모델링 도구

비즈니스 프로세스 모델링 도구는 기업의 전략과 정책 그리고 프로세스를 설계하고 모델링하는데 사용된다. 기업의 공통적인 활동을 선진 경영 기법(Best Practice)이라는 템플

릿으로 제시하여 일선 업무 설계자 및 관리자가 이를 바탕으로 새로운 업무 로직을 용이하게 추가 변경하도록 한다. 기존 XML 기반 프로세스 구문 작성을 통한 업무 프로세스 작성의 단점인 가파른 학습 곡선을 보완하기 위해 시각적이고 직관적인 모델링 인터페이스를 채택하였다. 모델링 도구를 이용하여 프로세스 설계, 단위 업무 설계, 프로세스 운영에 관련한 업무를 처리할 수 있다.



(그림 7) 비즈니스 프로세스 모델링 툴 주요 화면

첫째, OrchestraXAStudio는 드래그-앤-드롭 방식, 검증 기능, 위저드 기능 등 다양한 편집 기능을 사용하여 비전문가도 복잡한 비즈니스 프로세스를 손쉽게 설계할 수 있는 프로세스 모델링 툴을 제공한다. 이는 업무 규칙이 반영되어 실제로 운영이 가능한 프로세스 기반의 시스템을 구축하는 효과를 볼 수 있다. 또한, 설계된 프로세스는 비슷한 유형의 업무에 재사용될 수 있어 업무생산성을 높이는데 기여한다. 둘째, 프로세스 내의 단위 업무(task unit)에 대해 프로세스 참여자, 실행 애플리케이션, 단위업무 완료 조건 및 다양한 분기 조건 등의 속성을 손쉽게 정의 내리도록 한다.

셋째, 프로세스 운영은 비즈니스 프로세스가 시작되면 기 정의된 룰에 따라 업무가 진행된다. 프로세스 참여자는 업무 포털을 통해 업무를 처리하게 되는데 수행하여야 하는 업무를 실시간으로 통보받고, 기 정의된 전자 양식 등의 애플리케이션을 통하여 업무를 수행하게 된다. 또한 실시간으로 프로세스 및 프로세스 참여자를 변경할 수 있는 동적 프로세스 변경 기능과 임의 라우팅 기능을 제공함으로써 실제 업무 처리 과정에서 발생할 수 있는 다양한 예외상황(프로세스 흐름 변경, 업무 담당자 변경 등)을 실시간으로 반영할 수 있다.

4.2 비즈니스 프로세스 실행 엔진(OrchestraXAServer)

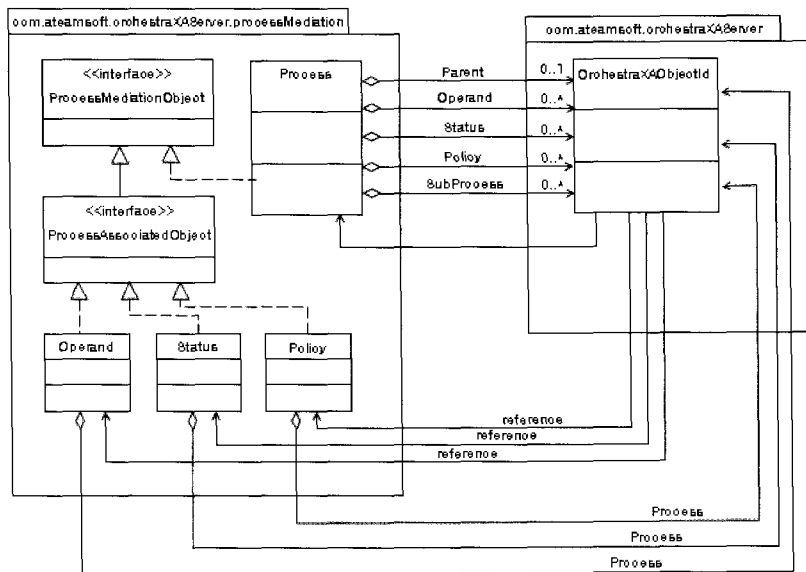
워크플로우 또는 프로세스 통합 애플리케이션의 신속한 배치와 유연한 실행을 위한 완전한 J2EE 기반의 프로세스 엔진이다. 일반적인 용어로 워크플로우 엔진이라고 하며 비즈니스 프로세스 인스턴스를 위한 실시간 실행 환경을 제공한다. 멀티 플랫폼이면서 애플리케이션 서버에 중립적인 J2EE 기반 프로세스 엔진으로서 다음과 같은 주요 개발 범위를 가진다.

