

설계 프로세스 관리를 위한 웹 기반 워크플로우 시스템

김영호*, 주경준**, 김동수***, 강석호****

A Web-based Workflow Management System for Design Processes

Kim, Y. H.* , Ju, K. J.** , Kim, D. S.*** and Kang, S. H.****

ABSTRACT

The design and development of a Web-based Workflow Management System (WfMS) is described in this paper. WfMSs are employed by many organizations to manage their business processes efficiently. The goal of this research is to provide an effective management of design processes in a distributed and heterogeneous environment-an environment in which workflow participants are geographically dispersed and use different computing platforms. We identify the requirements of WfMSs for the distributed and heterogeneous environment. We present an approach to applying the Web technology to WfMS. The developed system is composed of three main modules: Workflow Engine, Web-based Clients, and Process Designer. The workflow engine is implemented in pure Java for portability to heterogeneous platforms. Furthermore, the client interfaces are all programmed using applets, and thus they can be readily accessed through the Internet/intranet. The Process Designer provides a build-time function of generating process specifications. Some important functions of the developed system are explained in this paper. A prototype system has been developed and applied to design processes. The system allows effective cooperation among designers who may be geographically separated.

Key words : Design Process, Internet, Workflow Management System

1. 서 론

워크플로우는 컴퓨터에 의해 자동으로 실행 및 관리되는 업무 프로세스를 말하며, 워크플로우관리시스템은 이 워크플로우를 정의하고 그에 따라 업무 프로세스를 실행하고 관리하는 소프트웨어이다⁽¹⁾. 기업 환경의 급속한 변화에 따라 워크플로우관리시스템의 중요성이 아주 높아지고 있다. 예를 들면, 최근 주목 받고 있는 전사적자원관리시스템(ERP: Enterprise Resource Planning), 제품정보관리시스템(PDM: Product Data Management), 공급망관리시스템(SCM: Supply Chain Management), 고객관계관리시스템(CRM: Customer Relationship Management), 전자상거래(EC:

Electronic Commerce) 등과 같은 대부분의 기업정보시스템은 이 워크플로우관리 기능을 핵심 기능으로 포함하고 있다. 이는 이들 기업정보시스템이 지원하는 업무가 대부분 절차에 따라 진행되는 프로세스를 가지고 있기 때문이다. 특히, PDM 시스템은 제품과 관련된 모든 정보의 생성 및 접근을 통제하여 전체 제품수명주기 상에서 정보 공유를 원활히 하고, 또 그 흐름을 효율적으로 관리하기 위한 시스템으로, 프로세스의 관리 기능이 필수적이라고 할 수 있다⁽²⁾. 따라서 설계 프로세스를 관리하기 위한 워크플로우관리 기능은 PDM 시스템의 근간을 이루는 요소라고 볼 수 있다.

워크플로우관리시스템은 다음과 같은 장점을 제공하여 업무 생산성을 제고한다. 첫째, 업무 흐름을 자동으로 통제하기 때문에 설계 프로세스와 같이 복잡한 프로세스를 거치는 업무 흐름도 손쉽게 처리할 수 있다. 둘째, CAD 데이터와 문서의 전달이 전자화되므로 업무를 신속히 처리할 수 있고, 또 불필요

*중신회원, 서울대학교 산업공학과
**한국전자통신연구원 컴소연 동시공학 연구팀
***학생회원, 서울대학교 산업공학과
****서울대학교 산업공학과

한 종이 문서 사용을 줄일 수 있다. 셋째, 데이터를 일관적으로 접근하고 제어하므로 데이터 불일치에 따른 문제점이나 정보의 부정확성이 감소하고, 따라서 업무 처리 결과의 정확도가 높아진다. 넷째, 업무 진행 현황을 실시간에 투명하게 감독할 수 있다. 다섯째, 그룹웨어 기능을 가지므로 능률적인 공동작업이 가능하다. 이러한 장점 때문에 워크플로우관리시스템은 앞에서 언급한 정보시스템의 핵심 모듈로 취급되고 있다.

본 연구의 목적은 인터넷/인트라넷을 이용하여 분산 환경에서의 설계 프로세스를 효율적으로 관리하기 위한 웹 기반 워크플로우관리시스템을 개발하는 것이다. 인터넷 사용 인구가 급속도로 증가함에 따라 많은 정보 시스템들이 웹 인터페이스를 제공하고 있다. Internet Explorer나 Netscape Navigator와 같은 웹브라우저는 사용자 인터페이스의 표준이라고 할 정도로 많은 사용자가 친숙한 인터페이스이다. 워크플로우관리시스템의 경우에도 많은 업체들이 웹 기반 시스템을 개발하여 제공하고 있다. 웹 기반 워크플로우관리시스템의 가장 큰 장점은 웹브라우저를 통해서 시스템에 접근할 수 있으므로 시간이나 장소에 구애 받지 않고서 시스템을 사용할 수 있다는 것이다. 또, 별도의 클라이언트 프로그램을 설치할 필요 없이 다운로드 받아 사용하므로 클라이언트 프로그램의 운용, 유지보수, 수정이 편리하다는 장점도 있다. 한편, 본 연구의 대상이 되는 프로세스는 설계 프로세스이다. 설계 업무 프로세스에서 생성되고 전달 및 저장되는 CAD 도면과 문서 등의 데이터 흐름과 설계 프로세스의 흐름을 인터넷을 통해 효과적으로 관리할 수 있는 시스템을 설계하고 개발하였다.

이 논문에서는 설계 프로세스를 관리하기 위한 웹 기반 워크플로우관리시스템의 설계와 개발에 대하여 설명하였으며, 다음과 같이 구성된다. 2장에서는 이질 분산 환경에서 워크플로우관리시스템이 갖추어야 할 요구사항 및 관련 연구를 설명하였다. 3장에서는 개발한 시스템의 전체 구조와 시스템 설계 사양을 설명하였다. 4 장에서는 본 연구에서 개발한 시스템의 주요 기능을 기술하였다. 5장에서는 결론 및 추후 연구 과제를 서술하였다.

