

# InfoFlow: A Web-based Workflow Management System

Yeongho Kim\*, Suk-Ho Kang\*, Dong-Soo Kim\*, Won-Chang Heo\*, Young-Myoung Ko\*,  
Sang-Jin Lee, Kyoung-Jun Joo \*\*

## Abstract

In this paper, we introduce the design and development of a web-based workflow management system. The goal of the developed system is to manage business processes occurring in the CITIS (Contractor Integrated Technical Information Services) environment. The system is composed of three main modules: Process Designer, Workflow Engine, and Client modules. The Process Designer is a module that provides the environment for the build-time function, which generates the specifications of processes. The module presents the capability of defining nested process models, which is powerful in particular for designing complex processes. Since the other two modules are implemented using pure Java technology, the Workflow Engine can be implemented on any platform and the Client programs can be accessed via the WWW interface. This indicates that there is no need to install any client programs at the client-sides. Users who has a connection to the internet with web browsers, such as Internet Explorer and Netscape Navigator, and has a proper right of access can utilize the normal client, monitoring client, and system administration client programs. Communications between the workflow engine and the clients are implemented using the java servlet mechanism. The workflow system can serve as the underlying platform of process management tool in CALS and CITIS environments. An example scenario of using the system is presented.

## 1. 서론

본 연구에서는 CALS 규약의 일종인 CITIS (Contractor Integrated Technical Information Services) 환경에서 발생하는 비즈니스 프로세스를 관리하기 위한 웹 기반 워크플로우 관리 시스템(InfoFlow System)을 설계하고, 구현하였다. 컴퓨터

를 이용하여 자동적으로 수행되는 비즈니스 프로세스를 워크플로우라 하며, 워크플로우 관리 시스템이란 워크플로우 로직을 컴퓨터로 표현하고 이 표현에 따라 생성된 순서를 따르는 소프트웨어 실행을 통해 워크플로우를 완전하게 정의, 관리, 실행하는 시스템을 의미한다[4]. 즉, 워크플로우 관리 시스템은 업무 흐름의

---

\* Department of IE, Seoul National University

\*\* ETRI

자동화, 정보 및 문서 전달의 전자화, 그리고 일관적 데이터 접근 및 제어를 통해 업무 프로세스의 개선, 통제, 관리, 공동작업을 지원하는 소프트웨어로 최근 많은 기업에서 주목 받고 있다.

최근에 인터넷 사용 인구가 급속도로 증가함에 따라 많은 정보 시스템들이 웹 인터페이스를 제공하고 있다. 워크플로우 관리 시스템도 마찬가지로 기존의 많은 상용 워크플로우 관리 시스템 공급업체들이 웹기반 시스템들을 개발하여 제공하고 있다. 대표적인 웹기반 상용 시스템에는 Ultimus 사의 Ultimus, Fujitsu 사의 i-Flow, Action Tech.의 Metro 등이 있다.

본 연구에서는 순수 자바 기술을 이용하여 InfoFlow 시스템을 개발하였다. InfoFlow 시스템을 이용하면, 웹 브라우저를 통해 인터넷에 접속할 수 있는 모든 작업자 및 시스템 구성원들이 워크플로우 시스템에 참여할 수 있다. 본 연구에서는 통신 하부구조로 CORBA 등의 미들웨어를 이용하지 않고, 순수 자바 기술인 서블릿을 이용하였으므로, 웹기반 클라이언트와 서블릿 엔진 사이의 통신은 HTTP 프로토콜로 통신이 이루어진다.

## 2. InfoFlow 시스템의 장점

본 연구에서 개발한 시스템은 웹 브라우저를 통해 인터넷에 접속 가능한 작업자들이 워크플로우 시스템에 참여할 수 있는 순수 자바 기술을 이용한 시스템이다. 본 시스템에 참여하기 위해서 클라이언트 프로그램을 따로 설치할 필요가 없고, 단지 웹 브라우저를 통한 인터넷 접속만 하면 된다. 즉, 순수 자바 기술로 구현하였으므로 기존의 많은 워크플로우 시스

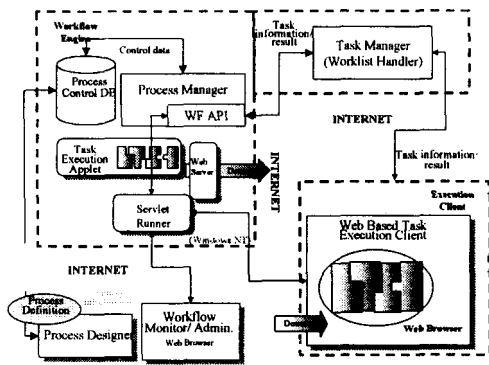
템에서 클라이언트 프로그램을 설치하거나 CORBA 등의 미들웨어를 설치해야 했던 번거로움을 피할 수 있다.