- 프로세스 정의 해석
- 프로세스 인스턴스 제어(생성, 활성화, 일시 중단, 종료 등)
- 순차 또는 병렬 연산, 마감일 스케줄링
- 워크플로우 제어 데이터 및 워크플로우 관련 데이터 유지
- 외부 애플리케이션을 호출하는 인터페이스

4.2.1 OrchestraXAServer 구현

OrchestraXAServer는 모든 알려진 워크플로우와 프로세스 모델링 패턴뿐만 아니라 임의로 복잡한 비즈니스 또는 애플리케이션 프로세스 로직을 다룰 수 있다.

OrchestraXAServer에서 비즈니스 프로세스는 프로세스 객체, 오픈엔드 객체, 상태(Status) 객체, 정책(Policy) 객체



(그림 8) 객체 모델 개념도

의 서로 다른 객체 유형으로 모델링된다.

(1) 프로세스 객체

객체프로세스 객체는 프로세스의 현재 상태에 대한 정보와 프로세스의 시작, 종료, 상태 변경에 대한 조건 그리고 프로세스 객체가 다른 객체들과 맺고 있는 관계를 포함하고 있는 서버 객체 모델의 주요 구성 요소이다. 프로세스 객체는 다음의 속성과 연관을 가진다.

- 속성: 없음
- 연관

parent process	프로세스 객체의 하위 프로세스 객체에 대한 OrchestraXObjectId
subprocesses	프로세스 객체의 하위 프로세스 객체에 대한 OrchestraXObjectIds 집합
operands	프로세스 객체와 연관된 오퍼랜드 객체들의 OrchestraXObjectIds 집합
statuses	프로세스 객체와 연관된 상태 객체들의 OrchestraXObjectIds 집합
current statuses	프로세스에 현재 적용된 상태들
policies	프로세스 객체와 연관된 정책 객체들의 OrchestraXObjectIds 집합

(2) 오퍼랜드 객체

오퍼랜드 객체는 프로세스 객체에 의해 모델링 되어진 비즈니스 프로세스에 관련된 임의의 데이터를 캡슐화 한다. 문자열로 표현될 수 있는 어떠한 값도 저장할 수 있다. 객체들 또는 복잡한 자료구조도 XML 표현 변환이나 Java 직렬화(serialization)를 통하여 저장할 수 있다. 다음과 같은 속성과 연관을 가진다.

- 속성

ID	OrchestraXAServer 시스템 내부의 객체를 유일하게 확인하는 문자열
label	프로세스를 확인하는데 id 대신 사용될 수 있는 짧고, 사람이 읽어 그 의미를 알 수 있는 문자열. 프로세스와 관련 있는 모든 객체들은 유일한 레이블을 가져야만 한다.
description	프로세스를 기술하는 텍스트. OrchestraXAServer 클라이언트에 의해 사용된다.
ACL(set of ACEs)	OrchestraXAServer 객체와 시스템 사용자 사이의 관계를 설정한다.
visible in subtree (true, false)	오퍼랜드 객체의 영역을 정의하는 진리값
value	오퍼랜드 객체 안에 저장되어 있는 데이터

- 연관

process	오퍼랜드 객체가 연관되어진 프로세스의 id
---------	-------------------------

(3) 상태 객체

상태 객체는 프로세스가 어떻게 진행되고 있는 지에 대한 정보뿐만 아니라 프로세스 객체의 조건을 트리거하거나 정책 객체의 실행에 원인을 제공하는데 사용되는 아주 단순한 객체이다. 상태 객체들은 다양한 목적으로 사용될 수 있다. 상태 객체는 다음과 같은 속성과 연관을 가진다.