2. 시스템 요구사항 및 관련 연구

2.1 시스템 요구사항

최근의 기업 업무 환경은 지리적으로 분산되어 있고, 상호 이질적인 전산 시스템을 사용하며, 공동 작

업의 중요성이 증가하는 것으로 특징지을 수 있다. 특히, 조직의 경계를 넘어 전세계적으로 다른 조직과 업무가 상호 연결되어 발생하는 경우가 많아 이질적인 환경 간의 통합이 중요한 문제로 부각되고 있다. 이러한 환경에서 설계 업무를 지원하기 위한 WfMS가 만족해야 하는 필수적인 요구사항을 규명하였으며, 이 가운데 본 연구와 주로 연관되는 주요 사항을 아래에 열거하였다.

- 공동작업 지원: 일반적으로 설계 업무에는 많은 업무담당자들이 공동 참여하며, 여러 분야의 전문 지식을 동원해서 문제를 해결해 나간다. 참여자들의 의사결정을 도와 주고 상호 관계를 조정하며 다양한 분야 전문가들이 협력할 수 있는 공동작업환경을 제공하는 것이 필요하다.
- 플랫폼 독립성(Platform independence): 워크플로우 엔진은 플랫폼에 독립적이어야 한다. 여러 하드웨어 및 운영체제 하에 설치되어 운용될 수 있어야 함을 의미한다.
- 위치 투명성(Location transparency): 이는 지리적으로 분산되어 있거나 자주 근무 지역을 이동하는 클라이언트 사용자를 지원하기 위한 요구사항이다. 위치 투명성을 지원하기 위해 클라이언트 프로그램을 서버로부터 다운로드 받아서 실행하는 것이 가능해야 한다.
- 효과적인 작업 통보: 조직의 경계를 넘나드는 프로세스를 관리하기 위해 태스크가 담당자에게 할당된 사실을 통보하는 효과적 메커니즘이 필요하다. 수행해야 할 업무를 주기적으로 확인할 의무가 없는 외부 사용자들은 적절한 방법을 통해 업무 도착 사실을 통보 받을 수 있어야 한다.
- 정보 공유: 분산 이질 환경에서 정보 공유는 조직의 장벽으로 인해 제한을 받는다. CAD 도면, 관련 문서 등과 같이 필요한 정보의 공유를 지원해 주는 시스템이 요구된다. 보다 활발한 정보 공유를 위해 자동 정보 획득 에이전트가 도입될 수 있다.

이러한 요구사항 이외에도 규모확장성(scalability), 사용자 인증 및 보안, 개방형 아키텍처, 고장 대응력(fault-tolerance), 병행 처리(con-current processing) 등과 같이 전통적 WfMS에서도 자주 거론되는 부가적 요구사항이 있다.

2.2 관련 연구

워크플로우관리시스템에 대한 많은 연구 가운데 본 연구와 직접 관련되는 주요 연구 결과를 분산 이질 환경에 대처하기 위한 WfMC(Workflow Management Coalition)의 표준화 관련 연구와 웹 기반 시스템 관련 연구로 나누어서 서술하고 본 연구와의 연관성을 알아보려고 한다.

먼저, 워크플로우의 표준화에 대한 연구로는 WfMC의 작업이 대표적이라 할 수 있다¹⁴⁾. 이 단체에서는 워크플로우의 인터페이스와 메타 모델에 대한 표준화를 주도하고 있다. WfMC에서 제안한 워크플로우 참조 모델에서는 워크플로우관리시스템을 프로세스 정의 도구, 워크플로우 구동 서비스, 클라이언트, 호출 애플리케이션, 관리 및 모니터링 도구 등의 구성 요소로 정의하고 있다. 본 연구에서 개발한 시스템도 WfMC의 표준 모델과 동일한 구조를 가지고 있다.

웹 기반 워크플로우시스템에 대한 연구도 인터넷 관련 기술이 발전함에 따라 동반하여 발전하고 있다. Georgia 대학의 LSDIS 연구실에서 개발한 WebWork는 CGI(Common Gateway Interface), 자바스크립트 등을 이용하여 개발한 초보적 시스템이라고 할 수 있다¹⁵⁾. Dartmouth 대학의 DartFlow는 이동 에이전트(mobile agent)와 CGI, 자바 등의 기술을 이용하여 개발한 시스템이다¹⁶⁾. 대표적 상용 웹 기반 시스템으로는 InConcert 사의 InConcert 2000¹⁷⁾, Staffware 사의 Staffware 2000¹⁸⁾, Action Technologies 사의 Metro¹⁹⁾, Ultimus 사의 Ultimus²⁰⁾, Fujitsu 사의 i-Flow²¹⁾ 등이 있다. 대부분의 시스템들은 웹가능(Web-enabled) 시스템으로 기존의 독립적인 클라이언트와 함께 클라이언트 인터페이스를 인터넷 환경에서 제공하고 있다. i-Flow와 같은 시스템은 웹 기반 시스템으로 웹 인터페이스와 함께 자바 워크플로우 엔진을 제공하고 있다.

2.1절에서 토의한 요구사항과 관련하여 기존 시스템의 기능과 구조를 분석한 결과 현재 대부분의 시스템들이 모든 요구사항을 충족하지는 못하고 있다. 본 연구에서 개발한 시스템은 분산 이질 환경에서의 실제 프로세스를 효과적으로 관리할 수 있는 웹 기반 시스템이라는 것이 가장 큰 특징이다. 네트워크를 통해 연결된 다수의 인원이 공동으로 참여하는 실제 프로세스를 투명하게 정의하고 관리할 수 있을 뿐 아니라 설계와 관련된 각종 정보의 흐름과 공유를 가능하도록 해 준다. 또한, 플랫폼 독립적인 설치와 접속이 가능하고, 업무담당자의 위치투명성을 보장해 주는 시스템이다.

3. 시스템 설계

본 장에서는 전체 시스템 아키텍처를 설명하고, 사용한 핵심적 인터넷 기술인 서블릿의 개념에 대하여 설명하였다. 마지막으로, 중첩 프로세스 모델의 개념과 시스템 설계 사양을 제시하였다.

3.1 시스템 아키텍처

본 연구에서 개발한 웹 기반 워크플로우관리시스템의 전체 아키텍처는 Fig. 1과 같다.

