

Web에서 데이터 흐름제어가 가능한 Mail Browser의 설계 및 구현

박 규 석[†] · 김 성 후^{††}

요 약

텍스트 위주의 메일 시스템은 응용면에서 그 한계성이 드러나 업무 및 정보처리를 자동화하고 문서흐름을 제어할 수 있는 GUI방식의 플랫폼이 개발되고 있다. 그러나 문서흐름의 자동화에 있어서 기존의 메일 시스템은 전송위주이기 때문에 수동적인 작업 수행이 될 수밖에 없다. 문서 흐름을 제어하는 플랫폼은 전자결재, 전자급고 및 문서관리, 제품 구성 관리 자동화에 아주 밀접하게 관여되어 있으며 정해진 절차에 따른 적정 정보와 문서에 대한 통제가 가능해야한다. 이러한 문제점을 해결하기 위해서는 작업흐름을 설계할 수 있는 전용 브라우저 및 문서의 흐름을 제어할 수 있는 엔진을 필요로 한다.

본 논문에서는 사용자 요구에 따라 문서의 흐름을 그래픽 인터페이스 방식으로 편집하여 절차적으로 문서를 보낼 수 있는 메일 브라우저를 개발하였으며, 메일 브라우저에서 작성된 문서의 흐름에 대한 스크립트 수행코드가 생성되도록 하였다. 또한 문서 흐름의 자동화를 위한 작업 흐름과 프로세스 관리 기능이 첨가된 데이터 플로우 엔진(Data Flow Engine)을 설계하고 구현하였다.

Design and Implementation of a Mail Browser that can control Data-Flow on the Web

Kyoo-Seok Park[†] · Seong Hoo Kim^{††}

ABSTRACT

On account of the text based mail system has it's limit to support multimedia applications, GUI based mail system platform was developed to control document flow and automatize information process. The existing mail system's to transmit data must need additional functions to automate document flow control. The platform of document flow control is deeply related to EDMS(Electronic Document Management System), workflow, Electronic Banking, DMS(Document Management System) automation, so it needs an ability to control proper data and document correctly. To resolve this problems, we are need of browser and engine to design work flow and to control documents flow.

In this paper, we develop a mail browser to design document flow by follow user's requirements. This system can generate executive script code for document flow, and we add the function of workflow and process management to automatize the document flow in this system, and then we implement this Data flow engine.

1. 서 론

인터넷 이용 기업들은 전자우편을 통하여 정보를 교환하며 업무를 수행하고 있다.

기존의 사용자 인터페이스의 경우 텍스트 위주로 사용되었으나, 응용 범위가 광범위해짐에 따라 GUI(Graphical User Interface)방식의 구현에 의한 다양한 플랫폼

* 본 논문은 1997년도 한국학술진흥재단의 공모과제 연구비 지원에 의하여 연구되었음.

† 정 회 원 : 경남대학교 정보통신공학부 교수

†† 정 회 원 : 창원대학 전산정보처리과 전임강사

논문접수 : 1999년 3월 16일, 심사완료 : 1999년 8월 21일

품으로 구성하여 업무처리 및 정보처리의 자동화뿐만 아니라 문서흐름을 제어할 수 있는 플랫폼의 개발이 진행되고 있다[7][9]. 이러한 시스템들은 WfMC(Workflow Management Coalition)를 중심으로 표준화 작업이 수행되고 있으며 CAD 데이터나 문서처리에 적용되고 있다[12].

