분산 트랜잭션 워크플로우 운용관리 툴 설계와 구현

°
*이봉석, *강태규, *김광훈, *백수기, **유영철
*경기대학교 대학원 전자계산학과
**한국전자통신연구원, 컴퓨터 소프트웨어 연구소

Design and Implementation of Administration Tool in Distributed Transactional Workflow Management System

°
*Bong-Seok, Lee, *Tae-Gyu, Kang, *Kwang-Hoon Kim, *Su-Ki Paik, *Young-Chul, Lew
*Dept of Computer Science, Kyonggi University
**Electronics & Telecommunications Research Institute

요 약

워크플로우 시스템은 한 조직체 내에서 운용되는 정보와 제어의 흐름을 효과적으로 자동화 해주는 역할을 담당해야 한다. 그러나 기업내의 실제업무에는 많은 변수들이 존재하고 다양하게 변화하고 있다. 또한 시스템이 분산 컴퓨팅화 되어 가기 때문에 그 흐름을 추적, 관리의 차원을 넘어서 업무들의 최적화, 통합화는 물론 분산된 시스템 내 서버들의 관리까지도 개입을 해야한다. 또한 특정작업의 과부하 또는 병목현상을 운용자가 개입하여 방지하고 개선함으로서 전체 작업의 효율을 향상시켜야 한다. 이에 운용관리 툴도 독립서버로 존재하는 것보다 여러 개의 서버로 분산시켜 구축함으로써 수행 객체들을 효율적으로 통제할 수 있다 또한, 시스템을 구성하고 있는 서버들의 관리도 중앙에서 모든 것을 통제하는 것보다 각각의 분산된 운용 서버들이 각각 관리함으로써 신뢰성, 안정성의 한계를 극복할 수 있다. 국제표준기구 역할을 하고 있는 WfMC(Worklow Management Coalision)에서 제시한 표준 모델을 근거로 각각의 인터페이스 규약에 따라 시스템을 개발하는데 있어 실질적인 기업의 업무에 적용되기에는 턱없이 부족한 점들을 이논문을 통해 지적한다. 또한 좀더 빠르게 변화하는 사회에 적격인 적응형(Adaptive) 워크플로우 관리 시스템의 특성을 지니도록 했다. 그리고 데이터베이스의 트랜잭션의 개념을 워크플로우 레벨에서 업무들의 수행시 에러나 장애시 복구처리 메커니즘으로 설계하고 구현을 하였다.

1. 서론

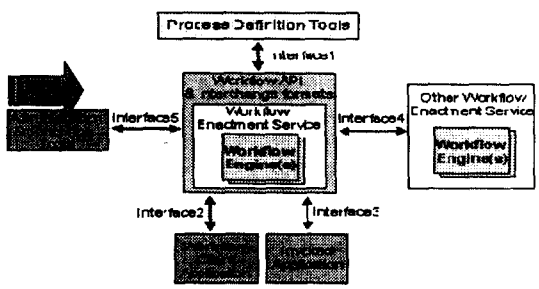
워크플로우 관리 시스템(WfMS)은 비즈니스 프로세스의 자동화를 위미한다. 이러한 WfMS 는 기업 전체를 대상으로 하는 기간 시스템으로서 기업 내의 모든 업무와 자원을 프로세스 관점에서 통합하여 관리하게 된다. WfMS 를 현존하는 기업 내부 데이터와 연계시키고, 프로세스 데이터를 안정된 트랜잭션의 관점에서 다루기 위하여 데이터베이스상에서 WfMS 를 구축하게 된다. 워크플로우 시스템은 크게 빌드타임(Build Time) 모듈과 런타임(Run Time) 모듈로 나뉘며 빌드타임은 기업에서 수행하는 업무들을 모델링하며, 런타임은 정의된 모델의 템플릿을 생성하고 생성된 템플릿이 실질적으로 동작하며 업무를 수행하게 되는데 이때 운용관리 툴과 연동관계를 유지하고 통신하게 된다. 그렇지만 현 기업들의 업무는 기업 특성상 여러가지 다양하게 변화를 가질수 있으며, 조직의 체계나 새로운 업무자체가 새로 생기거나 없어지거나 할 수도 있다. 이는 업무를 모델링 할 때 아무리 많은 업무들을 정의한다 하더라도 실기업의 업무를 모두 합리적이고 적절하게 표현하는데는 다소 무리가 있다. 적응형 워크플로우에서는 이런 사항들을 운용 툴을 통해 운

운용자가 개입함으로써 유연하고 신속하게 대처할 수 있는 부분들을 늘려나갈 수 있다. 따라서 운용툴이 이런 특성들을 갖추기 위해서는 WfMC 표준에서 제공하는 모델로는 턱없이 부족한 면이 많다. 이에 본 논문에서는 그러한 점들을 지적하고 또한 그런 제한사항을 보안하기 위해 추가되는 기능들을 설계하고 구현함으로써 보다 강력하고 안정적인 운용툴을 개발하고자 하였다. 2 장에서는 운용툴의 정의와 기능을 표준안과 비교하여 제시한다. 3 장에서는 개발환경과 수행환경에 대해 설명한다. 4 장에서는 수행 결과에 대해 설명한다. 마지막으로 5 장에서는 결론과 제안사항을 설명한다.