그림에서 보는 바와 같이 시스템은 크게 프로세스 정의기, 워크플로우 엔진, 클라이언트 모듈로 구성되어 있다. 프로세스 정의기는 독립 실행 프로그램으로 프로세스 설계자가 업무 프로세스를 모델링하기 위해서 사용한다. 모듈화된 프로세스 모델링을 지원하고, 설계된 프로세스 정의를 엔진이 이해할 수 있는 형태로 변환하여 이를 데이터베이스에 저장한다. 신규로 프로세스를 모델링 한 경우에는 모델의 오류를 먼저 검증한 후 데이터베이스에 저장한다. 저장된 프로세스 모델은 언제든지 프로세스 정의기에서 수정하거나 다른 프로세스 모델의 설계에 활용할 수 있다.

워크플로우 엔진은 프로세스 데이터베이스에 저장된 프로세스 정의에 따라 전체 업무흐름을 감독하고 통제한다. 프로세스 진행 상황에 따라 업무 실행 스케줄을 만들어서 프로세스를 진행시킨다. DB에 저장된 프로세스 정의에 따라 업무담당자를 결정하여 태스크를 할당한다. 엔진은 자바 서블릿 API를 이용하여 구현하였는데, 다음 절에서 서블릿의 개념과 본 연구에서 채택하게 된 이유에 대하여 상세히 설명하였다. 엔진은 순수 자바로 구현되었기 때문에 2장에

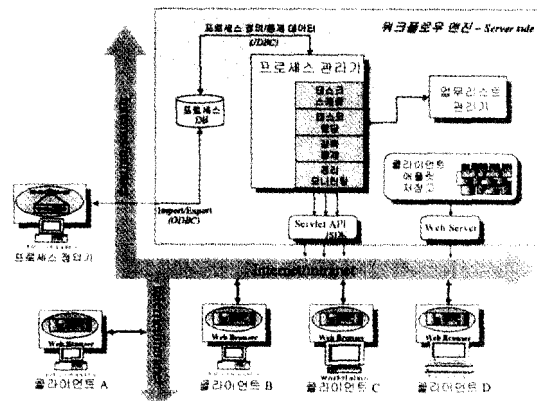


Fig. 1. Overall system architecture.

서 언급한 주요 요구사항인 플랫폼 독립성을 보장해 준다. 일단 엔진이 태스크를 할당하고 나면, 업무리스트 관리기에서 태스크들을 관리한다. 각 사용자는 자신의 업무리스트로부터 태스크를 선택하여 작업한다. 이러한 풀 방식과 함께, 간단한 푸시 기법인 E-mail을 통한 통보 기능이 제공된다. 이는 2장에서 언급한 요구사항 중 하나인 효과적인 작업 통보 기능을 충족시킨다. 태스크가 할당될 때마다 동시에 엔진에서 자동으로 담당자에게 E-mail을 발송한다. E-mail은 PDA나 핸드폰 등의 이동 통신 장비를 통해 즉각 전달 받을 수 있기 때문에 태스크 할당 사실을 즉시 확인할 수 있다. 또한, 시스템에 자주 참여하지 않는 사용자에게도 유용한 작업 통지 수단이 된다. 이 밖에도 엔진은 수행 독촉, 중지, 일시중지(재개) 등과 같은 관리 감독 기능과 실행 중 프로세스 모니터링, 완료된 프로세스 및 태스크에 대한 통계 자료 제공 기능 등도 지원한다.

클라이언트 모듈은 사용자가 직접 상호작용하는 인터페이스이다. 사용자는 자신의 역할과 책임에 따라서 초기화 사용자, 일반 사용자, 시스템 관리자로 분류된다. 모든 사용자 인터페이스는 자바 애플릿으로 구현하였기 때문에 웹 브라우저를 통해서 다운로드 받는다. 이렇게 함으로써 2장에서 언급한 주요 요구사항 중 하나인 위치투명성을 보장 받을 수 있다. 또한, 웹 기반 클라이언트/서버 시스템의 일반적인 장점인 유지보수가 용이하다는 장점이 있다. 정보 공유에 관한 요구사항을 만족시키기 위해서 클라이언트 인터페이스는 웹 환경에서 작동하는 문서관리 시스템과 상호 연결되어 있다. 이 문서관리 시스템은 CAD 도면, 관련 문서 등을 효과적으로 저장, 관리, 공유하기 위해 개발되었으나, 상세한 내용은 본 논문의 범위를 벗어나는 내용으로 논의를 생략하였다.

3.2 자바 서블릿

엔진과 클라이언트를 구현하기 위해 클라이언트와 서버 사이의 통신용 API (Application Programming Interface)를 제공해 주는 자바 서블릿을 구현 도구로 채택하였다. 자바 서블릿은 자바기능 웹서버와 같은 요청/응답 지향 서버를 확장하는 모듈이다¹¹⁾. 서블릿과 서버의 관계는 애플릿과 클라이언트의 관계와 같다. 그래픽 사용자 인터페이스나 웹 브라우저를 통해 작동하지 않는다는 점에서 서블릿은 애플릿과 다르다. 대신, 서블릿은 Jrun이나 Servletrunner와 같이 웹서버 상에서 작동하는 서블릿 엔진과 요청 및 응

답을 통해 상호작용한다. 클라이언트 프로그램은 웹 브라우저나, 또는 인터넷을 통해서 연결되어 웹서버를 통해서 요청을 하는 다른 프로그램이 될 수 있다. 웹서버와 같이 작동하는 서블릿 엔진은 클라이언트의 요청을 처리하는데, 서블릿 프로그램 쪽으로 응답을 반환한다. 다시 서블릿 프로그램은 HTTP 폼으로 클라이언트에게 응답을 전달한다.

서블릿은 CGI 스크립트의 효과적인 대안이다. 본 연구에서 서블릿을 채택한 이유 중의 하나는 서블릿은 다중스레드 환경에서 사용자의 동시 요청을 처리할 수 있다는 점이다. 서블릿은 서블릿 엔진의 매핑에 의해 클라이언트 요청에 대한 응답을 보내 준다. 매핑은 서블릿 인스턴스와 그 서블릿이 데이터를 반환할 클라이언트의 URL의 연결관계를 말한다. 그러나, 매핑이 반드시 일대일일 필요는 없으며, 여러 개의 서블릿 인스턴스와 하나의 URL이 매핑될 수도 있다. 예를 들면, 처리 부하를 균등화하기 위해 여러 개의 서버에서 작동하는 분산 서블릿 엔진의 경우 각 서버마다 서블릿 인스턴스가 동작할 수도 있다.