- 속성

ID	OrchestraXAServer 시스템 내부의 객체를 유일하게 확인하는 문자열
label	프로세스를 확인하는데 id 대신 사용될 수 있는 짧고, 사람이 읽어 그 의미를 알 수 있는 문자열. 프로세스와 관련 있는 모든 객체들은 유일한 레이블을 가져야만 한다.
description	프로세스를 기술하는 텍스트. OrchestraXAServer 클라이언트에 의해 사용된다.
ACL(set of ACEs)	OrchestraXAServer 객체와 시스템 사용자 사이의 관계를 설정한다.
visible in subtree (true, false)	오퍼랜드 객체의 영역을 정의하는 진리값
value	오퍼랜드 객체 안에 저장되어 있는 데이터

- 연관

process	오퍼랜드 객체가 연관되어진 프로세스의 id
---------	-------------------------

(4) 정책 객체

정책 객체는 모델링 되어지는 프로세스에 관련된 임의의 로직을 캡슐화한다. 이 객체를 통해 프로세스는 커스터마이징되며 극도로 유연해진다. 정책 객체는 Java 클래스와 이 실행 파일을 트리거하는 이벤트를 연관 짓는다. 정책 객체는 다음의 속성들과 연관들을 가진다.

- 속성

ID	OrchestraXAServer 시스템 내부의 객체를 유일하게 확인하는 문자열
label	프로세스를 확인하는데 id 대신 사용될 수 있는 짧고, 사람이 읽어 그 의미를 알 수 있는 문자열. 프로세스와 관련 있는 모든 객체들은 유일한 레이블을 가져야만 한다.
description	프로세스를 기술하는 텍스트. OrchestraXAServer 클라이언트에 의해 사용된다.
ACL(set of ACEs)	OrchestraXAServer 객체와 시스템 사용자 사이의 관계를 설정한다.
event source	정책 객체의 실행을 위해 OrchestraXAEvent 객체가 가져야만 하는 이벤트 소스
event name	정책 객체의 실행을 위해 OrchestraXAEvent 객체가 가져야만 하는 이벤트 이름 (null 값은 어떠한 유형의 OrchestraXAEvent 객체도 정책 객체를 실행할 수 있음을 가리킨다)
event attributes	정책 객체의 실행을 위해 OrchestraXAEvent 객체가 가져야만 하는 이벤트 속성
language	정책 객체에 사용된 언어
source code type (CODE, CLASSNAME, URL)	문자열 sourceCode 속성이 정책 객체의 실행가능한 코드를 위한 클래스 이름, url 또는 원시 소스를 포함하고 있는 지를 알아보는데 사용되는 속성

- 연관

process	오퍼랜드 객체가 연관되어진 프로세스의 id
---------	-------------------------

각각의 정책 객체들은 실행 가능한 Java 클래스와 BSF (Bean Script Framework) 스크립트 두 가지 유형 중 하나로 구성된다. 정책 객체의 "source code type" 속성은 정책 객체의 유형과 실행할 수 있는 명령어에 대한 접근법을 가리킨다.

정책 객체의 실행은 이벤트의 발생으로 트리거된다. 이벤트 소스는 크게 세 가지로 구분되는데 프로세스 변경, 상태의 변경 그리고 프로세스 타이머가 있다. 외부 시스템 또한 이벤트를 통지할 수 있다. 각각의 이벤트는 이벤트 유형, 이벤트 소스, 이벤트 속성 테이블, 그리고 이벤트의 영역을 결정하는 프로세스 ID 로 구성된다. 영역을 결정하는 프로세스 ID가 포함되면 명시된 프로세스 또는 그 상위의 프로세스와 관련된 정책 객체 또는 루트 노드에 있는 정책 객체가 이벤트에 의해서 트리거될 수 있다. 테이블 내 속성들은 이벤트 유형에 특유의 성질을 갖는다. 프로세스 변경 이벤트 소스는 비즈니스 프로세스 실행 엔진인 OrchestraXAServer의 내부 서비스인 "ProcessCommandService" 이다. 프로세스 명령 서비스에 의해 전송되는 서로 다른 이벤트 이름은 ProcessStarted, ProcessFinished, StatusAddition, StatusRemoved 등이 있다.

첫째, ProcessStarted 이벤트는 ProcessID (시작된 프로세스 ID), ProcessLabelPath (시작된 프로세스의 레이블 경로)와 같은 속성들을 가진다. 둘째, ProcessFinished 이벤트는 ProcessID (중단된 프로세스 ID), ProcessLabelPath (중단된 프로세스의 레이블 경로)와 같은 속성들을 가진다. 셋째, StatusAddition 이벤트는 ProcessID (프로세스 ID), ProcessLabelPath (프로세스의 레이블 경로), StatusID (상태 ID), StatusLabelID (상태의 레이블 경로)와 같은 속성들을 가진다. 넷째, StatusRemoval 이벤트는 ProcessID (시작된 프로세스 ID), ProcessLabelPath (프로세스의 레이블 경로), StatusID (상태 ID), StatusLabelID (상태의 레이블 경로)와 같은 속성들을 가진다.

