

초대형 워크플로우 관리 시스템의 설계 및 구현

Design and Implementation of a Very Large-Scale Workflow Management System

안 형 진* 김 광 훈**
Hyung-Jin Ahn Kwang-Hoon Kim

요 약

최근 기업들은 조직 내외의 협업 또는 협력에 의해 발생하는 거대량의 업무들을 안정적으로 처리해줄 수 있는 초대형 워크플로우 관리 시스템의 등장을 요구하고 있다. 워크플로우 벤더들은 초대형 워크플로우 서비스를 지원하는 워크플로우 관리 시스템을 구현하기 위해, 하드웨어들의 추가적 확장을 통한 워크플로우 관리 시스템의 성능 향상을 모색하고 있다. 그러나 워크플로우 관리 시스템의 소프트웨어적인 아키텍처를 고려하지 않은 단순한 물리적 확장은 시스템 구축에 소요되는 시간적 또는 비용적 낭비를 야기할 수 있다. 이러한 한계를 개선하기 위하여, 본 논문에서는 워크케이스 기반 워크플로우 아키텍처를 제안하고, 해당 아키텍처를 토대로 하는 초대형 워크플로우 관리 시스템을 구현한다. 또한 워크케이스 기반 초대형 워크플로우 관리 시스템에 관한 생성 반응 시간 관련 실험을 통해서, 논리적인 소프트웨어 아키텍처의 개선이 워크플로우 관리 시스템의 초대형성에 매우 큰 영향을 미칠 수 있음을 증명하고자 한다.

ABSTRACT

Recently, many organizations such as companies or institutions have demanded induction of very large-scale workflow management system in order to process a large number of business-instances. Workflow vendors have focused on physical extension of workflow engines based on device-level clustering, so as to provide very large-scale workflow services. Performance improvement of workflow engine by simple physical-connection among computer systems which don't consider logical-level software architecture leads to wastes of time or cost for construction of very large-scale workflow service environment. In this paper, we propose workcase-based workflow architecture and implement a very large-scale workflow management system based on the architecture. We prove that software architectures to be applied on a workflow engine have an effect on scalability and performance through workcase response-time evaluation of our proposed system.

☞ KeyWords : 워크케이스 기반 워크플로우 아키텍처(Workcase-based Workflow Architecture), 초대형 워크플로우 관리 시스템(Very Large-Scale Workflow Management System)

1. 서 론

이전의 워크플로우 관리 시스템들은 문서 결제, 출장 신청 등 단위 조직 내에서 빈번하게 발생하는 소규모의 내부 비즈니스 프로세스들에 대한 자동화를 제공하는데 이용되어 왔다[1-3]. 최근 여러 기업들 사이의 협업 또는 협력에 의해 비즈

니스 프로세스의 절차가 더욱 복잡해지고 조직 규모가 점차적으로 증가함에 따라, 다수의 클라이언트들이 요구하는 거대량의 작업 요청들을 안정적으로 처리할 수 있는 초대형 워크플로우 관리 시스템의 등장을 요구하고 있다[4,5]. 초대형 워크플로우 관리 시스템[6]은 다수의 기업 및 조직 구성원들이 참여에 의해 발생하는 대량의 워크플로우 인스턴스들을 안정적으로 처리하도록 지원하는 시스템을 말한다. 초대형 워크플로우 관리 시스템은 많은 수의 워크플로우 인스턴스들을 효율적으로 관리할 수 있도록 지원하는 확장성(Scalability), 워크플로우 관리 시스템의 가용 공간

* 정 회 원 : KT 중앙연구소 선임연구원
hjahn07@kt.com

** 종신회원 : 경기대학교 컴퓨터과학과 교수
kwangs@kgu.ac.kr

[2009/03/03 투고 - 2009/03/05 심사 - 2009/05/04 심사완료]

내에 상주하는 워크플로우 인스턴스들에 관한 안정적인 처리를 지원하는 신뢰성(Reliability) 및 워크플로우 인스턴스들을 처리하는 과정에서 일어나는 예외 상황들에 대한 복구를 통해 워크플로우의 문맥적 연속성을 유지할 수 있도록 지원하는 가용성(Availability) 등을 보장할 수 있어야 한다. 이 중에서 특히 확장성의 보장은 초대형 워크플로우 관리 기술 분야의 가장 주요한 이슈로 다루어지고 있다[6,7].

현재 대부분의 워크플로우 시스템 벤더들은 자사 보유 시스템의 확장성을 증가시키기 위하여, 워크플로우 관리 시스템을 구성하는 하드웨어들의 추가 확장을 통해 성능 향상을 모색하고 있다. 그러나 워크플로우 관리 시스템의 소프트웨어적인 아키텍처를 고려하지 않은 단순한 물리적 확장은 추가된 장치들에 비례한 성능을 보장받기 어려우며, 시스템 구축에 소요되는 시간 또는 비용의 낭비를 야기할 수 있다. 기존 워크플로우 관리 시스템의 구조적인 한계 및 구현 방법의 비효율성을 개선하기 위하여, 본 논문에서는 워크케이스 기반 워크플로우 아키텍처를 제안하고 해당 아키텍처를 토대로 하는 워크플로우 관리 시스템을 구현하고자 한다. 또한 본 논문에서 제안하는 워크케이스 아키텍처가 초대형 워크플로우 서비스를 제공하는데 적합한 구조임을 입증하기 위하여, 우리는 현재 대부분의 워크플로우 시스템 벤더들이 사실 표준으로 채택하고 있는 액티비티 인스턴스 기반 아키텍처와 생성 반응 시간 관련 비교 실험을 수행한다. 본 실험의 결과로부터, 우리는 논리적인 워크플로우 아키텍처의 개선이 워크플로우 관리 시스템의 초대형성 향상에 매우 큰 영향을 미칠 수 있음을 증명하고자 한다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구로써 워크플로우 관리 시스템의 성능 개선 및 확장을 위한 이전 연구들에 대해 살펴보고 그 방법들이 가진 한계점들을 기술한다. 3장에서는 초대형 워크플로우 서비스에 대한 개념적인 정의 및 목표에 대해 구체적으로 설명한다. 4장에서는

기존의 전형적인 액티비티 인스턴스 기반 아키텍처가 가진 문제를 기술하고, 본 논문에서 목표로 하는 초대형 워크플로우 서비스 제공에 적합하도록 개선한 워크케이스 기반 워크플로우 아키텍처를 제안한다. 5장에서는 본 논문에서 제안하는 워크케이스 아키텍처 기반의 워크플로우 관리 시스템을 설계 및 구현한다. 6장에서는 워크케이스 아키텍처가 초대형 워크플로우 서비스를 제공하는데 적합한 아키텍처임을 증명하기 위한 생성 반응 시간 관련 비교 실험을 수행한다. 마지막으로 7장에서는 본 논문의 결론 및 향후 연구 과제를 기술한다.