서블릿은 여러 요청들을 동시에 처리할 수 있고, 요청들을 동기화시킬 수 있다. 서블릿은 요청을 다른 서버나 서블릿에게 전달할 수도 있는데, 이러한 이유에서 서블릿은 태스크의 종류나 조직의 경계에 따라서 동일한 내용을 처리하는 여러 서버들 사이에 부하를 균형화하고, 하나의 논리적 서비스를 여러 서버에 걸쳐 분할하는데 이용될 수 있다. 그러므로, 본 연구에서는 서블릿이 분산 이질 환경에서 작동하는 설계 프로세스를 관리하기 위한 WIMS의 효율적인 개발 도구라고 판단하였다. 서블릿은 자바로 개발되는데, 자바가 상대적으로 프로그래밍이 쉽기 때문에 전체 개발 노력과 시간이 감소하는 장점도 있다.

3.3 중첩 프로세스 모델 지원

일반적으로 프로세스의 구조는 이를 구성하는 여러 태스크와 태스크 간의 선후관계로 정의된다. 전통적 접근법에서 태스크는 프로세스를 구성하는 요소로서 프로세스의 하위 개념이다. 그러나, 중첩 프로세스 모델에서 태스크는 자신의 업무 내용을 더욱 자세히 기술하고 있는 프로세스를 포함할 수 있다. 태스크는 프로세스를 포함하는 동시에 자기 자신은 다른 프로세스의 구성요소가 될 수 있다. 따라서 태스크와 프로세스 간에 부모-자식 관계가 교차해서 발생할 수 있다. 본 연구의 중첩 모델에서는 루트 태스크(root task)가 최상위 객체가 된다.

중첩 프로세스 모델링의 장점은 다음과 같다.

- 하향식(Top-down)으로 프로세스를 모델링하므로, 복잡한 프로세스도 쉽게 설계하고, 분석할 수 있다.
- 캡슐화, 다형성, 상속성 등과 같은 객체지향모델의 장점을 프로세스 모델링과 시스템 구현에 차용할 수 있는 이론적 근거를 제공하여 프로세스 모델링과 시스템 개발이 간편해진다.
- 자주 사용하는 프로세스 단위들을 라이브러리 형태로 제공함으로써 모델의 재사용성이 증가하고, 또한 프로세스 설계에 필요한 노력이 줄어든다.
- 프로세스 하나하나가 분산 환경에서 독립적으로 통제와 제어가 가능한 단위이므로 실행시간 캡슐화(Run-time Encapsulation)의 기초가 된다. 실행시간 캡슐화는 전술한 프로세스 중첩 설계에 바탕을 두고 분산된 여러 워크플로우 엔진이 하나의 프로세스를 구성하는 몇 개의 서브프로세스를 각각 독립적으로 처리하는 기능을 말한다. 실행시간 캡슐화에 대한 상세한 논의는¹¹⁾을 참조하기 바란다.
- 이기종 워크플로우 시스템간의 상호운영성을 위한 방안으로 사용될 수 있다.

설계 프로세스와 같이 비정형적인 프로세스, 즉, 사전에 완전히 정의하기 어렵고, 실행시간에 프로세스 내용이 결정되거나 변경이 발생하는 프로세스의 경우 본 연구에서 제안한 중첩 프로세스 모델링과 실행시간 캡슐화 기능을 통해 효율적으로 관리 가능하다. 제품을 개발하기 위한 설계 프로세스는 여러 부서의 인원들이 참여하는 복잡한 작업이다. 본 연구의 프로세스 모델링은 중첩 프로세스 모델을 지원하기 때문에 프로세스 정의 시 부서 수준의 중첩 태스크로 정의하여 프로세스를 실행시키고, 실행시간에 담당 부서에서 상세한 프로세스로 전개시켜 수행하는 것이 가능하다.

3.4 핵심 DB 스키마, 태스크 상태 및 분기 유형

프로세스 정의기를 통해 생성된 프로세스 모델은 데이터베이스에 저장된다. Fig. 2는 중첩 모델링을 지원하기 위한 데이터베이스 스키마를 보여 주고 있다. 테이블 이름과 주요 애트리뷰트가 그림에 나타나 있다. WF_PROCESS와 WF_TASK 테이블은 프로세스에 대한 구조적 정의와 관련한 정적인 정보가 저장된다. 반면, WF_PROCESS INSTANCE와 WF_TASKINSTANCE 테이블에는 프로세스 인스턴스가 실행될 때 생성되는 동적인 운용 정보가 저장

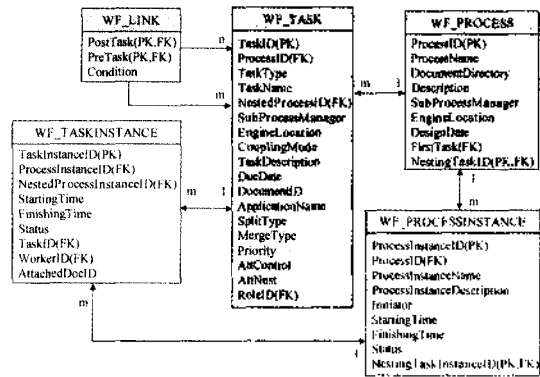


Fig. 2. Important database schema.

된다.

WF_TASK 테이블에는 NestedProcessID라는 애트리뷰트가 있다. 태스크가 중첩 태스크인 경우 이 필드는 중첩 프로세스를 가리킨다. 만일, 태스크가 더 이상 분해될 수 없는 원시 태스크인 경우 'NULL'값을 갖는다. WF_PROCESS 테이블의 NestingTaskID 애트리뷰트는 자신을 중첩하고 있는 중첩 태스크를 가리킨다. WF_PROCESS 테이블에는 구성 태스크 중 첫 번째 태스크를 의미하는 FirstTask 필드가 있다. WF_LINK 테이블에 저장된 선후관계를 참조하여 모든 후행 태스크들을 찾을 수 있다. WF_TASK 테이블에 있는 CouplingMode, SubProcessManager와 EngineLocation 필드는 설계 프로세스를 효과적으로 정의하고 실행 관리하는 기능인 실행시간 캡슐화를 지원하기 위한 것으로¹¹⁾을 참조 바란다. 나머지 애트리뷰트들은 이름으로부터 쉽게 그 역할을 유추할 수 있다.