타이머 이벤트 소스는 비즈니스 프로세스 실행 엔진인 OrchestraXAServer의 내부 서비스인 "TimerService" 이다. 모든 타이머 이벤트는 "TimerExpired"라는 이름을 가진다. 속성들은 타이머를 생성하는 프로세스 작성자에 의해 명시되지만 모든 타이머는 자신을 포함하는 프로세스를 확인하는 "ProcessID" 이름의 속성을 가진다.

OrchestraXAServer는 정책 코드가 Java, JavaScript, Tcl, 그리고 Python에 국한되지 않고 다양한 언어로 작성될 수 있도록 지원하며 다음과 같은 특징을 갖는다.

- 프리컴파일된(pre-compiled) Java 클래스 사용

java.lang.Runnable 인터페이스 또는 com.ateamsoft.orchestraxa.PolicyScript 인터페이스를 구현하는 어떠한 Java 클래스도 정책 객체에 의해서 호출될 수 있다. Navigator 객체는 정책 객체로부터 OrchestraXAServer 객체에 접근하는데 사용된다.

- Bean Script Framework 사용

BSF는 다른 프로그래밍 언어가 Navigator 객체를 접근하도록 허용한다. Navigator 클래스는 정책 객체로부터 Orchestra-

XAServer 객체를 접근하는데 유용하다. Java 정책 객체로부터 Navigator 객체를 얻기 위해 다음의 코드를 사용한다.

```
Navigator navigator = (Navigator) bsf.lookupBean("navigator");
```

- 예외 처리

정책 객체는 비동기적으로 수행되어지고 서버에 대한 특별한 요청과 필연적으로 관련이 없으므로, 정책 객체의 수행 도중 예외가 발생하면, OrchestraXAServer는 클라이언트에 대한 책임으로 에러를 보고할 필요는 없다. 결과적으로 정책 객체의 실행 도중 예외가 발생하면 OrchestraXAServer는 단순히 서버 로그에 에러를 기록한다.

- 재사용 가능한 정책 객체

정책 객체를 재사용하게 하는 첫 번째 단계는 데이터들 스크립트에 하드 코딩하지 않고 오퍼랜드 객체에 두는 것이다. OrchestraXAServer는 새로운 정책 템플릿을 추가하는 메커니즘을 제공한다. 정책 템플릿("매개변수화된 정책"이라고도 불린다)은 xml/policies 디렉토리 내에 XML 파일을 생성함으로써 OrchestraXAServer에서 사용 가능해진다.

4.2.2 OrchestraXAServer 클라이언트 구현

비즈니스 프로세스 실행 엔진 OrchestraXAServer와 통신할 수 있는 인터페이스는 EJB 세션 빈 메소드, HTTP를 통한 SOAP 메시지 전송, HTTP를 통한 OrchestraXAServer XML 전송 등 세 가지이다. 내부 작동 메커니즘과는 관계없이 모든 인터페이스에 의해 사용되는 기본 개념은 동일하다. 애플리케이션은 특정 유형의 요청 생성, 인증 토큰, 매개변수 추가, 그리고 OrchestraXAServer로 요청을 전송한다. 이에 대한 응답으로 OrchestraXAServer는 성공 또는 실패를 가리키는 수치 코드와 상응하는 텍스트 메시지를 회신한다. 인증 토큰은 로그인 요청시 적절한 신용평가를 제공하여 세션 초기에 획득된다.

(1) 인증

로그인 요청시를 제외하고 각각의 요청은 유효한 인증 토큰을 요구한다. 이 인증 토큰은 유효한 사용자 이름과 비밀번호와 함께 로그인 요청 전송 시 획득된다. 로그인에 성공하면 차후의 요청에 사용되는 인증 토큰을 포함하는 응답이 생성된다. 토큰은 서비스를 요청하는 사람 또는 개체를 확인하는데 사용된다.