과거에 많은 정보 시스템들에서 클라이언트 서버 구조로 구현했을 때, 별도의 클라이언트 프로그램을 설치하는 것이 일반적이었으나, 최근 인터넷의 대중화와 웹 사용의 급격한 증가에 힘입어 주요 정보 시스템인 ERP(Enterprise Resource Planning), PDM(Product Data Management), EDM (Electronic Data Management) 시스템들이 웹기반의 클라이언트 인터페이스를 제공하고 있다. 본 연구에서는 이러한 추세를 반영하여 웹기반 워크플로우 시스템을 구축함으로써, 시간과 장소에 구속되지 않고 웹 브라우저를 통한 클라이언트의 시스템 참여를 가능하도록 하였다.

본 시스템이 갖는 장점은 다음과 같이 요약될 수 있다. 첫째, 별도의 클라이언트 프로그램의 설치가 필요 없으므로 시간과 장소에 상관없이 웹 브라우저가 있는 곳에서는 시스템에 참여할 수 있다. 둘째, 웹 브라우저를 통한 시스템 참여가 가능하므로 많은 사용자들이 워크플로우 시스템을 쉽게 수용하여 작업할 수 있다. 따라서, 사용자들이 워크플로우 시스템을 보다 적극적으로 이용할 수 있다. 셋째, 웹 기술인 서블릿을 이용함으로써 통신 프로토콜을 HTTP 하나로 통일할 수 있다. 넷째, 순수 자바 기술을 이용하였기 때문에 클라이언트 프로그램의 설치 및 변경 시 배포 작업이 필요 없다는 점이다. 클라이언트 프로그램의 설치가 필요 없으며, 변경 사항이 발생하는 경우 서버 쪽의 프로그램만 변경하면 되기 때문에 클라이언트는 별도의 배포 및 재설치가 필요 없다.

### 3. InfoFlow 시스템 구조

본 연구에서 개발한 웹기반 워크플로우 시스템의 전체적인 구조는 다음 [그림 1]과 같다.



[그림 1] InfoFlow 시스템 전체 구조

[그림 1]에서 보는 바와 같이 본 시스템은 크게 프로세스 디자이너, 워크플로우 엔진, 워크플로우 참여 클라이언트, 시스템 관리자 및 모니터링 클라이언트로 구성된다. 엔진은 NT 서버에서 구동되고 있으며, 프로세스 통제를 위한 데이터베이스는 엔진이 워크플로우를 구동하는데 필요한 통제 데이터와 관련 데이터를 저장하여 관리한다. 엔진 쪽에 있는 태스크 관리자 모듈은 클라이언트에게 할당되는 작업 리스트를 관리한다. 클라이언트와 엔진 사이의 통신 메커니즘으로는 HTTP 프로토콜을 사용하는 서블릿을 이용하였다. 프로세스 디자이너를 제외한 각종 클라이언트 모듈들은 자바 애플릿으로 구현하였다. 프로세스 디자이너는 자동화하고자 하는 비즈니스 프로세스를 정의하는 모듈로서 독립 실행형 클라이언트이다. 또한, 프로세스 디자이너는 설계된 프로세스 정의를 엔진이 이해할 수 있는 형태로 포맷팅하여 프로세스 정의 DB에

넣는 기능을 수행한다.

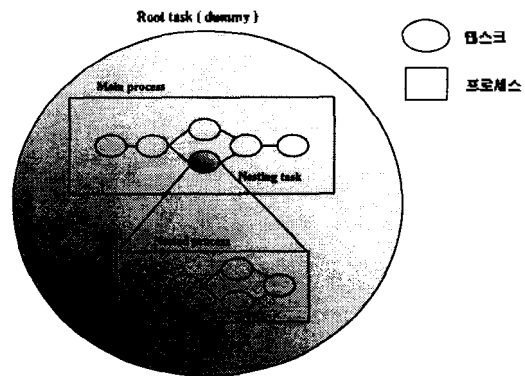
### 4. 워크플로우 엔진과 프로세스 디자이너

본 장에서는 InfoFlow 시스템의 프로세스 중첩 설계 지원을 위한 방안과 워크플로우 관리시스템의 핵심 모듈이라 할 수 있는 엔진과 프로세스 디자인의 기능과 구현 사항에 대하여 설명하였다.