본 연구에서 개발한 시스템에서 사용하는 태스크와 프로세스의 속성 및 행동양식을 설명하기로 한다. Fig. 3은 개발한 시스템의 구성요소들 간의 상호작용을 보여 주고 있다. Kumar와 Zhao의 연구¹²⁾와 Casati 등의 연구¹³⁾ 등과 같은 많은 연구에서도 유사한 상호작용이 설명된 바 있다.

그림에서 서버 모듈에 해당되는 모듈들은 사각형으로 표시되어 있고, 클라이언트 모듈은 원으로 표시되었다. 가장 중요한 상호작용은 워크플로우 엔진과 일반 클라이언트 모듈 사이에서 일어 난다. 일반 클라이언트 측에서 할당된 작업을 완료하고 나면 엔진으로 리턴 코드를 반환한다. 리턴 코드는 FINISH, REJECT, SUSPEND, RESUME, ABORT, 및 FAIL 중 하나이다.

모든 태스크는 'NotReady' 상태로 초기화된다. 자

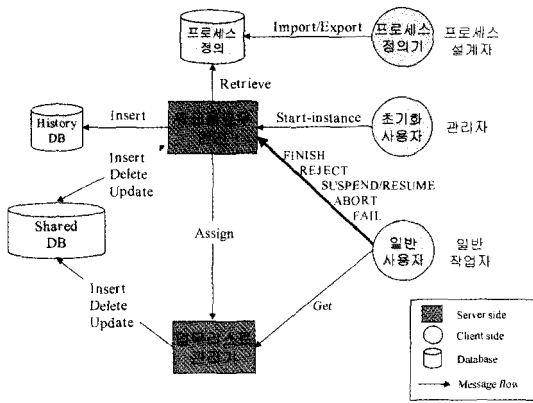


Fig. 3. Messages among system components.

신의 선행 태스크가 모두 끝나면 'Ready' 상태가 되고, 워크플로우 엔진은 'Ready' 상태의 태스크를 업무담당자에게 할당한다. 이 때 태스크의 상태는 'Doing'으로 변한다. 태스크가 끝날 때에는 리턴 코드에 따라 상태가 변한다. 태스크의 최종 상태는 'Done', 'Failed', 'Aborted' 중에 하나이다.

한편, 본 연구의 프로세스 모델은 다양한 종류의 흐름을 처리하기 위하여 다음과 같은 분기 유형을 정의하였다. 프로세스의 흐름은 직렬 분기/병합이거나 병렬 분기/병합 중의 하나이며, 병렬 분기/병합의 유형에는 다음 네 가지가 있다.

- 동시분기/병합(Concurrent split/merge): 병렬분기 태스크가 종료하면 바로 다음의 후행 태스크가 동시에 시작된다. 분기되어 진행되는 모든 태스크들이 완료해야 동시분기가 끝난다.
- 선택분기(Alternative split/merge): 선택분기 태스크가 종료하면 동시분기와 마찬가지로 모든 후행 태스크가 병렬적으로 시작된다. 이 중에 적어도 하나의 태스크가 끝나면 선택분기가 끝난다.
- 상호배제 OR분기(Exclusive-OR split/merge): 이 유형의 태스크가 종료하면 후행 태스크 중 하나만 선택되어 시작된다. 나머지 분기 방향의 태스크들은 무시된다.
- 조건분기/병합(Conditional split/merge): 조건분기 태스크가 끝나면 조건을 평가하여 그 결과에 따라 하나의 태스크가 선택된다.

4. 시스템 기능 및 구현

본 장에서는 프로세스 정의기를 이용한 프로세스

모델링, 워크플로우 엔진의 통제 절차, 그리고 웹 기반 클라이언트 사용 등 개발한 시스템의 중요한 기능을 설명하였다. 개발한 시스템은 실제 프로세스의 효율적인 관리뿐만 아니라 제품 개발이나 설계 업무에서 부가적으로 발생하는 일반 프로세스의 관리 또한 가능한 시스템으로 실제 기업 환경에서 적용 가능하다.

4.1 프로세스 정의기

프로세스 정의기는 워크플로우관리시스템에 의해 통제되고 관리되는 업무 프로세스를 GUI 환경에서 정의하고 이를 데이터베이스에 저장하는 워크플로우 설계 기능을 제공한다. 다음에서 프로세스 정의기의 세부 기능을 설명하였으며, 프로세스 정의기 프로그램 사용 예는 Fig. 4와 같다.

(1) 설계 및 편집: 대부분의 WfMS에서 제공하는 기본 기능으로 사용자는 다이어그램 윈도우를 통해 프로세스 모델을 설계하고 정의할 수 있다. 아이콘 팔레트에 있는 아이콘을 클릭함으로써 태스크를 만들 수 있고, 태스크의 속성은 팝업 윈도우를 통해 입력한다. 본 연구에서 지원하는 프로세스 모델은 WfMC에서 제안한 표준 모델을 따르고 있으며, 여기에 덧붙여 중첩 프로세스 모델을 지원하고 있다.

(2) 중첩 프로세스 설계: 프로세스 정의기에서 복잡한 프로세스는 계층적으로 모델링된다. Fig. 4의 왼쪽에 계층적 프로세스의 예가 나와 있다. 전체 프로세스의 계층적 구조는 프로세스 계층도 윈도우를 통해서 한 눈에 트리 구조로 확인 가능하다. 이 윈도우에서 중첩 태스크가 선택되면, 그 태스크에 해당되는 중첩 프로세스 다이어그램 윈도우가 활성화된다.