(2) OrchestraXAServer 클라이언트 생성

com.ateamsoft.orchestraxa.client.OrchestraXAProxy 인터페이스는 OrchestraXAServer와 상호작용을 위한 메소드를 제공한다. com.ateamsoft.orchestraxa.client 패키지는 OrchestraXAProxy 인터페이스를 구현한 두 개의 클래스 XMLOrchestraxaProxy 와 EJBOrchestraxaProxy를 제공한다. XMLOrchestraxaProxy 클래스는 OrchestraXAServer와 통신하기 위해 XML/HTTP를 사용한다. EJBOrchestraxaProxy 클래스는 같은 기능을 제공하기 위해서 EJB 무상태

세션 빈과 통신한다. 대부분의 경우에 있어 EJBOrchestraxa-Proxy가 보다 나은 성능을 보이지만, XMLOrchestraxaProxy는 방화벽을 통하여 통신할 수 있는 이점을 제공한다.

(3) 업무 포털 웹 애플리케이션 생성

OrchestraXA 포털 프레임워크는 OrchestraXAServer와 통신하는 웹 애플리케이션을 구축하는데 기본 설비를 제공하는 패턴, 관례, 그리고 Java 클래스들의 집합이다. OrchestraXA 포털 프레임워크로 구축된 웹 애플리케이션은 JSP 페이지, Java 서블릿, 그리고 web.xml 환경설정 파일로 구성된다. JSP 페이지는 어떠한 Java 코드도 포함하지 않으므로 모든 동작은 서블릿에서 일어난다. 각각의 서블릿은 "action" 객체에 일을 위임한다. 서블릿은 "action" 객체가 OrchestraXAServer와 통신하도록 OrchestraXAProxy 객체를 "action" 객체에 전달한다.

4.3 비즈니스 프로세스 모니터링 툴

비즈니스 프로세스 모니터링 툴은 비즈니스 프로세스 운용 콘솔을 제공한다. 이를 통해 프로세스 관리자는 비즈니스 단위와 개인들이 주어진 활동과 업무를 어떻게 수행하고 있는지, 마감시한이 촉박한지, 회사의 전략을 위협하고 있는 잠재적 요인들은 무엇인지를 추적한다. 웹 기반 애플리케이션인 모니터링 툴은 프로세스 관리자로 하여금 인터넷으로 연결된 어느 곳에서든지 실행중인 업무 아이템에 대해서 경영 및 교정을 수행하게 한다. 측정된 통계 정보는 최적화된 업무 분석과 성능 향상을 위해 비즈니스 프로세스 모델링 툴로 다시 피드백 된다. 이러한 자료들은 일상적인 기업의 비즈니스 수행에 대한 현실적인 이해를 유지하는데 필수적이다. 실시간 비즈니스 최적화를 통해 자원의 부적절한 할당, 업무 지연, 프로세스 병목현상을 방지할 수 있다. 비즈니스 프로세스 모니터링 툴의 주요 특징으로는 다음과 같다.

(1) 동적인 비즈니스 정보의 가시화

비즈니스 프로세스가 어떻게 협업 목표를 달성하는 지를 설명하기 위해 각각의 업무와 사용자 레벨의 부하 정도 및 상태를 포함한 업무 아이템의 실시간 뷰를 제공한다.

(2) 연속적인 비즈니스 프로세스 향상

일정하게 실시간 업무 상태 및 분석 자료를 프로세스 모델링 툴에 피드백 시킴으로써 지속적인 비즈니스 프로세스 향상을 가능하게 한다.

(3) 비즈니스 요구에 적절한 대응

일시적인 업무 부하의 불균형 및 프로세스 병목현상 발생 시 워크플로우 프로세스를 중단, 조정 및 재설정함으로써 비즈니스 프로세스가 원활하게 진행하도록 한다.

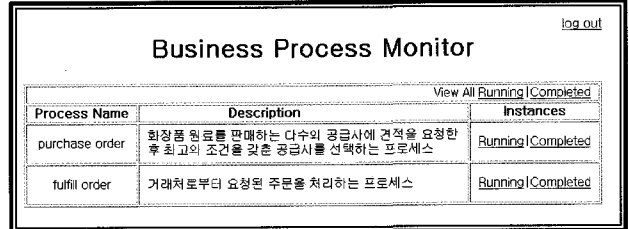
(4) 직관적인 인터페이스

기업의 비즈니스 프로세스가 전체적으로 어떻게 조직되어

져 있으며 각각의 프로세스 및 하위 프로세스 상태 그리고 액티비티의 상태를 직관적으로 파악할 수 있게 하였다.