최근에는 웹 브라우저의 확산으로 워크플로우 시스템의 클라이언트로서 웹을 사용하려는 경향도 확산되고 있으나 사용자 편의를 위해 GUI 방식으로 진행 처리하기에는 어려움이 따르고 있어, 자바 기반 또는 전용 어플리케이션으로 설계 및 개발하고 있다. 정보 전송 방법은 데이터베이스를 활용하는 방법과 기존의 메일 시스템을 활용하는 방안이 제시되고 있지만, 기존의 메일 시스템의 경우는 단순히 메시지 전송 기능만 활용할 수 있기 때문에 문서의 보관 및 스케줄에 의한 문서 전달 기능의 통제가 어렵다. 또한 기존의 워크플로우 시스템들은 통합적 관리 수단으로 데이터베이스를 기준으로 하여 설계하고 있기 때문에 멀티미디어 데이터를 다수 사용자에게 전달할 경우 많은 공간이 소요되므로 지속적인 정보관리가 어려운 면도 있다. 반면에 개개인의 정보관리가 가능하기 때문에 데이터베이스의 액세스 비율을 줄일 수 있으며 분산적인 공간 효율성을 가지는 장점도 있다.

본 논문에서는 이러한 장단점을 고려하여 기존의 메일 시스템을 이용하면서 문서의 흐름을 자동화할 수 있는 데이터 플로우 시스템(Data Flow System)과 문서의 흐름을 태스크 단위로 아이콘화 시켜 절차적으로 수행 가능한 DFG (Data Flow Grammer)를 제안한다. 논문의 구성은 우선 데이터 플로우 시스템을 크게 클라이언트/서버 기준으로 나누었으며, 클라이언트는 문서의 흐름을 설계하는 그래픽 프로세스 모델링 도구와 제어 코드 생성기, 그리고 제어 코드 전송기능 등을 설계하였고, 서버는 문서를 프로세스 단위로 자동화하여 처리할 수 있고 제어 코드의 관리 및 통제가 가능한 데이터 플로우 엔진(Data Flow Engine)을 설계하고 구현하였다.

2. 관련 연구

2.1 워크플로우 시스템

워크플로우란 전체적으로 혹은 부분적으로 비즈니스 프로세스를 컴퓨터화하여 편리하게 하거나 자동화하는

것이다[10]. 워크플로우 관리 시스템이란 업무 흐름의 자동화, 정보 및 문서의 전자화, 그리고 일관적 데이터 접근 및 제어를 통한 업무 프로세스의 개선, 통제, 관리, 공동작업을 지원하는 소프트웨어 전반을 일컫는다[11]. 워크플로우 관리 시스템은 이러한 워크플로우 로직을 컴퓨터에 의해 정해진 실행 순서에 따라 정보를 유연하게 전달하기 위한 기능들을 제공한다[11].

초창기의 워크플로우 시스템으로 SCOOP[15], Office-Talk[16] 등이 개발된 이후 수 많은 워크플로우 시스템들이 발표되었다. FileNet의 Visual Workflow[17]는 이미지 라우팅에서부터 유래되어 본격적인 워크플로우 시스템으로 개발되었으며, IBM사는 워크플로우 시스템에 트랜잭션 개념을 도입한 FlowMark[19]를 개발하였다. 한편 Georgia 대학에서는 CORBA에 기반한 METEOR[9] 워크플로우 시스템을 개발하였고, Stuttgart 대학의 ConTracts[18]는 복수의 트랜잭션을 하나의 트랜잭션으로 통합하는 구조를 제공하고 있다.

최근에는 Web을 기반으로 하여 운용되는 시스템들이 개발되고 있으며, Web에 의하여 구동되고 있다.

2.2 작업 흐름 작성기

워크플로우 시스템은 작업 흐름을 작성하기 위하여 그래픽 인터페이스 도구 또는 아이콘 기능들을 제공하고 있다. 대표적인 도구로는 CASE 도구, Georgia대학의 Graphical Workflow Designer[2], Twente대학의 Workflow Design Tool[3]등이 있다. 이러한 도구들은 다이어그램 방식으로 적용되며, 작업 흐름을 기준으로 활용되고 있다.