#### 4.1 프로세스의 중첩 지원

본 연구에서는 중첩 설계를 지원하기 위하여 중첩의 기본이 되는 태스크가 프로세스를 포함하고 있는 방식을 취하고 있다.

본 연구에서 사용할 중첩된 프로세스의 구조를 살펴보면 [그림 2]와 같다.



[그림 2] 중첩된 프로세스의 구조 (Recursive)

#### (1) 태스크

중첩 설계 시 태스크는 프로세스를 포함하면서 또한 자기 자신은 다른 프로세스에 종속이 된다. 다시 말하면 태스크는 프로세스의 자식이 될 수도 있고 부모가 될 수도 있으므로 기존의 프로세스의 하

부로서 존재하는 것이 아닌 프로세스와 같은 수준에서 정의된다.

### (2) 프로세스

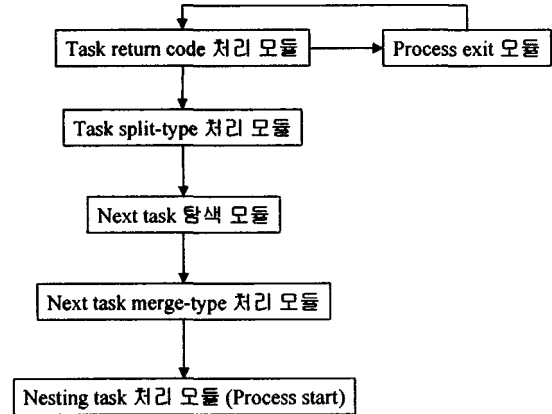
중첩 설계 시 프로세스는 태스크를 포함하고 있지만 또한 자기 자신이 다른 태스크에 종속이 될 수도 있다. 이러한 경우 메인 프로세스와 서브 프로세스를 구분하는 것은 의미가 없다. 왜냐하면 중첩이 여러 단계에 걸쳐서 일어날 경우 미들 레이어에 위치한 프로세스는 상위 프로세스에서 보기에 서브 프로세스이지만 하위 프로세스 입장에서 보기에 메인 프로세스이기 때문이다. 따라서 프로세스를 정의할 때 메인 프로세스와 서브 프로세스를 구분하지 않고 태스크와 프로세스와의 관계를 통해 구분을 하도록 하였다.

### (3) 작동 원리

[그림 2]에서 보는 바와 같이 최상위 프로세스 상위에 이 프로세스를 포함하는 Root task 를 정의한다. 프로세스 인스턴스가 시작되면 이 Root task 가 작동하게 되고, Root task 는 자기 하부에 존재하는 프로세스를 실행시킨다. 프로세스를 수행하던 중에 프로세스 중첩하고 있는 태스크를 만나면 이 태스크는 자기 하부의 프로세스를 시작시키는 역할을 하고, 그 프로세스 인스턴스가 종료하면 수행된 결과를 리턴 받는다. 최종적으로 메인 프로세스가 끝나면 그 결과값을 Root task 에 돌려주게 되고 Root task 의 경우에는 다음 태스크가 없으므로 종료된다. 이러한 방식으로 진행하게 될 경우 중첩이 여러 단계에 걸쳐서 일어날 경우에도 일관성을 잃지 않고 작동을 할 수 있다.

## 4.2 엔진 구성 모듈

본 연구에서 개발한 워크플로우 엔진의 구성 모듈은 [그림 2]와 같다.



[그림 3] 엔진 구성 모듈

각 모듈에 대한 설명을 아래에 상세히 서술하였다.

#### (1) Task return code 처리 모듈

Task return code 처리 모듈에서는 할당된 작업이 서버로 결과가 돌아올 때에는 완료, 실패, 반려, 일시중지/작업재개 등의 return code 를 전달하게 된다. 본 모듈에서는 이러한 return code 를 전달 받아 각 return code 에 해당되는 함수를 호출한다.

#### (2) Task 분기 종류(split-type) 처리 모듈

다음 작업을 찾기 전에 우선 그 작업의 분기 종류를 확인 한다. 분기 종류에는 AND, OR, XOR, CONDITION 분기가 있으며 각 분기에 해당되는 Next task 탐색 모듈의 함수를 호출한다.