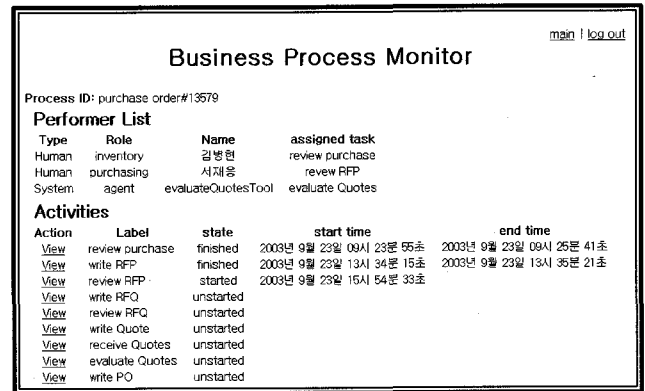
(5) 유용한 의사결정 정보 제공

추정된 통계가 아닌 실시간 기반의 비즈니스 프로세스 분석 결과를 보여준다.



(그림 9) 최상위 프로세스 레벨 관리자 인터페이스

(그림 9)는 최상위 프로세스 레벨 관리자 인터페이스를 보여주고 있다. 그림에서 "Process Name" 필드는 비즈니스 프로세스를 구성하는 최상위 레벨의 프로세스 이름이다. 각각의 프로세스는 여러 개의 하위 프로세스로 구성되어 있다. "Description" 필드는 프로세스의 주요 내용을 간략히 서술한다. "Instances" 필드는 각각의 비즈니스 프로세스 인스턴스에 대하여 실행중이거나 완료된 프로세스 목록을 출력한다. 마지막으로 "View All Running | Completed" 필드는 정해진 일정 기간 동안 비즈니스 프로세스 인스턴스로서 실행중이거나 완료된 모든 프로세스 리스트를 출력한다. (그림 10)은 개별 프로세스 레벨 관리자 인터페이스를 나타낸다.



(그림 10) 개별 프로세스 레벨 관리자 인터페이스

- Process ID: 프로세스 인스턴스 인식 스트링
- Performer List: 비즈니스 프로세스 실행에 관련된 수행자 목록

속 성	설 명
Type	액티비티를 수행하는 개체가 사람인지 자동화된 시스템인지를 나타낸다. 사람의 경우 "Human", 자동화된 시스템의 경우 "System"으로 표시한다.
Role	해당 업무를 수행하는 개체의 역할을 표시한다.
Name	해당 업무를 수행하는 개체들의 이름을 표시한다.
assigned task	해당 액티비티에 할당된 업무를 나타낸다.

- **Activities:** 비즈니스 프로세스를 구성하는 액티비티들의 집합.

속 성	설 명
Action	해당 액티비티의 상세한 내용을 출력
Label	해당 액티비티의 명칭
State	해당 액티비티의 상태. unstarted. started. finished
Start time	액티비티 인스턴스의 시작 시간.
End time	액티비티의 인스턴스의 종료 시간.

5. 도구의 비교 분석

항목 \ 도구	ADOME-WFMS	WIDE
모델링 방법	실행관점의 모델링 방법 제시	모델링 방법 제시하지 못함
모델링 도구	프로세스 정의 언어를 지원하나 모델링 도구 미비	프로세스 정의 언어를 지원하나 모델링 도구 미비
워크플로우 엔진 적용 여부	미적용	설계 측면에 부분적용
모니터링 도구	오프라인상의 지원	오프라인상의 지원

Jablonski	본 도구
변화 관점의 모델링 방법 제시	워크플로우 특성을 지원하는 복합 모델링 방법 제시
프로세스 정의 언어 지원하나 모델링 도구 미비	프로세스 참여자, 정보시스템, 비즈니스 프로세스간의 관계를 정의할 수 있는 도구 구현
미적용	XPDL을 지원하는 엔진 구현
오프라인상의 지원	웹 기반의 모니터링 도구 구현됨