2. 관련 연구

초장기의 워크플로우 관리 시스템들은 오피스 환경에서의 문서 전달 또는 업무 전이의 자동화를 제어 관리하는 역할을 담당하였다. 초기의 OfficeTalk-D[8,9]와 같은 워크플로우 관리 시스템들은 임의의 비즈니스 목표를 달성하는데 필요한 업무들의 순차적, 병행적 또는 선택적 절차들에 대한 자동화를 제공함으로써, 조직에서 일어나는 다양한 워크플로우 프로세스들의 처리 소요 시간 단축 및 비용 절감 효과를 실현해주었다. 이에 따라 기업들은 점차적으로 단순한 문서 및 작업에 필요한 파일들의 자동 전이가 아닌, 높은 복잡도를 가지는 비즈니스 프로세스들에 대한 정확한 처리를 제공해줄 수 있는 워크플로우 관리 시스템을 요구하게 되었다. 트랜잭션 지향적 워크플로우 관리 시스템(Transactional workflow management system)[10]은 하나의 워크플로우 인스턴스 수행에 관한 전체 서비스 트랜잭션의 무결성을 보장하여 비즈니스 프로세스 자동화의 오류를 최소화하기 위한 시스템을 말한다. Stuart et al.[11]은 CORBA를 이용하여 워크플로우 서비스 제공 시 발생하는 동적 예외 상황들에 대한 방지 및 복구를 지원하는 트랜잭션 워크플로우 관리 시스템을 제안하였다. Choi et al.[12]은 다양한 플랫폼 운영

환경 상에서 트랜잭션 워크플로우 서비스 제공이 가능하도록 하기 위하여, 페트리-넷(Petri-net)[13]을 이용한 워크플로우 모델링 방법을 제안하였다. 또한 K. Guntzel[14]는 웹 서비스 환경 상에서 동작하는 워크플로우 프로세스 인스턴스의 트랜잭션 무결성을 보장하기 위한 프로세스 모델링 방법 및 아키텍처를 제안하였으며, 이와 유사하게 Olaf et al.[15]은 SOA(Service Oriented Architecture) 환경에서의 워크플로우 서비스를 수행하는 과정에서 시간 제약적 트랜잭션 진행과 이로부터 발생하는 예외 상황에 대한 보상을 지원하는 트랜잭션 워크플로우 관리 방법에 대해 제시하였다.

최근 기업 및 조직들은 네트워크 인프라의 급속한 발달로 인해 여러 조직들 간의 협업 또는 협력이 빈번해지고 그에 따른 비즈니스 프로세스 규모 증가에 따라, 다수의 업무 수행자 및 리소스들의 개입을 통해 발생하는 대량의 작업들을 안정적으로 처리해줄 수 있는 초대형 워크플로우 관리 시스템(Very large-scale workflow management system)[4,6,7]의 등장을 요구하고 있다. 현재 BizFlow, InConcert, Staffware, FloWare, Notes, ActionWKF, FlowWorks, FlowMark와 같은 다수의 워크플로우 벤더 제품들이 대규모 워크플로우 관련 인스턴스들의 처리를 위한 초대형 워크플로우 서비스를 고려하고 있다. 그러나 대다수의 제품들은 워크플로우 관리 시스템의 사실 표준 아키텍처인 OMG의 JointFlow[16]가 근간으로 하는 액티비티 인스턴스 기반 워크플로우 아키텍처를 채택하고 있다. 또한 일부 벤더에서는 시스템 구성 장치들의 추가적 연결 및 확장을 통해 성능 향상을 모색하는 하드웨어적 중심적인 초대형 워크플로우 환경 구축을 지향하고 있다. 그러나 워크플로우 관리 시스템의 소프트웨어적인 아키텍처를 고려하지 않은 단순 하드웨어 확장만으로는 추가 대비 성능을 보장받기 어려우며, 결과적으로 시스템 환경 구축에 소요되는 시간 및 비용적 낭비를 야기할 수 있다.

3. 초대형 워크플로우 서비스 개요

기존의 워크플로우 관리 시스템들은 일반적으로 단일 조직이 운영하는 비즈니스 프로세스들에 대하여, 해당 조직 내에서 관리하는 리소스들이 자동적으로 처리할 수 있도록 제어 관리하는 인트라 조직 환경에서의 워크플로우 서비스(Intra-organizational Workflow Service)를 중점적으로 제공해왔다[17,18]. 그러나 최근에 기업들은 기업들 양자간 또는 다자간의 협력(Inter-organizational Workflow[19-22] 또는 Cross-organizational Workflow[23-27]) 및 다수 구성원들이 참여하는 협업(Collaborative Workflow)[28,29]등에 의한 조직 규모의 증가, 프로세스 구성 업무들의 복잡도 증가 및 리소스들에 할당되는 작업 부하율의 증가 등에 따라 발생하는 대량의 워크플로우 관련 인스턴스들을 처리할 수 있는 초대형 워크플로우 관리 기술의 일반화 및 해당 기술의 확산을 요구하고 있다. 임의의 워크플로우 관리 시스템이 거대량의 워크플로우 인스턴스들을 원활하게 제어 및 관리하는 것이 가능하기 위해서는 해당 런타임 인스턴스들을 충분하게 용적할 수 있는 확장성이 우선적으로 보장되어야 하며, 이상적인 워크플로우 확장성을 보장하는 초대형 워크플로우 서비스를 제공하기 위해서는 (그림 1)에서 보는 바와 같이 워크플로우 복잡도를 결정하는 세 가지 관점에서의 초대형성을 모두 고려할 수 있어야 한다[33,34].

- 워크플로우 리소스 관점

- (Workflow Engagement Perspective)

워크플로우 관리 시스템은 역할, 팀, 부서 또는 조직 등에 소속된 개인 수행자 또는 시스템과 같은 리소스 컴포넌트들이 임의의 워크플로우 프로세스 내 단위 업무들을 실시간으로 처리할 수 있도록 조정하는 역할을 수행한다. 워크플로우 리소스들은 워크플로우 런타임 클라이언트(Workflow Runtime Client) 도구 또는 실시간 자동 응용 호출

(Automated Workflow-enabled Application Invocation) 메커니즘을 통해 자신에게 할당된 워크아이템들을 수행한다. 하나의 워크플로우 리소스가 자신의 한 워크아이템을 처리하기 위해 워크플로우 관리 시스템에게 이벤트 메시지들을 전송하는 시점에, 워크플로우 관리 시스템은 해당 워크아이템의 처리를 위해 워크플로우 인스턴스를 생성 또는 활성화해야 한다. 이러한 사실로부터 워크플로우 리소스들이 프로세스 업무 처리에 개입할 시 워크플로우 관리 시스템 내에는 그에 따른 처리를 위한 워크플로우 관련 인스턴스들이 발생한다는 점을 알 수 있으며, 결과적으로 워크플로우 리소스가 워크플로우 관리 시스템의 확장성에 직접적으로 관여한다는 점을 알 수 있다.