2.3 관련 연구 동향

근간에 워크플로우에 대한 연구가 활발히 진행되고 있다. 관련 표준화 연구로는 WfMC가 워크플로우의 5가지 인터페이스 및 메타모델 표준화를 주도하고 있으며 [12], PIF(Process Interchange Format) Working Group에서는 프로세스 정보 교환 형식에 대한 표준을 연구하고 있다. 또한 NIST (National Institute of Standards and Technology)에서는 프로세스 표현 언어에 대한 필요사항을 명세 하였다[13]. 한편 대학 또는 연구기관에서는 데이터베이스 트랜잭션 기반 프로세스 모델링 및 분석이나, 특정 도메인을 적용대상으로 하는 워크플로우 관리 시스템 구현과 상호운용성, 프로세스 표현 시맨틱에 대한 수학적 접근 및 분석, 프로세스 정의 언어 사양 및 시맨틱에 대한 연구등이 행해지고 있다[14].

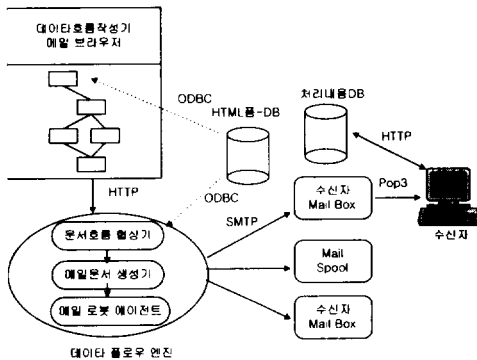
본 논문에서는 데이터베이스 트랜잭션 기반의 기존 워크플로우 시스템을 메일 시스템 기반의 트랜잭션이 가능하도록 구성하였으며, 문서의 흐름을 작성하는 그래픽 프로세스 모델링 도구는 메일 시스템에 연동하여 자동화 될 수 있도록 프로세스 정의 언어를 기반으로 구성하였다. 또한 일관적인 문서의 자동화를 위해서 데이터 플로우 엔진을 구현하여 통제 및 제어정보를 데이터베이스 기준으로 관리하며, 또한 멀티미디어 데이터를 다수 사용자에게 전송할 때 공간과 액세스의 비효율성을 보완하기 위해 문서의 내용이 포함된 제어 코드 원본만 데이터베이스에 저장하고 사본은 개인에게 메일 방식으로 라우팅되는 방식을 채택하였다.

3. 시스템 구성

제안하는 메일 브라우저와 데이터 플로우 엔진에 대한 처리모델 및 구성은 다음과 같다.

3.1 처리 모델

문서의 흐름을 제어하기 위해서는 크게 클라이언트와 서버의 두 분류로 나눌 수 있으며, 정보 전달 수단 및 수행 모델은 (그림 1)과 같이 나타낼 수 있다.



(그림 1) 데이터 플로우 수행 모델

메일 브라우저는 정보 전달 및 상호 동작을 위하여 SMTP 프로토콜을 지원하며, 전송된 정보를 획득하기 위하여 POP3 프로토콜을 지원한다. 그리고 문서의 흐름을 그래픽 프로세스 모델링 방식으로 지원하여 태스크 단위의 이미지 속성으로 데이터의 흐름 제어가 가능하다. 각각의 태스크단위인 이미지는 상호 연결 가능하도록 하여 데이터 흐름에 대한 시각 표현이 가능

하며, 생성된 개념 그래프는 점진적으로 수행되도록 스크립트 언어 코드로 변환시킬 수 있다.

메일 브라우저에서 데이터 흐름을 설계한 코드는 작업 수행을 위하여 HTTP 프로토콜을 이용하여 데이터 플로우 엔진이 있는 웹서버에 등록되며, 데이터 플로우 엔진은 데이터 흐름을 작업 스케줄에 따라 수행하도록 제어하는 일을 담당한다.