#### (3) Next task 탐색 모듈

태스크 분기 종류 처리 모듈에서 호출된다. 각 분기 종류 별로 정해진 함수가 있

고, 각 함수별로 정해진 루틴에 따라 다음 작업을 얻어 온다.

#### (4) Next task merge-type 처리 모듈

다음 작업으로 결정된 작업의 merge type 을 확인한다. Split type 과 일대일 대응이 되는 AND, OR, XOR, CONDITION 병합이 있으며 각 merge type 별로 해당되는 함수를 호출한다.

#### (5) Nesting task 처리 모듈(Process start)

다음 작업으로 결정된 작업의 중첩 여부를 확인 한다. 작업의 중첩여부를 나타내는 애트리뷰트가 참일 경우 그 작업이 중첩하고 있는 프로세스를 시작 시킨다. 처음 프로세스가 시작할 경우에는 중첩 처리의 일관성을 위해 메인 프로세스를 중첩하고 있는 Root task 가 initiator 에 의해 들어 오고, 이에 따라 메인 프로세스가 시작된다.

#### (6) Process exit 모듈

작업이 마지막 작업일 경우 해당 프로세스를 종료하고 그 프로세스를 중첩하고 있는 작업의 return code 를 완료로 바꾼 후 태스크 return code 처리 모듈로 보낸다.

### 4.3 디자이너 구현 및 사용법

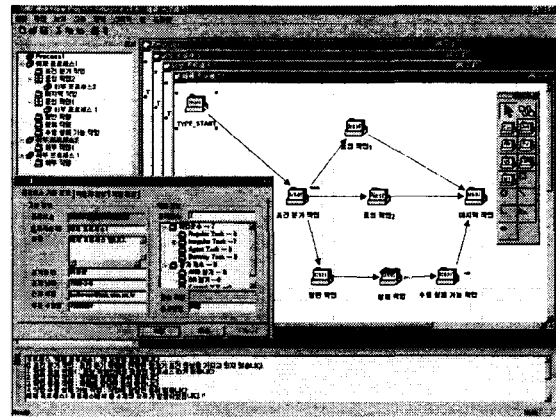
프로세스 디자이너는 워크플로우 관리 시스템에 의해 통제되고 관리되는 업무 프로세스를 GUI (Graphical User Interface) 환경에서 정의하고 이를 데이터베이스에 저장하여 수행하는 워크플로우 설계 기능(Build-Time Function)을 제공하는 모듈이다.

본 연구에서 개발한 프로세스 디자이너는 링크로 연결된 작업 노드로 표현되는 다이어그램 형태로 업무 프로세스를 설계할 수 있는 환경을 제공하며, 각 작업의 속성, 프로세스의 속성, 분기 조건 등의 데이터를 입력할 수 있는 기능을 제공한다. 또한, 중첩 프로세스의 계층 구조를 쉽게 조감하고 편집할 수 있는 환경과, 설계된 프로세스의 오류를 검증하고 이를 데이터베이스에 입출력 할 수 있는 인터페이스 기능도 제공한다.

디자이너 프로그램은 Visual C++로 구현하였으며, GUI 개발을 위해서 Stingray 사의 Objective Diagram 라이브러리를 이용하였다.

#### 4.3.1 프로세스 디자이너의 화면 구조

본 프로세스 디자이너의 초기 실행 화면은 [그림 4]와 같다.



[그림 4] 프로세스 디자이너 화면

프로세스 디자이너는 다이어그램 뷰와 작업 팔레트, 프로세스 브라우저와 메시지 창, 그리고 속성을 입력하는 각종 대화 상자를 사용한 설계 환경을 제공한다. [그림 4]의 다이어그램 뷰와 작업 팔레트는 링크로 연결된 작업 노드의 다이어그램

램 형태로 업무 프로세스를 정의할 수 있도록 해 주며, 각각의 작업 노드와 링크에는 작업 속성과 분기 속성, 조건 속성 등을 입력할 수 있는 대화상자를 제공한다. 프로세스 브라우저에서는 중첩된 업무 프로세스의 계층 구조를 쉽게 조감하고 편집할 수 있도록 해 주며, 이는 다이어그램 뷰의 내용과 일관성 있게 연동되어 작동한다. 또한 하단의 메시지 창에는 프로세스의 오류 검증 결과를 제공함으로써 설계 시 발생하는 설계 오류를 방지해 준다.