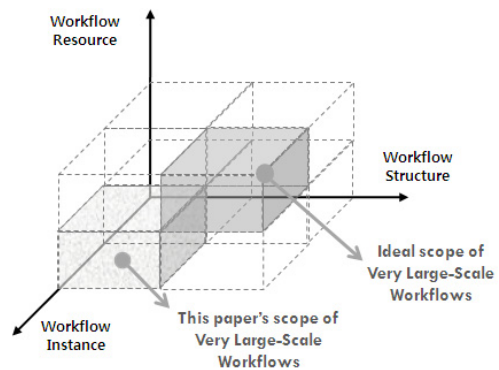
- 워크플로우 프로세스 구조적 관점
(Workflow Structure Perspective)

최근의 기업들은 해당 조직 내에서 자체적으로 관리하는 지역적 형태의 비즈니스 프로세스들 뿐만 아니라, 다른 기업들과의 협력을 통한 글로벌 형태의 비즈니스 프로세스들을 동시에 운영해나가고 있다. 두 기업체 이상이 참여하는 다자간 협력 관계에 의해 형성된 비즈니스 프로세스들은 일반적으로 참여 기업들이 관리하는 여러 비즈니스 프로세스들의 결합에 의해 다수의 액티비티들이 존재하는 구조를 가진다. 워크플로우 관리 시스템이 높은 복잡도를 가지는 대규모 워크플로우 프로세스를 처리하기 위해서는 해당 워크플로우 프로세스로부터 발생하는 대량의 인스턴스들을 감내할 수 있는 확장성이 필수적으로 보장되어야 한다.

- 워크플로우 인스턴스 관점
(Workflow Instance Perspective)

초대형 워크플로우 서비스는 일반적으로 단위 워크플로우 관리 시스템 또는 하나 이상의 워크플로우 관리 시스템들로 구성된 워크플로우 수행 서비스(Workflow Enactment Service)들에 의해 최

소 수천 개에서 최대 수백만 개에 이르는 인스턴스들에 대한 제어 및 관리가 가능한 서비스를 의미한다. 임의의 워크플로우 관리 시스템 또는 워크플로우 수행 서비스가 초대형 워크플로우 서비스 도메인 상에 상주하는 거대량의 워크플로우 인스턴스들을 빠르고 안정적으로 처리할 수 있기 위해서는 시스템의 성능을 저해하지 않음과 동시에 현재의 활성 인스턴스 양에 적절한 가용 공간을 유연하게 확장하는 것이 가능해야 한다.

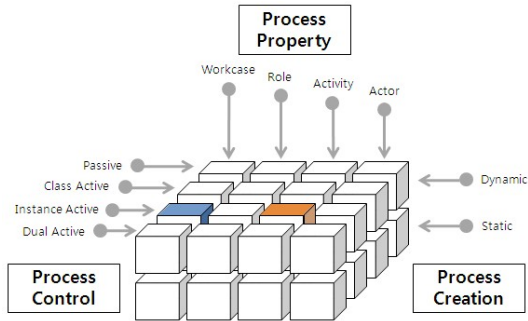


(그림 1) 워크플로우 복잡도 결정 요소 및 연관 관계

기존 대다수의 워크플로우 관리 시스템들은 워크플로우 인스턴스 처리 가용 공간에 대한 확장성 확보를 위해서 하드웨어 중심적인 시스템 성능 증가를 추구해왔다. 워크플로우 구현 벤더들은 하드웨어의 추가 증설을 통한 워크플로우 관리 시스템의 성능 확장 시에 전형적인 분산 환경에서 적용해왔던 시스템 컴포넌트 배치 방안들을 이용하고 있으나, 워크플로우 프로세스가 가진 특징들을 세부적으로 고려한 이상적인 분산 배치라 하기 어렵다. 본 논문에서는 워크플로우 도메인 특장적인 소프트웨어 아키텍처가 워크플로우 관리 시스템의 초대형성에 어느 정도의 영향을 미치는지에 대해 구현 및 실험을 통해 분석해보고자 한다.

4. 워크케이스 기반 워크플로우 아키텍처

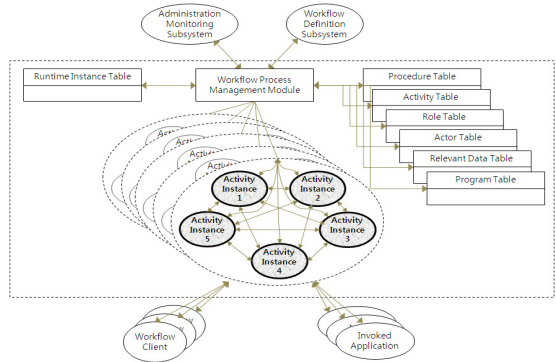
C. A. Ellis et al.[10]는 워크플로우 프로세스를 구성하는 업무 절차, 조직 구조 및 프로세스 수행 방법 등의 다양한 관점들을 고려한 개념적인 워크플로우 아키텍처 분류 프레임워크를 제안하였다. 본 논문에서는 워크플로우 프로세스의 전반적인 수행을 능동적으로 관리하는 프로세스 인스턴스인 워크케이스라는 리소스 개념을 추가하여 기존의 분류 프레임워크를 확장하고 있다. 워크플로우 관리 시스템을 구성하는 소프트웨어 아키텍처들은 워크플로우 프로세스의 수행을 주관하는 롤, 액터, 액티비티, 워크케이스 등의 리소스들의 관점을 고려하는 프로세스 속성(Process property) 항목과 워크플로우 관리 시스템 상에서 워크플로우 프로세스의 컨텍스트를 처리하는 인스턴스 구조를 고려하는 프로세스 제어(Process control) 항목 및 워크플로우 인스턴스 생성 방법에 관한 프로세스 생성(Process creation) 항목들을 통해 개념적인 분류가 가능하다.