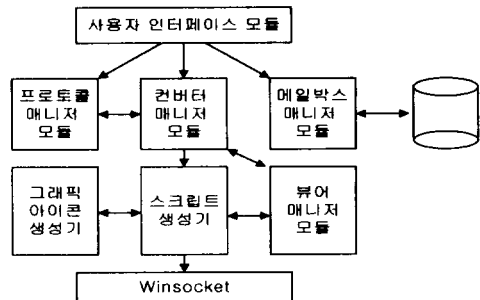
데이터 플로우 엔진에서의 정보 전송방식은 Database-driven 방식과 Mail-driven 방식을 혼용하여 사용하였으며, 메일 브라우저에서 생성된 데이터 흐름 코드는 데이터베이스에 등록되어 수행될 코드로 보관한다. 또한 데이터 흐름의 자동화를 위하여 작업 스케줄링 정보의 참조와 수행 상태의 모니터링을 위해 데이터베이스를 이용한다.

3.2 메일 브라우저의 구성

본 논문에서의 메일 브라우저는 기존의 유닉스 메일 시스템의 MTA(Mail Transfer Agent)와 클라이언트 UA(User Agent) 구성 방식을 이용하며, MIME형식과 HTML형식의 문서를 지원할 수 있도록 하여 미디어 자원을 인라인형 방식과 분산형 방식으로 접근할 수 있도록 설계하였으며, 또한 문서의 흐름을 제어할 수 있는 스크립트 생성기와 그래픽 프로세스 모델링 도구로 구성된다.

3.2.1 메일 브라우저의 구성 모듈

본 논문에서 설계 및 구현한 메일 브라우저는 크게 송신부, 수신부, 그리고 데이터 흐름 설계 도구로 구성된다. 세부 구성으로는 네트워크 모듈, 코드 생성 모듈, UA (User Agent) 모듈, 사용자 인터페이스 모듈, Viewer 모듈로 구성되며 그 구성도는 (그림 2)와 같다.



(그림 2) 구성모듈

- 프로토콜 매니저 모듈

메일을 전송하는 SMTP프로토콜과 메일을 전송 받기 위한 POP3 프로토콜로 구성되며, 분산 자원을 받기 위한 HTTP 프로토콜로 이루어진다.

- 메일 박스 매니저

메일 서버에서 전송 받은 메일을 로컬 시스템에 저장시키고 관리하는 모듈이다. 파일 매니저 형식으로 관리하게 된다.

- 사용자 인터페이스

메일 브라우저를 사용할 수 있도록 GUI환경을 제공하며 보통 메일 부분의 User Agent수행을 병행하게 되고, 메시지를 받아 디스플레이 시켜준다.

- 컨버터 매니저

전송 받은 메시지를 텍스트 형태로 보여 주고 메일 내용을 디스플레이 할 수 있도록 변환해주는 기능을 가지고 있으며 HTML을 텍스트로 변환하는 기능과 MIME을 텍스트로 변환하는 알고리즘으로 이루어진다.

- viewer 매니저 모듈

인라인 메타 데이터와 분산된 메타 데이터를 디스플레이 하는 기능으로 MIME Header 형식에 따라 표현된다.

- 스크립트 생성기

그래픽 프로세스 모델링 도구에서 작성된 문서 흐름의 모델에 따라 스크립트 언어로 번역한다.

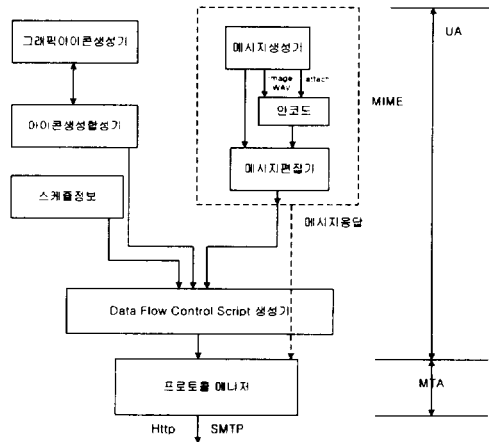
- 그래픽 아이콘 생성기

그래픽 프로세스 모델링 도구에 사용되는 사용자 인터페이스의 집합과 규칙에 따라 인터페이스의 적용 범위를 설정할 수 있는 기능으로 구성된다.