#### 4.3.2 프로세스 디자이너의 기능

프로세스 디자이너의 세부 기능은 다음과 같다.

##### (1) 프로세스 다이어그램의 설계 및 편집 기능

이 기능은 프로세스 디자이너의 기본 기능으로써 설계자는 작업 팔레트와 다이어그램 뷰가 제공하는 환경을 통해 업무 프로세스의 형태를 설계해 나가게 된다. 각 작업과 링크를 선택하면 작업 속성 및 분기 조건과 같은 데이터를 입력할 수 있다.

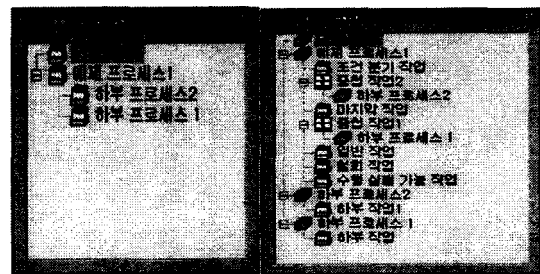
또한, 부가적 기능으로 설계자의 편의를 제공하기 위해 다이어그램의 확대 및 축소, 눈금 격자의 설정 및 조절, 각 작업 아이콘들의 그룹화와 Cut & Paste, 작업 아이콘들의 정렬, 색깔 및 폰트 지정 등의 다양한 세부 기능을 제공한다.

##### (2) 중첩 프로세스의 설계 및 편집 기능

[그림 5]는 프로세스 브라우저를 통해 중첩 프로세스의 계층구조를 나타낸 모습이다. 본 프로세스 디자이너는 이와 같이

트리 형태의 계층구조를 갖는 프로세스를 설계할 수 있으며, 이는 프로세스 브라우저를 통해 조감할 수 있다.

프로세스 브라우저는 설계의 편의를 위하여 [그림 5]와 같이 두 가지 형태의 트리를 제공한다. 좌측의 트리는 프로세스에 포함된 작업들과 중첩되어 있는 프로세스들을 한 단계로 제공하고 다이어그램 뷰의 설계 환경과 연동시킴으로써 많은 프로세스를 한 번에 설계할 경우에 효율적으로 사용될 수 있다. 우측의 트리는 각각의 프로세스의 중첩 구조를 최하위 수준까지 보여 줌으로써 프로세스간의 중첩 구조를 한번에 확인할 수 있도록 한다.



[그림 5] 프로세스 브라우저와 계층구조

##### (3) 프로세스 정보의 오류 검증

설계된 프로세스가 엔진을 통해 정확하게 구동 되기 위해서는 설계 시 발생할 수 있는 모든 오류가 제거되어야 한다. 프로세스 디자이너에서는 설계된 프로세스를 데이터베이스에 입출력 하기 전에 설계의 검증 과정을 거침으로써 잘못된 프로세스 설계로 인한 워크플로우 엔진의 오동작을 미연에 방지하고 있다. 검증 과정을 통해 다음과 같은 종류의 오류들이 감지된다.

- 기본 속성의 누락 - 작업 ID, 작업 이름, 담당 역할, 분기, 병합 속성 등

의 누락 감지.

- 작업간의 연결 관계에서의 오류 - 연결되지 않은 링크와 작업 노드 감지, 분기와 병합의 대칭성 검증.
- 작업 속성과 연결 관계의 불일치 - 작업의 분기, 병합 속성과 실제 연결 구조의 일관성 확인.

#### (4) 프로세스 정보의 데이터베이스 입출력

기존의 프로세스 정보는 새롭게 갱신하거나 다른 프로세스의 하부 프로세스로 재사용하기 위해서 데이터베이스로부터 입출력할 수 있다.

### 5. 웹 인터페이스 제공 클라이언트

InfoFlow 시스템의 웹 인터페이스 클라이언트 모듈은 초기화 사용자(Initiator) 클라이언트, 일반 클라이언트, 모니터링 모듈, 시스템 관리 모듈의 네 가지로 분류된다.

이들 클라이언트 모듈을 구현하기 위해서, 서블릿 구현에는 Sun 사의 JSDK2.0 을 사용하였고, DBMS 는 관계형 데이터 베이스인 오라클을 이용하였다. 오라클사의 JDBC thin 드라이버를 통해 자바 엔진에서 DB 로 연결하도록 하였다. 웹 인터페이스를 제공하는 각종 클라이언트 애플릿은 볼랜드 사의 JBuilder 2.0 을 사용하여 구현하였고, 개발에 사용된 JDK 버전은 JDK1.2 이다.