(그림 2) 워크플로우 아키텍처의 개념적 분류 프레임워크

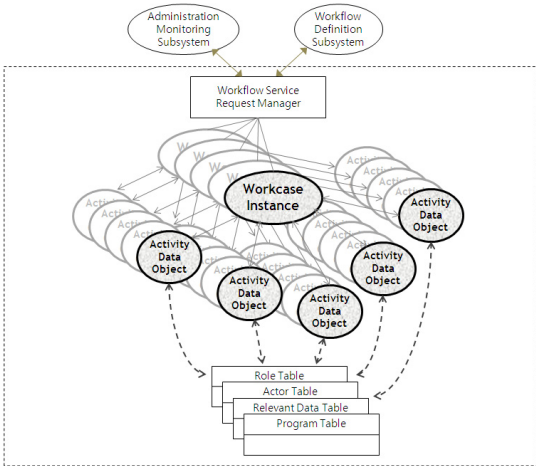
현재 대부분의 워크플로우 관리 시스템들은 객체 관리 기술 표준화 그룹인 OMG(Object Management Group)에서 명세하고 있는 표준 워크플로우 아키텍처인 JointFlow를 이용하고 있다. (그림 3)에서 보는 바와 같이, OMG의 JointFlow는 액티비티 인스턴스들이 사전 정의된 워크플로우

프로세스의 절차에 따라 주도적으로 업무를 진행해나가는 구조인 동적 인스턴스 액티브 유형의 액티비티 인스턴스 기반 워크플로우 아키텍처이다. 본 논문에서 제안하는 워크플로우 아키텍처는 워크플로우 관리 시스템에 의해 생성 후 구동되는 프로세스 인스턴스인 워크케이스가 업무 절차에 따라 동적으로 액티비티 데이터들을 참조하며 워크플로우 프로시저를 처리해나가는 동적 인스턴스 액티브 유형의 워크케이스 기반 워크플로우 아키텍처이다.



(그림 3) 동적 인스턴스 액티브 유형의 액티비티 인스턴스 기반 워크플로우 아키텍처

m개의 액티비티들로 구성된 하나의 워크플로우 프로세스에 대하여 업무 수행자들의 개입에 의해 n번의 워크케이스 생성 요청 이벤트가 발생한다고 가정할 때, (그림 3)과 (그림 4)에서 보는 바와 같이 OMG JointFlow와 같은 액티비티 인스턴스 기반 워크플로우 아키텍처를 사용하는 워크플로우 엔진은 $n * m$ 개의 워크플로우 인스턴스들을 처리해야 하는데 반해, 워크케이스 아키텍처 기반의 워크플로우 엔진은 n개의 워크케이스 인스턴스들을 처리할 수 있어야 한다. 따라서 워크케이스 기반 워크플로우 아키텍처는 액티비티 인스턴스 기반 워크플로우 아키텍처에 비해 워크플로우 인스턴스들의 처리량을 감소시키며, 이로부터 워크플로우 관리 시스템의 확장성을 증가시킬 수 있음을 예측할 수 있다.



(그림 4) 동적 인스턴스 액티브 유형의 워크케이스 기반 워크플로우 아키텍처

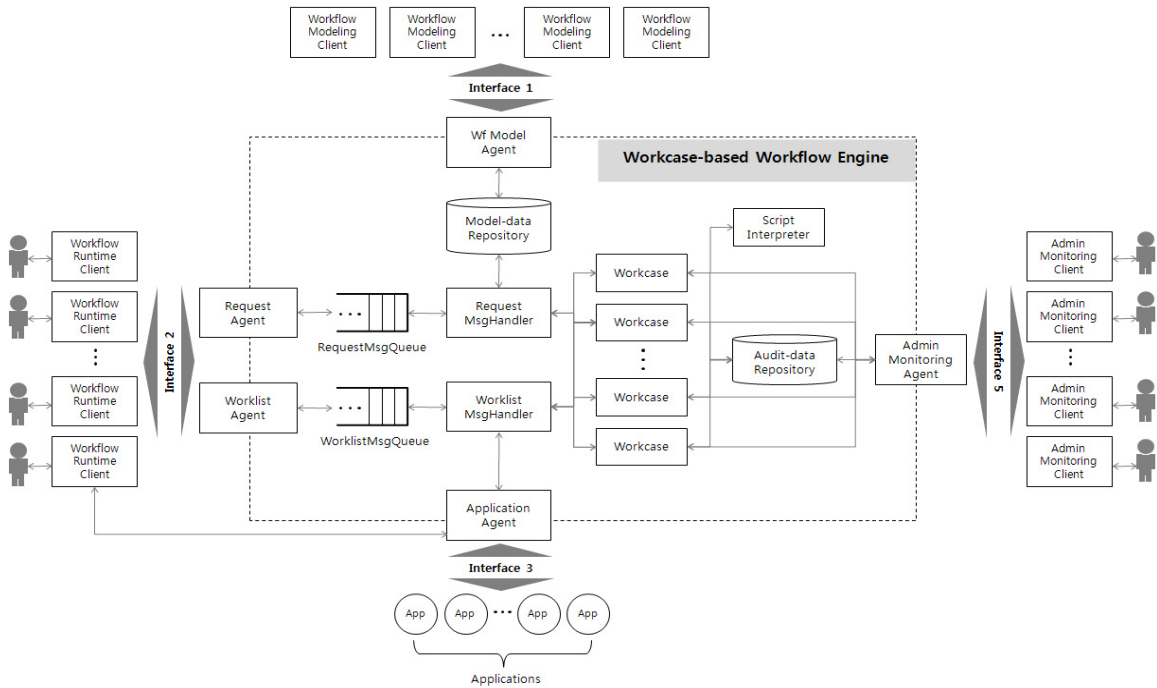
5. 워크케이스 기반 워크플로우 관리 시스템 설계 및 구현

이 장에서는 초대형 워크플로우 서비스를 제공하기 위하여 본 논문에서 제안하고 있는 워크케이스 아키텍처 기반의 워크플로우 관리 시스템을 설계 및 구현한다. 워크케이스 기반 워크플로우 관리 시스템은 액티비티 인스턴스에 의해 업무 프로시저 처리가 주도되는 기존의 액티비티 기반 워크플로우 관리 시스템과 달리, 워크플로우 프로세스 인스턴스인 워크케이스가 업무 절차에 따른 적절한 수행 시점에 액티비티 정의 템플릿들을 동적으로 데이터로써 활용하며 처리해나가는 메커니즘을 이용한다. 본 논문에서의 워크케이스 기반 워크플로우 관리 시스템은 전형적인 클라이언트-서버, 클러스터링 기반의 확장 클라이언트-서버 또는 분산 환경 상에서 다수의 클라이언트들이 발생시키는 작업 요청들에 대한 안정적인 처리를 지원하는 대표적인 서버-사이드 기술인 Java EE(Java Platform, Enterprise Edition)의 EJB(Enterprise Java Beans) 프레임워크[30]를 기반 기술로 사용하고 있다. 다음의 (그림 5)는 EJB 프레임워크를 이용한 워크케이스 기반 워크플로우

관리 시스템의 구현 레벨 아키텍처를 나타내고 있으며, 다음과 같이 기본적인 네 개의 인터페이스들을 제공한다. 여기서 인터페이스 4는 워크플로우 엔진 간 상호운용을 정의하는 인터페이스로써 본 논문의 범위에서는 제외하고 있다.