3.2.2 메일 브라우저의 송수신처리

메일 브라우저는 데이터 흐름 스케줄 정보 전송과 수신된 문서의 응답 및 전송의 두 가지 측면에서 활용된다. 데이터 흐름 스케줄 정보 전송은 데이터 플로우 엔진 서버에 스케줄 정보를 등록하기 위해서 HTTP 프로토콜을 이용하여 데이터 플로우 엔진의 웹 서버에 전송하며, 데이터 흐름 스케줄 정보에 따라 데이터 플로우 엔진에서 전달받은 메시지의 응답 및 전송정보는 SMTP 프로토콜을 이용하여 메일박스로 전달된다.

메일 브라우저의 송신부 모델은 (그림 3)과 같다.



(그림 3) 메일브라우저의 송신 모델

메시지의 전송정보 및 응답은 전자메일 표준 프로토콜인 SMTP를 이용하며 두 MTA간의 NVT ASCII를 이용하여 상호 메시지를 교환하여 클라이언트에서 서버로 명령을 전송하게 된다. SMTP 명령은 HELO, MAIL, RCPT, DATA 그리고 QUIT를 사용하여 전송을 수행하게 된다.

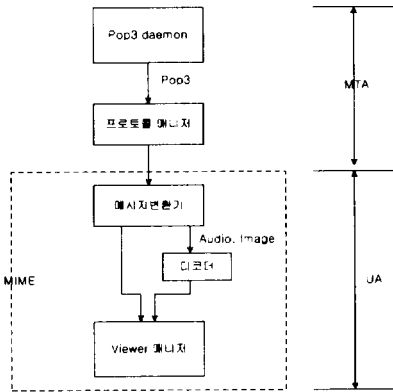
SMTP는 7비트 코드 형태의 전송만을 허용하기 때문에, 메일 문서에 포함된 한글 표기와 멀티미디어 데이터는 MIME(Multipurpose Internet Mail Extensions) 표준을 이용해 7비트 코드형태로 변환하여 인코드 된다. 인코딩 기법은 한글과 같은 8비트 문자를 처리하기 위해 Quoted-printable기법을 사용하며, 메일문서의 헤더부분은 Base64기법으로 처리하였다.

데이터 흐름 스케줄 정보는 그래픽 아이콘 생성기로 문서의 흐름을 작성하게되며, 작성된 스케줄 정보는 MIME 처리된 메일 문서와 함께 Data Flow Control Script 생성기를 통하여 스크립트 언어를 생성하게 된다.

데이터 플로우 엔진에서 데이터 흐름 스케줄 정보 스크립트를 구문 분석기를 통하여 메시지 부분과 스케줄 정보를 추출하게 되며 스케줄 정보에 따라 메시지 부분을 라우팅을 하게된다. 라우팅시 사용자의 메일박스에 메시지 부분을 전달하게 된다. 메일 브라우저에서의 수신모델은 (그림 4)와 같다.

전송된 전자 메일 문서를 보여주기 위해서는 문서 분석기가 필요하며 분석된 데이터는 실제 미디어 자원을 표현하기 위해 디코딩 과정을 거쳐야한다.

데스크탑 PC 사용자가 메일 서버에 접속하여 메일



(그림 4) 메일 브라우저의 수신모델

을 전송하는 POP3 프로토콜을 이용하여 메일박스에서 수신된 문서를 읽어들이는다. POP3 클라이언트와 서버의 상호동작은 권한 상태, 트랜잭션 상태, 업데이트 상태의 3가지 상태를 가진다.

인터넷 포트 110번을 이용하여 메일 서버에 연결되며 USER 명령으로 메일 서버의 계정을 확인하고 PASS 명령으로 패스워드를 입력하여 로그인 절차를 수행하게 된다. 트랜잭션 명령으로 메일 박스에서 문서를 읽어 들이며, 읽어들이는 문서가 MIME 형식을 가지면 메일 문서를 라인 단위로 읽어들이어 헤드 필드를 검사하게 되며 Mime-Version, Content-Type, Content-transfer-Encoding, Content-ID, Content-Description 등을 분석하고 text, html, Image, audio 등 sub-type을 검사하게 된다. viewer 매니저를 통하여 텍스트, 이미지, 오디오 등을 표현한다.