2. 운용툴의 소개

2.1 운용툴의 정의

운용툴은 정의된 프로세스나 실시간에 진행중인 프로세스, 또는 이미 수행이 완료된 프로세스의 상황을 모니터링을 통해 추적하고 추출되거나 조작된 정보를 바탕으로 해서 사용자 또는 역할을 관리하거나 자원 관리, 프로세스의 진행에 관계하여 감독 명령하는 기능을 운용자에게 제공하는 툴을 의미한다.



<그림 1> WfMC 의 모델

<그림 1>과 같이 WfMC 표준안에는 Interface5 으로 정의되며 엔진과의 연동동작을 한다. 공유 데이터베이스를 통해 빌드타임에 정의된 프로세스에 대해서도 운용자가 독자적으로 혹은 요구시에 적극적으로 개입할 수 있다.

2.2 운용툴의 제공기능

WfMC 표준안에서는 운용 툴이 갖추어야 할 최소한의 기능들을 기술하고 있고 또한 주로 업무제어 기능중심으로 모델링되어 있으며 Function 중심으로 분류되어 있다. 하지만 본 논문에서는 이를 자원제어기능을 분산환경을 고려하여 시스템 제어 기능으로 확장했으며 이것의 주요 기능으로는 하나의 시스템이 여러 개의 서버들로 구성이 된다면 각각의 서버에 위치한 수행객체들의 수행상태에 따라 서버의 기능을 정지, 시작시킬수 있으며, 또한 로드 밸런싱을 고려해서 특정서버에 부화에 따라 생성객체의 최대수를 할당할 수 있는 기능을 추가하였으며, 기능 수행에 있어서 객체 중심으로 다시 기능을 분류하고 확장하였다. 설계 및 구현을 통해 작업가능한 기능들은 프로세스 리엔지니어링 즉 비즈니스 프로세스 환경에 변동이 있으면 변동에 따라 반영할 수 있다. 동적변경(Dynamic Reconfiguration)의 개념보단 업무를 수행할 참여자를 변경한다거나 만료일자를 연장하는 기능을 말한다. 개선할 곳이 어디인지를 보며 업무 진행의 관리가 가능하다. 시스템 관리를 서버 맵구조의 환경 설정 파일 유지를 통해 쉽게 할 수 있다. 객체지향으로 되어 있기 때문에 다른 WfMS 에도 쉽게 접목할 수 있다. 이는 표 1 과 표 2 에 나타나 있다.

<표 1> WfMC 기능중심의 분류

사용자 관리 기능 : 사용자의 특권설정, 삭제, 수정
역할 관리 기능 : 사용자와 작업그룹간의 관계정의, 삭제 , 수정
자원 제어 기능 : 프로세스나 엑티비티 인스턴스들간의 동시성 수준 지정, 삭제, 수정
프로세스 관리 감독 기능 : 프로세스나 엑티비티 인스턴스의 상태변경, 속성할당, 종료, 강제종료

<표 2> 추가기능을 포함한 객체중심의 분류

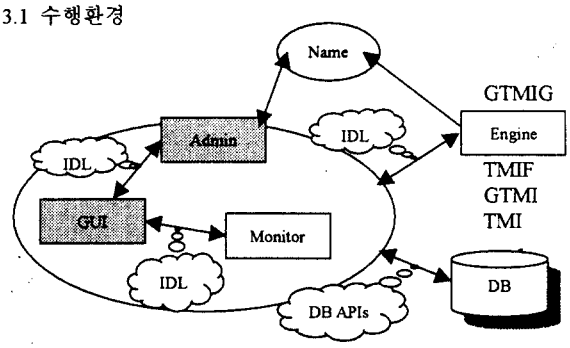
시스템 제어 레벨 : 서버 시작, 중지, 최대 생성 가능 객체수 할당
사용자/역할 레벨 : 사용자나 작업그룹간의 특권 설정, 관계정의, 삭제, 수정
프로세스 정의 레벨 : 속성할당, 탬플릿생성, 버전관리
프로세스 인스턴스 레벨 : 프로세스 인스턴스들의 상태변경, 속성할당, 종료, 강제종료
엑티비티 인스턴스 레벨 : 엑티비티 인스턴스의 상태변경, 속성할당, 종료, 강제종료