6. 결 론

본 논문은 고객의 관점에서 BPM 실현을 위해 필요한 기능들을 프로세스 관리 사이클별로 구현하였다. 결과물은 비즈니스 프로세스 모델링 도구인 OrchestraXASudio, 비즈니스 프로세스 실행 엔진인 OrchestraXAServer, 그리고 비즈니스 프로세스 모니터링 툴인 OrchestraXAMonitor의 컴포넌트로 구성되어 있다. 먼저 비즈니스 프로세스 모델링 도구인 OrchestraXASudio는 BPM의 기반을 이루는 프로세스 매핑 도구로, 기업의 프로세스를 문서가 아닌 데이터베이스화를 통한 매핑 도식화 표준화 및 웹에서의 공유를 지원한다. OrchestraXASudio는 전략 자원 프로세스간 상관관계를 명확하게 표현하고 도식화하며, 프로세스 중심의 업무관리와 지속적인 변화관리를 제공한다. 비즈니스 프로세스 실행 엔진인 OrchestraXAServer는 표준화된 프로세스를 실행시키는 업무 프로세스 자동화 도구로, 워크플로우 또는 프로

세스 통합 애플리케이션의 신속한 배치와 유연한 실행을 위한 J2EE 기반의 프로세스 엔진이다. 멀티 플랫폼이면서 애플리케이션 서버에 독립적인 J2EE 기반 프로세스 엔진은 본 논문의 핵심이다. OrchestraXAServer는 모든 알려진 워크플로우와 프로세스 모델링 패턴뿐만 아니라 임의로 복잡한 비즈니스 또는 애플리케이션 프로세스 로직을 다룰 수 있다. 비즈니스 프로세스 모니터링 도구인 OrchestraXAMonitor는 실행되고 있는 프로세스의 사이클-타임은 물론, 프로세스와 핵심성과지표(KPI: Key Performance Indicator)등을 모니터링하여 제어할 수 있다. 특히 사이클-타임 시뮬레이터 등을 활용해 사전에 프로세스 개선 효과를 검증할 수 있어 개선 시나리오에 따른 'What-If' 분석 기능을 제공하는 동시에, BPM은 물론 BPR 프로세스 개선 활동을 극대화할 수 있다.

참 고 문 헌

- [1] George L. Kovács and Paolo Paganelli, "A planning and management infrastructure for large, complex, distributed projects—beyond ERP and SCM", Computers in Industry, Vol.51, Issue 2, pp.165-183, June, 2003.
- [2] E. Cream, Web Services Essentials, O'Reilly, First Edition, 2002.
- [3] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing", IEEE Computer, January, 2004.
- [4] Workflow Management Coalition, "The Workflow Reference Model", Workflow Management Coalition, January, 1995.
- [5] G. A. Bolcer, G. Kaiser, "SWAP: Leveraging the Web to Manage Workflow", IEEE Internet Computing, Vol.3, No.1, January-February, 1999.
- [6] Workflow Management Coalition, "Workflow Standard-Interoperability Internet Wf-XML Binding", Workflow Management Coalition, July, 1998.
- [7] D. Barbara, S. Mehrota, and M. Ruinkiewicz, "INCAs: Managing Dynamic Workflows in Distributed Environments", Journal of Database Management, Vol.7, No.1, pp.515, 1996.
- [8] BEA, IBM, and Microsoft, "BPEL4WS Version1.0", August 2002, available at <http://www.106.ibm.com/developerwork/webservices/library/ws-bpel>
- [9] Chang-Mog Lee, "Design and Implementation of B2Bi Collaboration Workflow Tool Based on J2EE, Computational Science and Its Applications - ICCSA 2006: International Conference, Glasgow, UK, May 8-11, Proceedings, Part IV, pp.1012 - 1021, 2006.

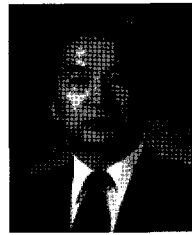


이 창 목

e-mail : cmlee@chonbuk.ac.kr
1999년 호원대학교 전자계산학과(학사)
2001년 전북대학교 전산통계학과(이학석사)
2005년 전북대학교 컴퓨터통계정보학과
(이학박사)
2005년~현재 전북대학교 영상정보신기술
연구센터 객원연구원

2006년~현재 전북대학교 전자정보 2단계 BK21사업단
전통문화영상관광사업팀 Post-Doc.

관심분야: CBSD, AOSD, BPM, 역공학 방법론, 소프트웨어
품질평가 등



장 옥 배

e-mail : okjang@chonbuk.ac.kr
1973년 고려대학교(학사, 석사)
1988년 산타바바라대 대학원(Ph.D.)
1974년~1980년 조지아 주립대, 오하이오
주립대 박사과정 수료
1980년~현재 전북대학교 전자정보공학부
교수

관심분야: 소프트웨어공학, 전산교육, 수치해석, 인공지능 등