3.2.3 그래픽 프로세스 모델링 도구

메일 브라우저는 문서의 흐름을 제어할 수 있는 그래픽 프로세스 모델링 도구를 제공한다. 기존의 작업 흐름 설계 도구로는 작업흐름을 나타내는 다이어그램으로 표시하여 사용하고 있지만 본 논문에서는 태스크의 개념적 의미관계로 설정하였다.

그래픽 프로세스 모델링 도구를 이용하여 작업 흐름의 순서를 설정할 수 있으며 문서의 흐름관계를 아이콘들간의 연결로 표현하고 각각의 아이콘은 태스크 속성을 가진다. 수행 속성은 행위자, 역할, 수행조건, 트리거링 타입으로 나누어 작업 수행 상태를 기반으로 작업 흐름관계를 표현한다.

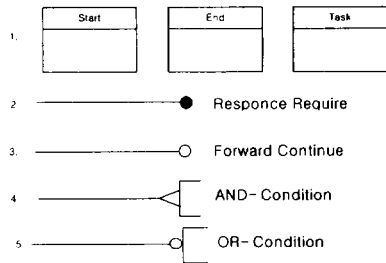
구성 매체 각각의 표현 시간과 매체들 사이의 시간적

관계를 명시하며, 구성 매체들의 시간적 특성, 명시해주는 시간모형(temporal mode)과 사용자의 요구나 선택을 수행할 수 있는 대화식(interactive)표현으로 구성한다.

이러한 그래픽 인터페이스는 그래픽 편집 기능 및 관련 아이콘이 제공되어야하며, 개체간의 흐름관계를 나타내기 위해 각각의 개체는 태스크 개념을 가지게 하고, 태스크 행위들간의 데이터 또는 제어 흐름들이 표현될 수 있도록 하였다.

3.2.4 그래픽 인터페이스의 구성

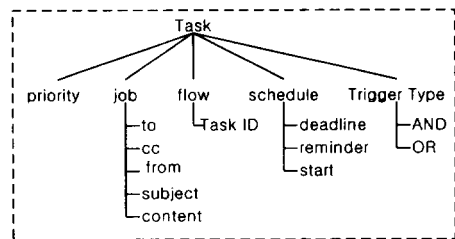
논리적 저작 모형은 데이터 흐름 제어 시스템의 그래픽 프로세스 모델링 도구인 Object Modeling Technique(OMT)의 객체 모형화(Object modeling)기법 중 일부를 변형하여 이용하며, 논리적 모형에 사용하는 OMT 그래프 기호들은 (그림 5)와 같이 표현한다.



(그림 5) 그래픽 아이콘

이 기호들은 OMT 기법 중 객체들 사이의 구조적 관계를 나타내는데 필요한 것들을 나타낸 것으로, 데이터 흐름제어 스크립트를 구성하는 구성 순서와 종류를 표현한다.

데이터 흐름제어 그래픽 프로세스 모델링 도구에서 지원되는 태스크 단위는 (그림 6)과 같이 정의된다.



(그림 6) 태스크 단위 모델

문법 : Task=[J, F, S, T, C]에서 J는 단위업무(Job)를 의미하며 전자 메일을 보낼 내용을 가지고 있다. F

는 단위업무간의 흐름(Flow)을 의미하며 단위업무에 대한 응답 또는 데드라인 날짜에 의해 다음 태스크로 진행하게된다. S는 일정 스케줄로써 단위 업무의 시작 시간, 데드라인, 독촉 시간으로 구성된다. T는 Trigger를 의미하며 AND 와 OR의 분기 조건을 가지고 있다. P는 우선 순위를 의미하며 태스크의 연결 조건에 이용된다.

제안하는 DFG(Data Flow Grammer)는 다음과 같이 정의된다.