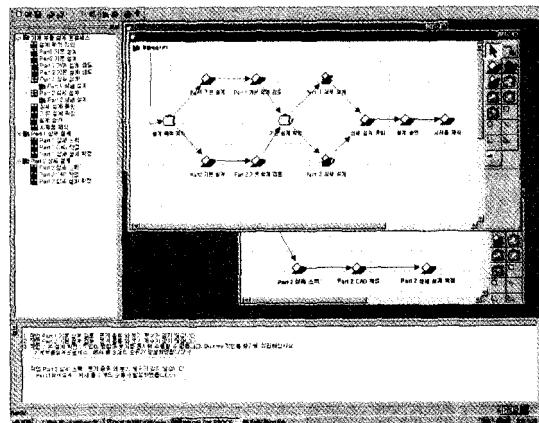


Fig. 4. Process designer.

(3) 오류 확인: 프로세스 정의기에서는 프로세스 모델에 존재하는 오류를 확인하는 기능을 제공한다. 전형적인 오류의 예에는 필수 속성의 누락, 잘못된 선후 관계, 태스크 속성과 연결 관계 사이의 불일치 등이 있다.

(4) Export 및 Import: 프로세스 모델은 데이터베이스에 저장되기 위해 Export될 수 있다. DB에 저장된 모델은 수정하거나 중첩하기 위하여 다시 Import할 수 있다.

프로세스 정의기는 Visual C++를 사용하여 구현하였으며, GUI 컴포넌트 개발을 위해 Stingray 사의 Objective Diagram을 이용하였다. 프로세스를 정의하는 작업은 프로세스의 설계자가 수행하는 작업이다. 따라서, 불특정 다수의 작업자가 시간과 공간을 초월하여 자주 접속하여 사용하는 프로그램은 아니기 때문에 독립실행형 클라이언트 프로그램으로 개발하였다.

Fig. 4에서 예제 프로세스는 기계 부품을 설계하기 위한 프로세스이다. 구성 태스크 중 'Part 1 상세 설계'와 같이 태스크 아이콘에 상자를 갖고 있는 태스크는 중첩 태스크를 의미한다. 중첩 태스크에 대한 상

세한 구성을 설계 시 정의하여 실행 할 수도 있고, 실행시간 캡슐화 기능을 통하여 실행시간에 정의하도록 할 수도 있다. 특히, 설계 프로세스에서 한 부서에서 담당하는 태스크 구성을 사전에 정의하기 어려운 경우 중첩 태스크로만 정의하여 실행시간에 해당 부서에서 상황에 맞게 정의하여 실행할 수 있다.

4.2 워크플로우 엔진

워크플로우 엔진은 워크플로우 인스턴스 실행 환경을 제공한다. 프로세스 정의를 해석하고, 프로세스 인스턴스를 생성하고 실행 및 관리하는 서버 프로그램이다. 이 외에도 프로세스 감독과 시스템 관리에 필요한 데이터를 유지, 관리한다. 이 데이터들은 사용자의 요청에 따라 클라이언트에게 제공된다. 엔진은 서블릿 API를 사용하여 자바로 구현되었고, HTTP 프로토콜을 통해 클라이언트와 요청과 응답을 주고받는다.

워크플로우 엔진에서 가장 핵심적인 모듈인 프로세스 실행 제어를 위한 흐름도는 Fig. 5와 같다. 흐름도는 5 가지 부분으로 나눌 수 있으며, 각 부분별 주요 기능을 다음에 설명하였다.

(1) 리턴 코드 처리: 엔진이 태스크의 리턴 코드

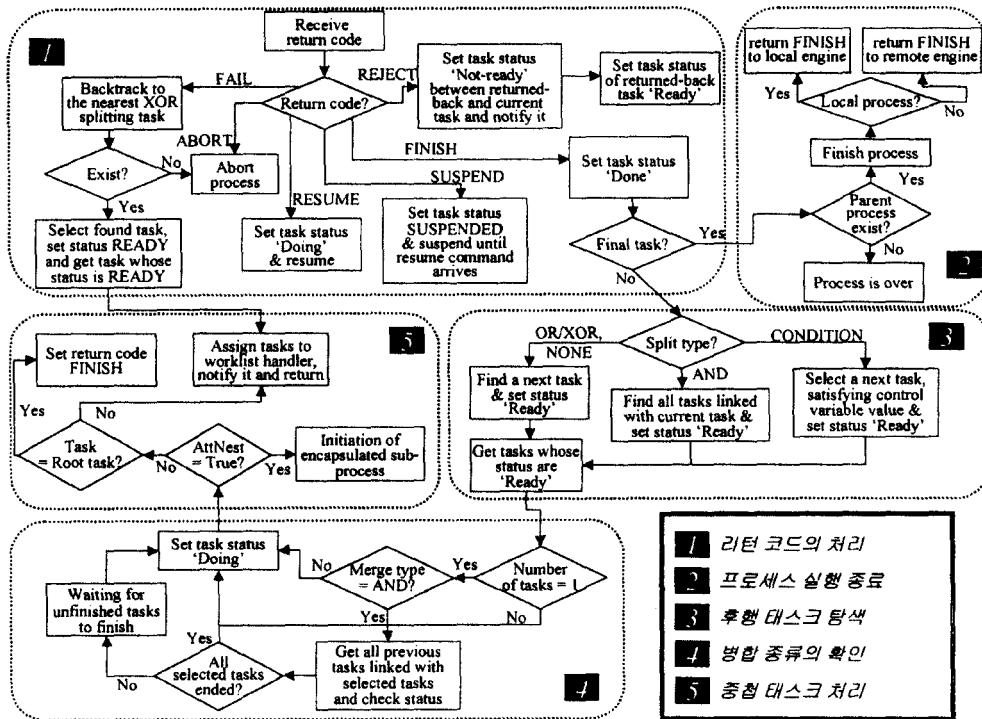


Fig. 5. Procedure of process control.

값에 따라 해당 함수를 호출한다.

(2) 프로세스 실행 종료: 프로세스의 마지막 태스크로부터 FINISH 리턴 코드가 반환되면, 그 프로세스는 종료한다. 이것은 그 프로세스의 중첩 태스크가 끝났다는 것을 의미한다. 이때, 만일 종료한 중첩 태스크가 루트 태스크이면 전체 프로세스의 실행이 완료된다.