- 인터페이스 1 : 워크플로우 프로세스 정의 사용자와 워크플로우 모델링 도구 간 인터페이스
- 인터페이스 2 : 워크플로우 업무 수행자와 워크플로우 엔진 간 상호 작용 인터페이스
- 인터페이스 3 : 워크플로우 엔진의 응용 애플리케이션 호출 인터페이스
- 인터페이스 5 : 워크플로우 관리/모니터링 사용자와 워크플로우 엔진 간 상호 작용 인터페이스

워크케이스 기반 워크플로우 관리 시스템을 구성하는 서비스 컴포넌트들의 기본적인 워크플로우 프로세스 수행 절차는 다음과 같다. 워크플로우 프로세스 정의 권한을 가진 사용자는 워크플로우 모델링 클라이언트 도구(Workflow Modeling Client Tool)를 이용하여 특정 비즈니스 프로세스에 관한 업무 절차 및 각 업무들에 대한 정보를 기술한 후 디플로이를 수행한다. 정의가 완료된 워크플로우 프로세스는 디플로이 과정을 거쳐 워크플로우 모델 에이전트(Workflow Model Agent)에 의해 워크플로우 엔진이 해석 가능한 워크플로우 모델 데이터로 저장된다. 최초 프로세스의 시작 권한을 가진 임의의 사용자는 워크플로우 런타임 클라이언트 도구(Workflow Runtime Client Tool)을 통해 워크플로우 프로세스를 구동시키기 위한 워크케이스 생성 및 시작 요청 메시지를 전달한다. 워크플로우 서비스 요청 처리 에이전트(Request Agent)는 워크케이스 요청 수행자로부터 수신한 메시지를 요청 메시지 큐(Request Message Queue)에 적재하고, 워크플로우 서비스 요청 메시지 핸들러(Request Message Handler)는 전달된 요



(그림 5) 워크케이스 기반 워크플로우 관리 시스템 구현 아키텍처

청 메시지를 해석한 후 실제 워크케이스 인스턴스를 생성 및 구동한다. 워크케이스는 워크플로우 모델에 대한 인스턴스로써, 업무 절차 진행, 워크아이템 할당 및 워크플로우 관련 데이터(Workflow Relevant Data)들의 입출력 처리 등과 같은 전반적인 프로세스 제어 흐름을 주관하는 역할을 담당한다. 워크케이스는 관리하고 있는 업무 절차에 따라 적절한 액티비티에 대응하는 워크아이템을 생성한 후 해당 업무 수행자에게 할당한다. 임의의 업무 수행자는 워크플로우 런타임 도구를 이용하여 제한된 시간 내에 자신에게 할당된 워크아이템을 처리한 후, 워크리스트 에이전트(Worklist Agent)에게 업무 처리 종료 메시지를 전송한다. 워크리스트 에이전트는 수신한 메시지를 워크리스트 메시지 큐(Worklist Message Queue)에 적재하고, 적재 이벤트를 전달받은 워크리스트 메시지 핸들러(Worklist Message Handler)는 워크리스트 메시지 큐로부터 메시지를 가져와서 해석

한 후 워크케이스에게 해당 워크아이템에 상응하는 액티비티의 처리가 종료되었음을 통지한다. 워크케이스는 통보받은 액티비티 데이터를 메모리에서 해제하고 다음 순서의 액티비티 데이터를 활성화하여 업무 절차를 진행해나간다. 워크케이스가 업무 절차에 따라 액티비티, 워크아이템 및 관련 데이터들을 처리해나가며 발생하는 런타임 상태 변경 데이터들은 감사 데이터 저장소(Audit Data Repository)에 저장되며, 상태 관리 및 모니터링 권한을 가진 사용자는 워크플로우 관리/모니터링 클라이언트 도구(Admin/Monitoring Client Tool)를 통해 워크플로우 엔진이 관리하는 워크케이스 및 관련 정보들에 대한 실시간 파악이 가능하다. 본 논문에서 구현하는 워크케이스 기반 워크플로우 관리 시스템은 다수의 클라이언트들에 의한 요청으로부터 발생하는 대량의 워크케이스들을 유지 관리하기 위하여, Java EE 웹 애플리케이션 서버에서 제공하는 인스턴스 풀링 기능 및 비동

(표 1) 워크케이스 기반 워크플로우 엔진의 구성 컴포넌트 리스트

컴포넌트 명	컴포넌트 구현 타입	컴포넌트의 수행 역할
Workflow Model Agent	Stateless Session Bean	워크플로우 모델링 클라이언트로부터 디플로이된 워크플로우 프로세스 정의의 엔진 가용 모델 데이터로 변환하여 모델 데이터 저장소에 저장함.
Request Agent	Stateless Session Bean	워크플로우 런타임 클라이언트가 전송하는 서비스 요청 메시지들을 요청 메시지 큐에 적재함.
Request Message Queue	JMS(Java Message Service) Queue	워크플로우 서비스 요청 처리 에이전트(Request Agent)로부터 전송된 메시지들을 적재 관리함.
Request Message Handler	Message-Driven Bean	워크플로우 서비스 요청 메시지 큐(Request Message Queue)에 적재되어 있는 메시지를 캡처한 후, 해당 메시지에 상응하는 실제 서비스를 수행함.
Workcase	Stateful Session Bean	워크플로우 모델 인스턴스으로써, 모델 내의 액티비티 순서에 의거하여 업무 흐름 진행, 워크아이템 할당, 클라이언트들과의 상호 작용 등 전반적인 워크플로우 서비스를 주도하는 역할을 담당함.
Script Interpreter	Stateless Session Bean	워크케이스가 업무 흐름을 진행해나가는 과정에서 분기 흐름을 결정해야 하는 경우, 이를 위한 조건 판단 및 룰에 관한 처리 및 결과를 리턴해주는 역할을 수행함.
Worklist Agent	Stateless Session Bean	워크플로우 런타임 클라이언트가 전송하는 워크아이템 관련 서비스 요청 메시지들을 워크리스트 메시지 큐에 적재함.
Worklist Message Queue	JMS Queue	워크리스트 에이전트(Worklist Agent)로부터 전송된 메시지들을 적재 관리함.
Worklist Message Handler	Message-Driven Bean	워크리스트 메시지 큐(Worklist Message Queue)에 적재되어 있는 메시지를 캡처한 후, 해당 메시지에 상응하는 실제 워크아이템 관련 처리 서비스를 수행함.
Application Agent	Stateless Session Bean	워크플로우 엔진이 직접적으로 액티비티 처리를 수행하기 위해 또는 업무 수행자가 자신에게 할당된 워크아이템을 처리하기 위해 필요한 응용 애플리케이션들의 호출, 시작, 종료 및 여러 관리 기능들을 제공함.
Admin/Monitoring Agent	Stateless Session Bean	수행 중 또는 완료된 워크케이스들에 대한 실시간 상태 판 및 모니터링을 위한 수행 기능들을 제공함.

기 메시지 큐잉 서비스를 이용한다. 다음의 (표 1)은 EJB 프레임워크를 이용하는 워크케이스 기반 워크플로우 엔진의 구성 컴포넌트들에 관한 타입 및 역할에 대해 세부적으로 설명하고 있다.