3. 수행 환경 및 구현

현재 워크플로우 제품들은 많이 출시되어 있지만 기업에 적용되어 성공을 거둔 제품은 보기 드물다. 이는 인터넷의 급속한 발달과 CALS, EC 등이 등장하면서 그러한 것들과 접목해서 구축하려면 웹-기반이어야만 하고, 사용자들의 전문적 지식이 없이도 사용할 수 있게 하려면 편리한 유저 인터페이스를 제공하여야 하며, 업무수행도중 오류나 장애가 발생했을 경우 경우에 따라 시스템 차원에서 자동적으로 복구를 하거나 대체적인 처리를 해주어야 하기 때문이다. 또한 다양하게 변하는 어떠한 업무에도 적용할 수 있도록 지원가능 하여야 하며, 타 WfMS 와의 Interoperability 를 보장하는 안정적인 특징을 갖지 못하기 때문이다.

따라서 이렇듯 견고하고 유용한 WfMS 의 특징을 지니기 위해선 설계단계에서 충분한 고려를 해야한다. 그러한 것들에 대해서는 운용 툴이 대처할 수 있는 기능들을 강화하여야만 한다.

3.1 수행환경

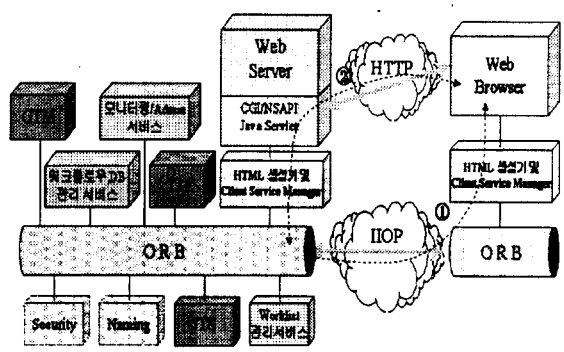


<그림 2> 수행환경도

CORBA 의 Naming Service 를 이용하며 중요한 특징은 운용/모니터링은 하나의 모듈이지만 내부적으론 각각 분리하는 방식을 채택한다. 이는 적용형 워크플로우를 지향함을 염두하고 강력한 운용/모니터링 도구 개발을 위해 서로간의 통신을 위한 인터페이스를 최소하기 위함이다. 또한 엔진의 객체형태로 존재하는 각각의 컴포넌트들의 레퍼런스들을 관리함으로써 직접 연동하여 수행도에서 DB 만을 접근하는 것보다 높일 수 있다. 공유 DB 에는 조직도 관련 내용과 빌드타임시 정의된 프로세스의 모든 내용이 저장되어 있으며 운용자는 필요시 사용자 인증을 획득한 후 운용툴을 통해 해당 내용을 삽입, 삭제, 조회, 업데이트를 DB Operation 을 통해 할 수 있다.

운용 서버의 최초 구동시 엔진의 GTMIG(Grobal Task Managing Instance Generator)로부터 생성된 레퍼런스를 CORBA 의 네이밍 서비스를 이용하여 획득하고 클라이언트로부터 요구가 발생하면 해당 객체와 직접 연동동작을 한다.

3.2 시스템과 클라이언트와의 연결 구조



<그림 3> component connectivity

클라이언트(운용자)는 웹 브라우져를 근간으로 경량 클라이언트(Thin Client)구조로 워크플로우 서버와 연결되어 있다. 클라이언트와 운용 서버의 연결은 클라이언트 환경에 ORB 가 존재하는 경우에는 IIOP 를 사용하여 운용 서버와 통신하며 클라이언트 환경에서 ORB 사용이 불가한 경우에는 HTTP 프로토콜을 사용하여 웹 서버를 경유

하여 통신한다.

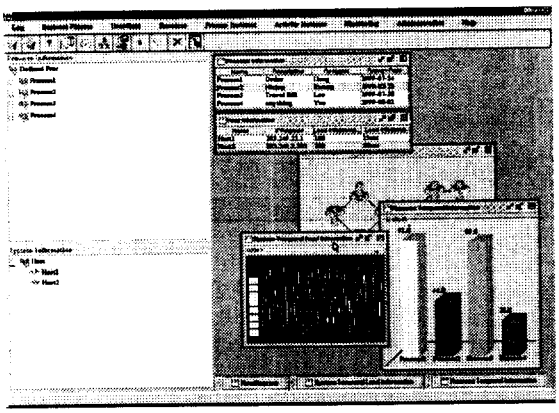
3.3 구현 환경

Unix나 NT환경에서 구현되거나 이들의 복합된 환경에서 구현되며, 통신 인프라로 CORBA(ORB)를 채용하고 플랫폼의 독립성을 보장하기 위해 Java 언어로 구현된다.