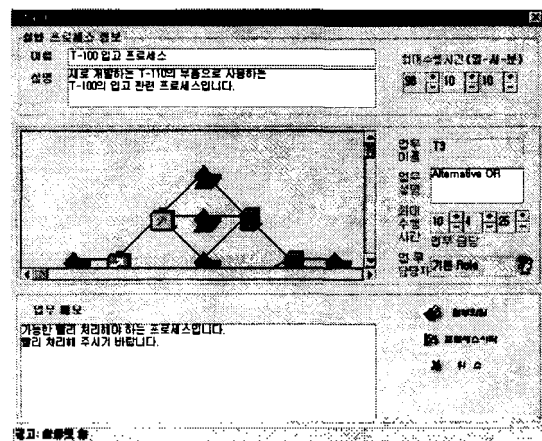
#### 5.1 Initiator, 일반 클라이언트 구현 및 사용법

##### 5.1.1. Initiator

Initiator 클라이언트는 InfoFlow 시스템이 관리하는 프로세스들 중 하나를 선택하여 해당 프로세스의 인스턴스를 생성시키는 애플리케이션이다. 이 때 인스턴스 별로 하위 작업에 대한 정보 및 관련 데이터를 입력하는 기능을 제공한다.

#### (1) Initiator 의 화면 구조 및 기능 설명

Initiator 클라이언트의 실행 화면은 [그림 6]과 같다. Initiator 는 프로세스 정의를 선택하는 것과 선택된 프로세스에 관련 데이터를 입력하는 두 부분으로 구성되는데 [그림 6]은 후자를 나타내는 화면이다. 현재 워크플로우 관리 시스템에서 저장되어 있는 프로세스 중 하나를 선택한 후, 이 인스턴스의 이름과 설명 입력, 각 하위 작업들의 담당자를 선정한다. 프로세스 디자이너에서 각 작업마다 담당 역할을 정의하기 때문에, Initiator 모듈에서 담당자 선정이 없으면 엔진이 부하 균형화(load balancing)를 통해 담당 작업자를 선정한다. 작업 기한을 의미하는 최대 수행시간의 변경도 가능하며, 사용자의 입력이 없는 경우 프로세스 정의에서 정의된 값을 그대로 이용한다.



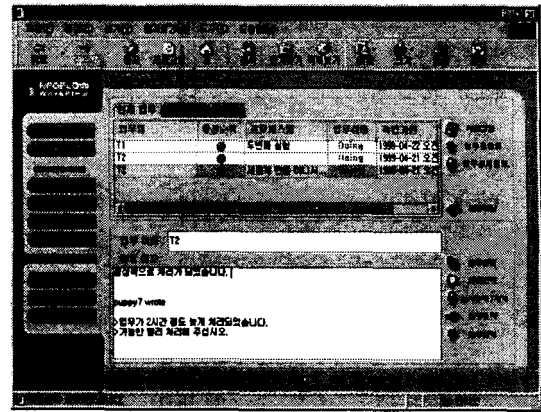
[그림 6] Initiator 클라이언트의 화면 구조

업무 흐름도를 통해 각 작업들의 최대 수행시간을 변경할 수 있으며, 담당자 선정의 경우 전체 업무 흐름을 고려한 입력이 가능하다.

### 5.1.2 일반 클라이언트

일반 클라이언트는 워크플로우 엔진과의 상호작용을 통해 일반 사용자가 워크플로우 관리 시스템을 이용하여 실제 업무를 처리할 수 있도록 하는 애플리케이션이다. 즉, 사용자에게 할당된 업무리스트를 관리하며, 현재 진행하는 각 작업에 대한 정보 및 관련 데이터를 저장하고, 처리된 작업의 결과를 서버로 전달하는 기능을 수행한다. 본 연구에서는 웹기반의 환경에서 위의 기능을 제공하는 일반 클라이언트의 개발을 통해, 사용자가 워크플로우 관리 시스템과의 상호작용을 통해 실제 업무를 처리할 수 있도록 하고 있다.