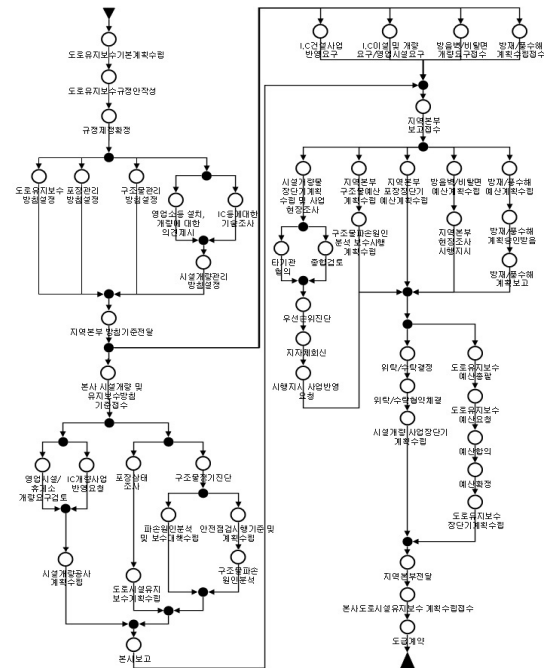
6. 성능 실험 및 평가

이 장에서는 워크케이스 기반 워크플로우 아키텍처가 초대형 워크플로우 서비스를 제공하는데 적합하다는 사실을 실험을 통해 검증하고자 한다. (그림 6)은 현재 한국도로공사에서 운영하고 있는 전국 도로들의 유지보수 및 관리 비즈니스 프로

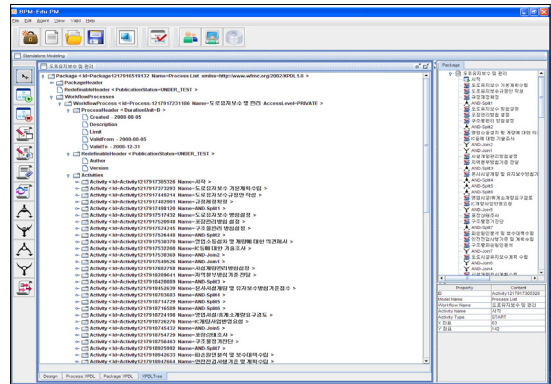
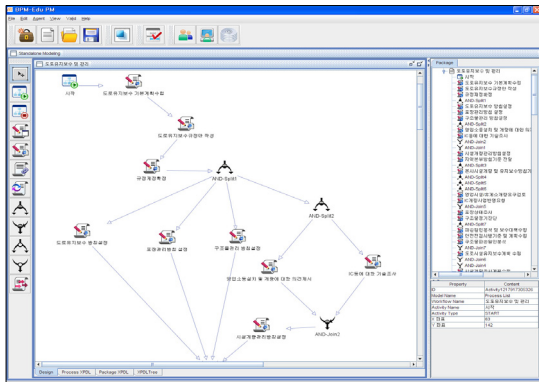
세스에 대한 ICN(Information Control Net)[8,9,31] 기반 워크플로우 모델을 나타내고 있다.

(그림 6)에서의 도로 유지보수 및 관리 워크플로우 모델은 서브플로우 액티비티와 연계되는 워크플로우 모델들의 액티비티들을 포함한 약 100여 개 이상의 액티비티들로 구성되어 있다. 또한 전국적으로 위치해있는 한국도로공사 지점들의 조직 구성원 및 현장 직원들이 각자의 업무 수행을 위해 워크플로우 엔진에게 대량의 작업 요청 메시지들을 발생시키는 초대형성의 특징을 가진다. 이와 같은 이유로 본 논문에서는 워크케이스 아키텍처의 초대형 워크플로우 적합성 검증을 위

해 해당 워크플로우 모델을 이용하고자 한다. (그림 7)은 우리가 자체 개발한 ICN 워크플로우 모델링 도구를 통해 실험에서 사용하기 위한 예제 시나리오에 관한 ICN 정형 нот이션 기반의 그래픽 모델 및 이에 상응하는 워크플로우 표준 실행 언어인 XPD(XML Process Definition Language)[32] 정의 내용을 나타내고 있다.



(그림 6) 도로 유지보수 및 관리에 관한 예제 워크플로우 모델



(그림 7) 예제 시나리오의 ICN 정형 нот이션 및 XPD 정의

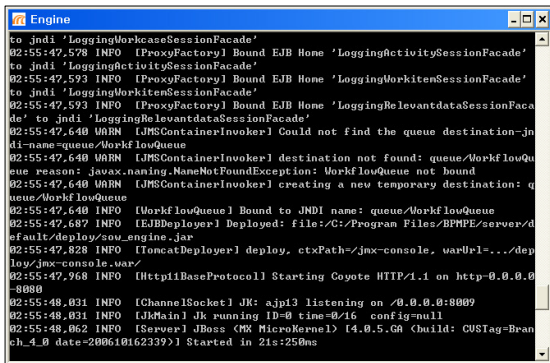
ICN 워크플로우 모델링 도구에 의해 생성된 예제 워크플로우 모델을 모델 데이터 저장소에 디플로이한 후, 실험 대상인 액티비티 인스턴스 기반 아키텍처와 워크케이스 아키텍처들의 초대형 성능 평가를 위해 다음의 (표 2)와 같이 동일한 워크플로우 관리 시스템 환경을 구성한다.

(표 2) 성능 실험을 위한 워크플로우 관리 시스템 환경 설정

대상 요소	설정
CPU	Intel Core2Duo T8300
RAM	1.96GB RAM
네트워크 대역폭	동일
DB 커넥션 풀 사이즈	100
메시지 큐 사이즈	1,000
워크케이스 개체 수	1,000,000
서비스 요청 메시지 도착 분포	랜덤 분포

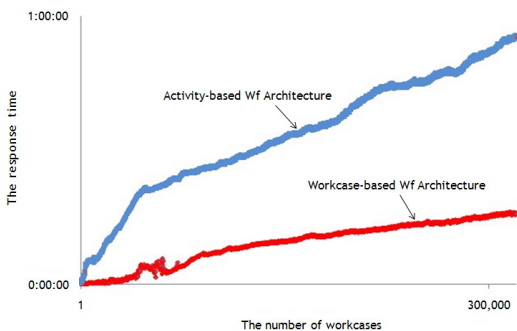
워크케이스 개체 수는 워크플로우 프로세스 시작 권한을 가진 업무 수행자가 워크플로우 런타임 클라이언트 도구를 이용해 워크플로우 엔진에게 워크케이스 생성 및 구동 서비스를 요청하는 수를 의미하며, 본 실험에서 워크플로우 엔진이 관리하도록 지정된 최대 워크케이스 수는 1,000,000개이다. 이 때, 워크케이스들은 클라이언트들이 랜덤 분포에 의거하여 워크케이스 생성 및 구동 서비스를 요청함에 의해 발생된다. 클라

이언트로부터 전송된 요청 메시지들은 지정된 사이즈를 가진 큐에 적재되며, 메시지를 소비하여 실제 서비스를 처리하는데 소요되는 시간은 워크플로우 엔진을 실행하는 하드웨어의 성능과 기타 EJB 컨테이너의 설정에 따라 달라질 수 있다. (그림 8)에서 보는 바와 같이, 본 실험을 위해 동작하는 워크플로우 엔진은 (표 2)에 기술되어 있는 환경 설정 내용들을 토대로 구동한다.