(3) 후행 태스크 탐색: 엔진은 태스크의 분기 종류를 확인하고 나서 후행 태스크를 결정하여 그 상태를 'Ready'로 바꾼다.

(4) 병합 종류 확인: 엔진은 다음 태스크가 실행할 수 있는지 확인하기 위해서 병합 종류를 확인한다. 만일 태스크의 병합 종류가 동시병합이라면 그 태스크는 선행 태스크들이 모두 끝나야 실행 가능하다.

(5) 중첩 태스크 처리: 실행 대상 태스크의 중첩 여부에 따라 처리하는 부분이다. 중첩 태스크의 경우¹⁶⁾에 상세히 설명된 중첩 태스크 처리를 위한 알고리즘을 따르면 된다. 만일 태스크가 중첩 태스크가 아니라 원시 태스크인 경우 엔진은 태스크의 담당자를 결정한다. 담당자가 역할로 정의되어 여러 명인 경우 부하 균형화 절차에 따라 담당자를 결정한다. 이 때 태스크의 상태가 'Doing'으로 바뀌고, 태스크는 업무담당자에게 할당된다. 동시에 엔진은 자바 메일 API를 사용하여 E-mail을 발송함으로써 작업의 할당 사실을 알려 준다. 이는 간단하면서도 효과적으로 작업을 통보하는 방법이다.

4.3 웹 기반 클라이언트

클라이언트 모듈은 초기화 사용자 클라이언트, 일반 클라이언트, 모니터링 모듈, 시스템 관리 모듈의 네 가지로 분류된다. 웹 기반 클라이언트 프로그램은 자바 애플릿으로 구현하였다.

(1) 초기화 사용자: 이 모듈을 통해 프로세스 모델의 인스턴스를 생성한다. 워크플로우 사용자는 데이터베이스로부터 프로세스 모델을 선택한 후 기존에 저장된 모델 값을 변경할 수도 있고, 생성한 인스턴스에 국한된 값을 입력할 수도 있다.

(2) 일반 사용자 클라이언트: 일반 클라이언트 인터페이스를 통해 업무담당자는 엔진으로부터 태스크 정보를 받아서 수행하고, 엔진으로 그 결과를 반환한다. Fig. 6에 이를 위한 인터페이스 화면 예가 나와 있다. 업무담당자가 수행해야 하는 태스크는 그림의 가운데에 있는 현재 업무 탭에 나타나 있다. 그 오른쪽에 있는 버튼을 사용하여 최신정보로 고치거나, 현재 선택된 프로세스의 전체 프로세스에 대한

정보를 열람하거나, 태스크에 대한 상세 정보를 확인할 수 있다. 예를 들면, '작업흐름도' 버튼을 누르면 그림의 아래 쪽에 보이는 프로세스 흐름 윈도우가 뜬다. 사용자는 '첨부화일' 버튼을 눌러서 설계 프로세스에서 사용되는 도면이나 관련 문서들을 관리하는 웹 기반 문서관리시스템을 사용할 수 있다. '프로세스명' 필드는 현재 태스크를 포함하는 프로세스의 이름을 나타내고 있다. 엔진으로 결과를 반환하기 전에 사용자는 다음 업무담당자에게 '업무 메모' 필드에 메시지를 입력하여 전달할 수 있다. 오른쪽 아래에 있는 버튼을 이용하여 엔진에게 리턴 코드를 반환한다. '완료 업무 모니터링' 탭을 통해 자신이 완료한 태스크의 진행 상황을 모니터링할 수 있다.

(3) 감독 통제 및 모니터링: 감독 관리자는 이 모듈을 이용하여 현재 실행 중인 프로세스의 상태를 모니터링하거나 독촉, 실행취소, 일시중지, 재개 등의 명령을 내릴 수 있다. 또한, 종료한 프로세스에 대한 통계 정보인 전체 수행도, 업무담당자의 태스크 수행 이력, 업무 부하 등을 열람할 수 있다. 모니터링 기능은 권한이 있는 사용자가 이용할 수 있다. zoom/in/out 기능이 제공되므로 상위 수준의 관리자는 자신이 관심 있는 전체 프로세스에 대한 축약된 정보를 제공받을 수 있다. 반면, 하위 수준의 관리자는 서브프로세스에 대한 상세한 정보를 필요로 할 수도 있다.

(4) 시스템 관리: 시스템 관리 모듈을 통해 사용자, 역할, 권한 등 본 연구에서 개발한 시스템을 운용하는데 필요한 환경을 설정한다.

Fig. 6의 왼쪽 메뉴 버튼을 누르면 해당되는 클라

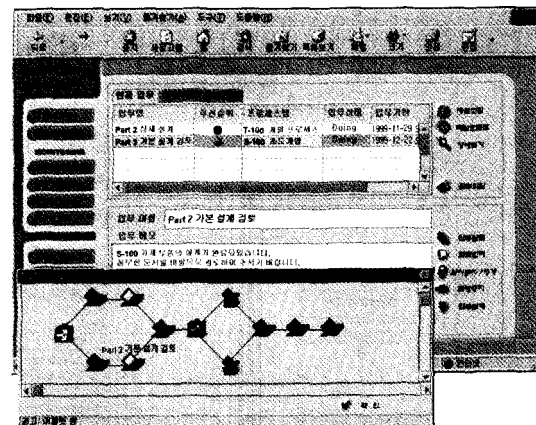


Fig. 6. User interface for normal clients.

이전 기능을 이용할 수 있다. 사용자는 자신의 권한에 따라 이용할 수 있는 기능에 제한을 받는다. 예를 들면, 모니터링 권한이 있는 사용자는 그림의 예제에서 보는 바와 같이 기계부품 설계 프로세스에 대한 진행 상태를 파악하여 현재 상황에 맞는 각종 감독 명령을 내릴 수 있다. 실제 현장에서 프로세스를 자동화할 때 가장 필요한 기능이 바로 상황을 파악하여 적절한 조치를 취하는 기능으로 본 연구에서 개발할 시스템은 진행상태에 맞게 다양한 명령을 내릴 수 있도록 하고 있다.