일반 클라이언트의 실행 화면은 [그림 7]과 같다. 일반 클라이언트의 화면은 탭과 메모 입력의 두 부분으로 구성되어 있다. 즉, 현재 사용자에게 할당된 작업리스트를 보여주는 테이블(현재 업무)과 사용자가 완료한 작업 중 그 작업이 속한 프로세스가 아직 진행되고 있는 프로세스인 경우 그 진행사항을 보여주는 테이블(완료 업무 Monitor)로 구성된 탭과 다음 작업자에게 전달할 업무 내용을 기록하는 메모 입력 부분으로 이루어져 있다. 또한, 오른쪽의 각 버튼을 눌렀을 때 선택된 각각의 현재 업무에 대해서 상세 정보를 보거나 이 업무가 포함된 프로세스 전체의 흐름도를 볼 수 있게 되어 있다.



[그림 7] 일반 클라이언트의 화면 구조

현재 작업자에게 할당된 작업이 끝났을 경우에 사용자는 화면 오른쪽 아래의 버튼을 통하여 그 결과를 서버로 전달할 수 있다. 즉, 사용자는 성공적으로 작업을 끝냈을 경우에는 업무 완료 버튼을, 이전 작업자에게 업무를 다시 돌려보내는 경우에는 업무반려 버튼을, 작업을 잠시 보류하거나 보류된 작업을 다시 시작할 경우에 일시중지/재개 버튼을, 선택된 작업이 속한 프로세스를 강제 종료(abort)시키는 경우에는 실행중지 버튼, 그리고 선택된 작업을 수행할 수 없을 때는 업무실패 버튼을 사용하여 워크플로우 엔진으로 그 결과를 전할 수 있다. 업무반려 버튼을 눌렀을 경우에는 이전 작업자들 중 어느 작업자에게 보낼 지를 업무 흐름도를 이용하여 쉽게 선택할 수 있도록 하였다.

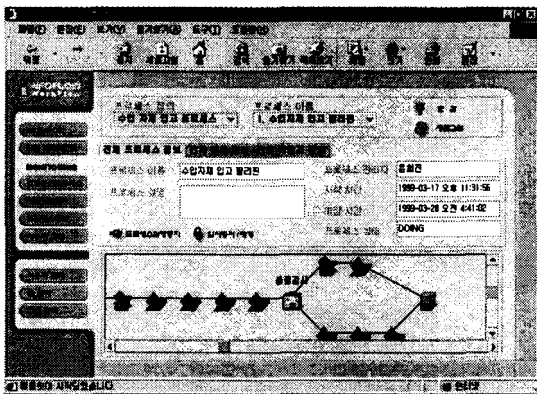
5.2 모니터링 모듈 및 시스템 관리 모듈  
모니터링 모듈은 현재 진행 중인 프로세스의 상태 감시와 로그에 기록된 이력 데이터를 열람하고, 통제 행동을 수행하는 기능을 제공한다. 시스템 관리 모듈은 사용자 관리, 역할 관리, 그룹 관리 등의 각종 관리자 기능을 제공한다.



### 5.2.1 모니터링 클라이언트

#### (1) 진행중 프로세스 모니터링

진행중 프로세스 모니터링 클라이언트는 워크플로우 관리 시스템 내에서 현재 진행되고 있는 프로세스 인스턴스들의 정보를 제공하고 여러 관리 기능을 제공함으로써 관리자가 효율적으로 시스템을 관리할 수 있도록 한다.



[그림 8] 진행중 프로세스 모니터링

[그림 8]은 현재 진행중인 프로세스를 모니터링 하는 화면으로 수입자재 입고 프로세스의 현재 진행상태를 보여 주고 있다. 진행중 프로세스 모니터링 클라이언트는 크게 프로세스 인스턴스 선택, 전체 프로세스 정보, 전체 업무 정보, 전체 작업자 정보 그리고 업무 흐름도로 구성되어 있다.

#### (2) 이력 정보 모니터링

이력 정보 모니터링 모듈에서는 프로세스 정의 별로 완료된 작업과 사용자의 통계 정보를 모니터링 할 수 있다.

#### (3) 프로세스 정보 모니터링 모듈

프로세스 정보 모니터링 모듈에서는 프로세스 정의에 해당되는 프로세스 인스턴스에 대한 통계 정보를 모니터링 할 수 있다.

스턴스에 대한 통계 정보를 모니터링 할 수 있다.

#### (4) 사용자 검색 모듈

사용자 검색 모듈에서는 현재 워크플로우에 참가하고 있는 작업자들을 이름, e-mail, 역할, 그룹등의 질의를 통해 검색할 수 있다.