(그림 9) 워크플로우 엔진 구동 화면

(그림 9)는 동일 워크플로우 엔진 상에서 수행하는 액티비티 인스턴스 기반 워크플로우 아키텍처와 워크케이스 기반 워크플로우 아키텍처 간의 워크케이스 생성 반응 시간에 따른 초대형성 성능 측정 결과를 나타내고 있다.



(그림 10) 대상 워크플로우 아키텍처들 간 초대형성 성능 측정 결과

본 성능 측정 결과로부터 알 수 있듯이, 워크케이스 기반의 워크플로우 아키텍처를 이용한 워크플로우 엔진이 액티비티 인스턴스 기반 아키텍처의 경우보다 전체 워크케이스 당 단위 워크케이스 생성 및 구동 소요에 필요한 반응 시간이 짧게 걸린다는 것을 확인할 수 있다. 이러한 결과로부터 워크케이스 기반 워크플로우 아키텍처를 채택한 워크플로우 엔진이 액티비티 인스턴스 기반 워크플로우 아키텍처를 사용한 엔진에 비해 대량의 워크케이스들을 관리하는데 요구되는 초대형성 제공에 보다 적합하다는 점을 도출할 수 있다. 본 실험을 수행하는 과정에서 액티비티 인스턴스 기반 아키텍처를 채택한 워크플로우 엔진은 약 42만 개의 워크케이스를 활성화시킨 시점에서 시스템 진행의 한계를 발생시켰다. 반면에 워크케이스 아키텍처를 선택한 워크플로우 엔진은 약 61만 개의 워크케이스들을 활성화 상태로 유지하는 것이 가능하였다.

7. 결론

오늘날의 기업들은 대규모 엔터프라이즈 환경에서 다양한 유형의 클라이언트들이 발생시키는 대량의 작업 요청들을 보다 빠르고 안정적으로 처리해낼 수 있는 초대형 워크플로우 관리 시스템의 도입을 요구하고 있다. 현재의 워크플로우 벤더들은 초대형 워크플로우 서비스를 제공하기 위하여 워크플로우 관리 시스템을 구성하는 하드웨어적 장비들의 추가적 연동을 바탕으로 한 물리적 확장을 지향하고 있다. 그러나 대다수의 벤더들이 워크플로우 관리 시스템의 소프트웨어적인 처리 메커니즘을 고려하지 않은 채 물리적 추가 확장을 적용함으로써 인하여, 추가된 하드웨어 대비 시간 및 비용적 낭비를 발생시킬 수 있다는 문제점을 가지고 있다.

본 논문에서는 워크플로우 관리 시스템의 소프트웨어적 관점에서 초대형 워크플로우 서비스 제공이 가능한 환경 구축을 지원하는 워크케이스

기본 워크플로우 아키텍처를 제안하고 있다. 워크 케이스 기반 워크플로우 아키텍처는 **OMG JointFlow** 아키텍처로 대표되는 기존의 액티비티 인스턴스 기반 워크플로우 아키텍처에 비해 워크플로우 엔진이 제어 관리해야 하는 인스턴스들의 수를 대폭적으로 감소시킴으로써 시스템의 확장성을 증대시키기 위한 아키텍처이다. 이를 증명하기 위해, 본 논문에서는 현재 대부분의 워크플로우 엔진들이 이용하고 있는 액티비티 인스턴스 기반 워크플로우 아키텍처와 본 논문의 제안 아키텍처인 워크 케이스 기반 워크플로우 아키텍처를 동일하게 설정된 구현 워크플로우 관리 시스템 상에서 동작시켜 두 아키텍처들의 워크플로우 초대형성 성능을 측정하였다. 본 논문의 실험 결과로부터 알 수 있듯이, 워크 케이스 기반 워크플로우 관리 시스템이 액티비티 인스턴스 기반 워크플로우 관리 시스템에 비하여 관리 가능한 워크플로우 인스턴스 수의 한계치가 더 높으며, 각 워크플로우 인스턴스들을 생성 및 구동하는데 소요되는 평균 반응 시간이 더 낮음으로써 보다 나은 초대형성 성능을 나타냄을 알 수 있다.

향후 연구로는 클러스터드 또는 그리드/P2P 인프라로 구성된 워크플로우 관리 시스템들에 워크 케이스 아키텍처를 적용하고 다른 워크플로우 아키텍처들과의 다각적인 실험을 통해 초대형성 성능 평가를 진행해나가고자 한다.

Acknowledement

본 연구는 경기도에서 지원하는 경기도지역협력연구센터사업[2008-111-2]의 일환으로 수행하였음. 또한, 본 연구는 경기대학교 특성화 사업의 연구장비를 활용하여 수행되었음.

참 고 문 헌

- [1] D. Georgakopoulos, M. Hornick, A. Sheth, 'An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure', *The Springer Journal of Distribution and Parallel Databases*, Vol. 3, No. 2, pp.119-153, 1995
- [2] C. A. Ellis, 'Team Automata', *The Proceedings of the ACM Group97 Conference*, 1997
- [3] F. Casati, S. Ceri, B. Pernici, G. Pozzi, 'Workflow Evolution', *The Journal of Data and Knowledge Engineering*, ScienceDirect, Vol. 24, Issue 3, pp.211-238, 1998
- [4] G. Alonso, H. Schek, 'Research Issues in Large Workflow Management Systems', *The Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems*, 1996
- [5] Ken Mei et al., 'OMG jointflow submission: Workflow Management Facility', *Object Management Group(OMG)*, 1998
- [6] C. A. Ellis, C. Maltzahn, 'Chautauqua: A Flexible Workflow System', *The Proceedings of the 30th HICSS Conference*, 1997
- [7] G. Alonso et al., 'Failure Handling in Large Scale Workflow Management Systems', *IBM Research Report, RJ9913*, 1994
- [8] C. A. Ellis, G. J. Nutt, 'Office Information Systems and Computer Science', *The Journal of ACM Computing Surveys*, Vol. 12, Issue 1, 1980
- [9] C. A. Ellis, 'Formal and Informal Models of Office Activity', *The Proceedings of the 1983 World Computer Congress, Paris, France*, 1983
- [10] C. A. Ellis, K. H. Kim, 'A Framework and Taxonomy for Workflow Architectures', *The Proceedings of ACM Group2000: The 4th International Conference on Design for Cooperative Systems*, 2000
- [11] S. M. Wheeler, S. K. Shrivastava, F. Ranno, 'OPENflow: A CORBA Based Transactional Workflow System', *The Journal of the Advances in Distributed Systems*, *The Lecture Notes in Computer Science*, Vol. 1752, pp.354-374, 2000