5. 결론 및 추후 연구과제

설계 프로세스를 관리하기 위한 웹 기반 워크플로우 관리시스템의 설계와 개발에 대하여 설명하였다. 워크플로우 관리시스템이 제공하는 여러 장점 때문에 PDM 같은 많은 기업 정보 시스템에서 핵심 모듈로 간주되고 있다. 본 연구의 목적은 분산 이질 환경에서 설계 업무 프로세스를 효과적으로 관리하는 시스템을 제공하는 것이다. 이 환경에서의 업무 관리에 요구되는 사항을 규명하고, 관련 연구들을 조사하였다. 엔진은 순수 자바로 구현하였으므로 플랫폼 독립적으로 설치 가능하다. 클라이언트 모플은 자바 애플릿으로 구현하였으므로 위치투명성 요구사항을 충족한다.

이질 전산 환경에서 이기종 워크플로우 시스템 사이에 완전한 상호운용성을 이룩하기 위해서는 프로세스 정의의 표준이 확립되는 것이 매우 중요하다. 상호운용성을 달성하기 위해 모든 워크플로우 엔진에서 공통의 API 집합을 지원해야 한다. 대안의 하나로 최근 제안된 XML로 된 SWAP 형식의 프로세스 정의를 이용하는 것이 있다. 또 다른 추후 연구 주제로 ERP, PDM, SCM, CRM, EC 등과 같은 정보 시스템의 통합을 위해 WfMS를 사용하는 것이 있다. 이러한 통합을 위해서 에이전트 기술¹⁹⁾이 효과적으로 도입될 수 있을 것이다.

감사의 글

본 연구는 한국과학재단 특정기초연구비(97-02-00-09-01-3) 지원으로 수행되었으며 지원에 감사드립니다.

참고문헌

1. Action Technologies, Inc., "ActionWorks Metro: e-Pro-

- cess Application Platform," <http://www.actiontech.com/Products/Metro/Overview/>, 1999.
2. A. Kumar and J. L. Zhao, "Dynamic Routing and Operational Controls in Workflow Management Systems," *Management Science*, Vol. 45, No. 2, pp. 253-272, Feb. 1999.
3. C. Petrie, S. Goldmann, and A. Raquet, "Agent-based Process Management," *Proc. of the Int'l Workshop on Intelligent Agents in CSCW*, Deutsche Telekom, Dortmund, pp. 1-17, Sep. 1998.
4. D. Hollingsworth, *Workflow Management Coalition Specification: The Workflow Reference Model*, WfMC specification, Jan. 1995.
5. ETRI, *Development of a Workflow Management System for CALS data*, Project Report, ETRI, Taejon, S. Korea, 1999.
6. F. Casati, S. Ceri, B. Pernici, G. Pozzi, "Deriving Active Rules for Workflow Enactment," *Proc. of 7th Int'l Conf. on Database and Expert Systems Applications*, Springer-Verlag Lecture Notes in Computer Science, pp. 94-110, 1996.
7. Fujitsu, "i-Flow Architectural White Paper," http://www.i-flow.com/about_iflow/, 1999.
8. G. A. Bolcer and G. Kaiser, "SWAP: Leveraging the Web to Manage Workflow," *IEEE Internet Computing*, Vol. 3, No. 1, pp. 85-88, 1999.
9. InConcert, Inc., "Enterprise Business Integration," <http://www.inconcert.com/products/inconcert2000/>, 1999.
10. J. A. Miller, D. Palaniswami, A. P. Sheth, K. J. Kochut, and H. Singh, "WebWork: METEOR2's Web-based Workflow Management System," Technical Report, University of Georgia, April 1997.
11. J. D. Davidson, *JavaTM Servlet API Specification*, v2.2, Sun Microsystems, Inc., Nov. 1998.
12. J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate, and G. Yost, *The PIF Process Interchange Format and Framework*, PIF Working Group, May 1996.
13. Staffware PLC, "Staffware Functionality," white paper, <http://www.staffware.com/home/products/2000WhitePaper.zip>.
14. T. Cai, P. A. Gloor, and S. Nog, "DartFlow: A Workflow Management System on the Web using Transportable Agents," Technical Report PCS-TR96-283 of Dartmouth College, May 1996.
15. Ultimus, "Ultimus Workflow Suite," <http://www.ultimus.com/release40/dld/ultbrief.pdf>, 1999.
16. Yeongho Kim, Suk-Ho Kang, Dongsoo Kim, and Joonsoo Bae, "WW-flow: A Web-based Workflow Management System Supporting Nested Processes," submitted to *IEEE Internet Computing*.
17. 김영호, 강석호, 이수홍, 유상봉, "분산, 개방, 지능형 제품정보관리시스템", 한국 CAD/CAM 학회 논문집, 제 4권, 제 3호, pp. 210-223, 1999.



김영호

1985년 서울대학교 산업공학과 학사
 1987년 서울대학교 산업공학과 석사
 1988년-1990 한국과학기술연구원 연구원
 1993년 North Carolina 주립대 산업공학과
 공학박사
 1995년-현재 서울대학교 산업공학과 교수
 관심분야: PDM, 동시공학, Workflow, 인터넷
 응용



김동수

1994년 서울대학교 산업공학과 학사
 1996년 서울대학교 산업공학과 공학석사
 1996년 현재 서울대학교 산업공학과 박사
 과정
 관심분야: Workflow, Agent Technology,
 PDM



주경준

1974년 고려대학교 산업공학과 학사
 1976년 고려대학교 산업공학과 공학석사
 1978년 한국과학기술연구소 연구원
 1991년 삼성전자 정보통신부분
 1991년-현재 한국전자통신연구원 컴퓨터 소
 프트웨어 기술연구소 전자상거래
 연구부 CALS 연구팀장
 관심분야: 기업간 전자거래기술(B to B
 EC, SCM, CALS)



강석호

1970년 서울대학교 물리학과 학사
 1972년 미국 Univ. of Washington 산업공
 학과 석사
 1976년 미국 Texas A & M Univ. 산업
 공학과 교수
 1976년-현재 서울대학교 산업공학과 교수
 관심분야: 생산정보시스템의 설계와 운용, 저
 능형 생산 시스템, 경영정보 시스
 템, 생산 계획 및 통제