- OS : Windows 95/98/NT, Solraris sparc 2.6
- ORB : javaIDL
- 구현언어 : JDK 1.2 이상
- 개발툴 : Jbuilder 3.0
- 데이터베이스 : Oracle Server 8.0.0.4.0.0

4. 수행 결과

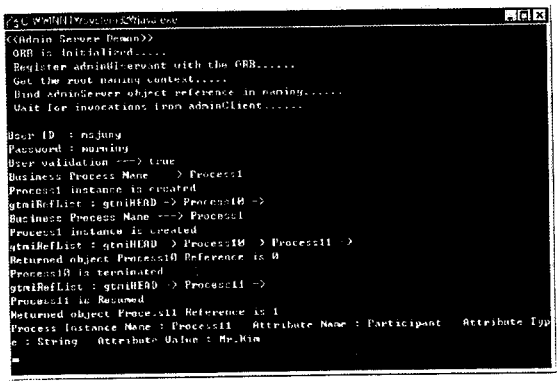
4.1.1 클라이언트(운용자 GUI)



<그림 4> 운용툴의 GUI

레벨별로 분리된 기능을 그대로 메뉴화하여서 운용자가 사용하기 쉽게 하였고, 많이 사용할 것으로 예상되는 기능들은 단축아이콘으로 구성하였다. 프로세스나 인스턴스들과 시스템에 연결되어 있는 서버들의 계층적 구조를 보기쉽게 2개의 트리 형태로 보여주고 있으며, 해당 노드들에 대한 Description은 테이블형태로 필요시 자세하게 나타내주게 되고, 다른 수치,통계정보는 모니터링기능을 이용하여 막대그래프와 라인그래프로 그래픽하게 나타냄으로써 사용자가 한눈에 알아보기 쉽게 하였다.

4.2 운용 서버



<그림 5> 운용툴 서버 데몬

운용 서버는 먼저 운용자 로그인을 담당하고 인증된 사용자인 경우 사용자 요구가 발생하는 경우 요구의 종류에 따라서 공유 데이터베이스나, 혹은 관리하고 있는 레퍼런스를 이용하여 엔진내 해당 수행객체와 연동동작을 한다. 객체들의 레퍼런스는 객체생성시 리스트에 삽입되고 수행이 끝나고 소멸시에는 리스트에서 제거된다.

5. 결론 및 제안 사항

운영서버가 분산 아키텍처를 지님으로 해서 툴의 안정성과 신뢰성을 보장하며, 많은 클라이언트에게 좀 더 빠른 서비스를 제공하게 된다, 또한 객체지향으로 되어있고 모니터링과 완전 분리되어 있어서 적응형 워크플로우의 운용툴로 발전시키거나 기능 확장시 용이함을 제공하고 있다.

현재는 운용서버가 독립서버로 서비스를 하는 형태를 가지고 있지만 이는 예상치 못한 이유로 서버 fail시 서비스를 계속할 수 없는 상태를 초래할 수도 있다. 따라서 전반적인 운용툴의 안정성을 높이기 위해 서버들에 의존적으로 분산하는 구조가 바람직하며, 운용 서버의 상태에 따라서 부하가 많을 경우도 다른 서버와 연결되서 계속 서비스를 받을 수 있도록 하는 특징을 지녀야 한다 또한 전체 시스템의 분산 구조를 고려해 시스템 레벨에서 운용자가 개입할 수 있는 부분을 찾고 기능을 더욱 확장해야 한다. 이는 조직체들이 다양하게 변화하는 기업환경과 분산 컴퓨팅 환경에 적절하게 대처하는 방안이 된다.

참고 문헌

[1] 컴퓨터 소프트웨어기술연구소
HANURI/TFlow System Design Specification
1998.12.22

[2] Johann Eder, Walter Liebhart
"Workflow Transactions" Dept of Informatics,
UNIV of Klagenfurt, Austria. January, 1996

[3] Paul Harmon And Mark Watson
"Understanding UML The Developer's Guide
With A Web-Based Application In Java " 1998

[3] Workflow Management Facility
"OMG Business Object Domain Task Force"
BODTF-REF 2 Submission

[4] "Workflow Management Coalition Audit Data
Specification" Document WFMC-TC-1015 , 1996

[4] ETRI 데이터공학연구부
한우리 2 공동연구사업 개요 및 99년도 중간보고서
1998. 7 .2

[5] "Workflow Management Coalition Terminology &
Glossary" Document number WFMC-TC-1011, 1996

[6] Nortel and University of Newastle upon Ytne, (1998)
" Workflow Management Facility Specification",
Revised submission, OMG document bom/98-03-01

[7] "Workflow Automation : Applications, Technology, and
Research"
SIGMOD conference 1995

[7] http://www.wfmc.org/standard

[8] http://www.omg.org