### 5.2.2 시스템 관리 모듈

시스템 관리 클라이언트는 워크플로우에 관계된 사용자, 역할, 그룹, 권한 등을 관리하는 클라이언트를 말한다. 시스템 관리 클라이언트는 사용자 관리 모듈, 역할 관리 모듈, 그룹 관리 모듈, 권한 관리 모듈의 4개의 모듈로 구성되어 있다.

#### (1) 사용자 관리 모듈

사용자 관리 모듈에서는 워크플로우에 참가하는 사용자를 등록, 수정, 삭제, 변경할 수 있다. 또한 사용자가 속하는 역할이나 그룹을 지정할 수 있다.

#### (2) 역할 관리 모듈

역할 관리 모듈에서는 역할을 등록, 수정, 삭제, 변경 할 수 있다. 특정 역할에 해당하는 권한을 할당할 수 있다.

#### (3) 그룹 관리 모듈

그룹 관리 모듈에서는 조직의 계층 구조를 관리한다. 이 모듈을 통해 새로운 부서를 생성하거나, 기존 부서를 변경, 삭제할 수 있다.

#### (4) 권한 관리 모듈

권한 관리 모듈에서는 수행 가능한 기능 목록(Operation List)을 조합하여 구성한

권한을 관리한다. 이 모듈을 통해 새로운 권한을 추가, 기존 권한을 변경, 삭제할 수 있다.

## 6. 결론

본 연구에서는 순수 자바 기술을 이용하여 웹기반 워크플로우 관리 시스템 (InfoFlow System)을 개발하였다. InfoFlow 시스템은 크게 4 가지 모듈로 구성된다. 첫째, NT 서버 기반의 자바 엔진이 가장 핵심적인 모듈이며, 둘째, 중첩 프로세스를 설계할 수 있는 프로세스 디자이너를 이용하여 CITIS 환경에서 발생하는 비즈니스 프로세스를 모델링할 수 있다. 셋째, 프로세스를 초기화하고 일반 클라이언트 작업 인터페이스를 제공하기 위해 웹기반의 클라이언트 인터페이스를 제공하였다. 넷째, 다양한 모니터링 정보 생성하여 추적하는 기능과 함께 사용자 관리, 역할 관리, 권한 관리, 그룹 관리 기능을 담당하는 시스템 관리 모듈을 구현하였다.

InfoFlow 시스템은 CITIS 환경에서 발생하는 다양한 프로세스 관리 기술로써 사용될 수 있을 뿐 아니라, PDM, ERP 등의 핵심 모듈로서 이용될 수 있을 것으로 기대된다.

## 참고문헌

- [1] 배준수, 정석찬, 서영호, 김영호, 강석호, 1998, CORBA 기반 분산 워크플로우 관리 시스템 개발, '98 한국경영과학회/대한산업공학회 공동학술대회 논문집, 1998. 4. 24. ~ 1998. 4. 25., 경성대학교
- [2] 배준수, 김동수, 정석찬, 배혜림, 서영호, 허원창, 김영호, 강석호, CITIS 지원 워크플로우 관리 시스템 개발 요구사항, 산업공학, 제10권, 제3호, pp. 63-73, 1997
- [3] C.Mohan, G.Alonso, R. Günthör, M.Kamath, Exotica: A Research Perspective on Workflow Management Systems, *Data Engineering* Vol. 18, No. 1, 1995
- [4] David Hollingsworth, Workflow Management Coalition Specification: The Workflow Reference Model, *WfMC specification*, 1994
- [5] Esin Gokkoca, Mehmet Altinel, Ibrahim Cingil, E. Nesime Tatbul, et al., Design and Implementation of a Distributed Workflow Enactment Service, *Proc. of Intl. Conf on Cooperative Information Systems*, 1997
- [6] Graig Schlenoff, Amy Juntilla, Steven Ray, Unified Process Specification Language: Requirements for Modeling Process, *NIST*, 1996
- [7] Jintae Lee, Michael Gruninger, Yan Jin, Thomas Malone, Austin Tate, Gregg Yost, The PIF Process Interchange Format and Framework v 1.1, *PIF Working Group*, May 24, 1996
- [8] Walt Scacchi, John Noll, Process Driven Intranets: Life-Cycle Support for Process Reengineering, *IEEE Internet Computing* 97/Sep,Oct, 1997