- [12] I. Choi, C. Park, C. Lee, 'Task-Net: Transactional Workflow Model based on Colored Petri Net', The European Journal of Operational Research, The ELSEVIER, Vol. 136, Issue 2, pp.383-402, 2002
- [13] K. Jensen, 'An Introduction to the Theoretical Aspects of Coloured Petri Nets', The Journal of Concurrency Reflection and Perspectives, The Lecture Notes in Computer Science, Vol. 803, pp.230-272, 1994
- [14] K. Guntzel, 'Web Services-Based Transactional Workflows: Advanced Transaction Concepts', The Proceedings of the OTM Workshops 2003, The Lecture Notes in Computer Science, Vol. 2889, pp.70-82, 2003
- [15] O. Zimmermann, J. Grundler, S. Tai, F. Leymann, 'Architectural Decisions and Patterns for Transactional Workflows in SOA', The Proceedings of the ICSSOC 2007, The Lecture Notes in Computer Science, Vol. 4749, pp.81-93, 2007
- [16] Object Management Group(OMG) and Workflow Management Coalition(WfMC), 'Workflow Management Facility - JointFlow, v1.2', 2000
- [17] J. Cardoso, R. P. Bostrom, A. Sheth, 'Workflow Managements Systems and ERP Systems: Differences, Commonalities and Applications', The Journal of Information Technology and Management, The Lecture Notes in Computer Science, Vol. 5, Number 3, pp.319-338, 2004
- [18] H. W. Kim, 'Modeling Inter and Intra-Organizational Coordination in Electronic Commerce Deployments', The Journal of Information Technology and Management, The Lecture Notes in Computer Science, Vol. 2, Number 3, pp.335-354, 2004
- [19] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, J. Schuldt, N. Weiler, 'WISE: Business to Business E-Commerce', The Proceedings of International Workshop on Research Issues in Data Engineering, pp.132-139, 1999
- [20] H. W. Kim, 'Modeling Inter and Intra-Organizational Coordination in Electronic Commerce Deployments', The Journal of Information Technology and Management, The Lecture Notes in Computer Science, Vol. 2, Number 3, pp.335-354, 2004
- [21] W.M.P. van der Alast, 'Modeling and Analyzing Inter-organizational Workflows', The Proceedings of the International Conference on Application of Concurrency to System Design, IEEE Computer Society, pp.1-15, 1998
- [22] W.M.P. van der Alast, 'Process-oriented Architectures for Electronic Commerce and Interorganizational Workflow', The Journal of Information Systems, Elsevier Science, Vol. 24, Issue 8, pp.639-671, 1999
- [23] J. Meng, S. Su, H. Lam, A. Helal, 'Achieving Dynamic Inter-organizational Workflow Management by Integrating Business Processes, Events and Rules', The Proceedings of the 35th International Conference on System Sciences, 2002
- [24] I. Chebbi, S. Dustdar, S. Tata, 'The View-based Approach to Dynamic Inter-organizational Workflow Cooperation', The Journal of Data and Knowledge Engineering, Elsevier, Vol. 56, Issue 2, pp.139-173, 2006
- [25] J. Klingermann, J. Wasch, K. Aberer, 'Adaptive Outsourcing in Cross-organizational Workflows', The Proceedings of the 11th International Conference on Advanced Information Systems Engineering, Vol. 1626, pp.417-421, 1999
- [26] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig, 'CrossFlow: Cross-organizational Workflow Management in Dynamic Virtual Enterprises', International Journal of Computer Systems

- Science and Engineering, Vol. 15, No. 5, pp.277-290, 2000
- [27] J. Vonk, W. Derks, P. Grefen, M. Koetsier, 'Cross-organizational Transaction Support for Virtual Enterprises', The Proceedings of the 5th IFICIS International Conference on Cooperative Information Systems, Vol. 1901, pp.323-334, 2000
- [28] M. Koetsier, P. Grefen, J. Vonk, 'Contracts for Cross-organizational Workflow Management', Electronic Commerce and Web Technology, Lecture Notes in Computer Science, Vol. 1875, pp.110-121, 2000
- [29] E. Kafeza, D. Chiu, I. Kafeza, 'View-based Contracts in an E-Service Cross-organizational Workflow Environment', The Proceedings of the 2th International Workshop on Technologies for E-Services, Vol. 2193, pp.74-88, 2002
- [30] J. Ball, D. B. Carson, I. Evans, K. Haase, E. Jendrock, 'The Java EE Tutorial', Sun Microsystems, 2006
- [31] C. A. Ellis, J. Gibbs, G. L. Rein, 'Groupware: Some Issues and Experiences', Communication of the ACM, Vol. 34, Issue 1, 1991
- [32] Roberto Shapiro et al., 'Process Definition Interface: XML Process Definition Language, v2.1', Workflow Management Coalition, WFMC-TC-1025, 2008
- [33] K. Kim, 'e-Chautauqua: A Very Large Scale Workflow Management Architecture and System', International Journal of Computer Science & Network Security, 2005.
- [34] H. Ahn, K. Kim, C. Kim, 'Performance Estimations of Clustered Architectures for Very Large-Scale Workflows', 4th Annual International Conference on Computer and Information, pp.288-293, 2005.

● 저 자 소 개 ●



안 형 진

2004년 경기대학교 전자계산학과 졸업(학사)
 2006년 경기대학교 대학원 전자계산학과 졸업(석사)
 2009년 8월 경기대학교 대학원 컴퓨터과학과 졸업(박사)
 2009년 8월~현재 KT 중앙연구소 선임연구원
 관심분야 : 워크플로우/BPM 기술, RFID/USN, 엔터프라이즈 그리드 워크플로우 기술
 E-mail : hjahn07@kt.com



김 광 훈

1984년 경기대학교 전자계산학과 졸업(이학사)
 1986년 중앙대학교 대학원 전자계산학과 졸업(이학석사)
 1994년 Univ. of Colorado at Boulder, 컴퓨터과학과 졸업(이학석사)
 1998년 Univ. of Colorado at Boulder, 컴퓨터과학과 졸업(이학박사)
 1986년~1991년 한국전자통신연구원 연구원
 1998년~현재 : 경기대학교 정보과학부 컴퓨터과학과 교수
 2006년~현재 : GRRC 콘텐츠 융합 소프트웨어 연구센터 센터장
 관심분야 : 워크플로우/BPM 기술, RFID/USN, CSCW, 웹서비스, 분산처리기술, 데이터베이스
 E-mail : kwang@kgu.ac.kr