- 즉, 데이터 플로우 문법 DFG=[I, L, M]
- I = { Start, End, Task }
- Task = [J, F, S, T, C]
- L = { Connect, Disconnect, delete}
- M =(AND-split, OR-split, response, forward)

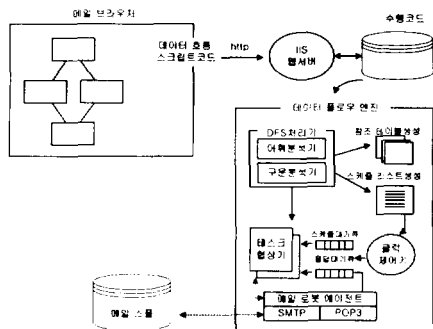
그래픽 연산자들은 Start, End, Task개체로 구성되고, 각각의 Task에는 스케줄러 시간이 존재한다. 개체간의 연결관계는 L로 나타내며 M은 연결 조건을 나타낸다. L에서 Connect와 Disconnect는 각 개체간의 연결 및 삭제에 제약 조건이 따르게 되며, M에서 AND-split와 OR-split 연결 관계의 추가 및 삭제시에도 Task의 Trigger type과 Task#에 영향을 준다.

3.3 데이터 플로우 엔진

데이터 플로우 엔진은 메일 브라우저에서 작성된 DFCS를 전송받아 수행되며, 데이터 플로우 서버를 구성하는 처리 및 구성은 다음과 같다.

3.3.1 데이터 플로우 엔진의 구성

데이터 플로우 엔진은 스케줄 정보를 받아들이기 위해 기존의 웹 서버 및 데이터 베이스와 연동하여 동작되게 하였으며 그 구성도는 (그림 7)과 같다.



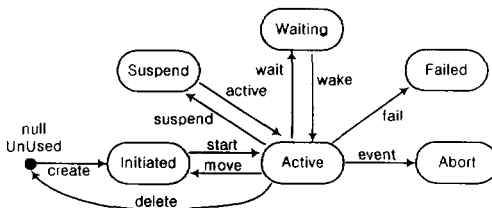
(그림 7) 데이터 플로우 엔진의 구성

데이터 플로우 엔진은 크게 DFCS 처리기, 태스크 협상기, 클럭 제어기, 메일 로봇 에이전트, 이벤트 조정기, 그리고 메일 생성기로 구성하였으며, DFCS 처리기는 수행 코드 데이터베이스에서 데이터 흐름 제어 스크립트 코드의 어휘 및 구문을 분석하게 되며 태스크 스케줄 정보 테이블, 태스크 수행 상태 테이블을 생성하고 스케줄 태스크 테이블에 수행 코드를 등록하게 된다. 각각의 흐름 제어 코드는 기본적으로 두 개의 테이블을 가진다.

언어 분석기에서 의미분석을 마치면 흐름 제어 코드마다 태스크 협상기의 쓰레드를 할당받아 스케줄 정보를 관리하게 되며, 클럭 제어기는 데이터 흐름 제어 리스트를 시간 알람기능을 이용하여 태스크 협상기로 보내기 위해 스케줄 대기큐로 디스패치 하는 기능을 가진다. 메일 로봇은 메일 라우트 기능과 메일 수집기능으로 구성되며, 메일 라우트 기능은 메일을 전송하는 SMTP 프로토콜로 구성되어 있으며 스케줄 되어있는 리스트 정보를 이용하여 데이터베이스에 저장되어 있는 업무 또는 문서내용을 각 메일박스로 전송하게 된다. 메일 수집기는 서버가 지정된 일정 시간에 데이터 플로우 전용 메일 스폴의 상호 응답이나 대기 지연 메시지를 추출하여 응답 대기큐에 적재한다. 태스크 협상기는 스케줄 대기 큐와 응답 대기큐의 정보를 이용하여 절차를 협상하고 진행하는 기능을 가진다. 태스크 협상기는 다중 쓰레드로 구성되며 각 쓰레드 데이터 흐름제어 정보의 처리기능은 생명 주기단위로 운영된다.

3.3.2 태스크 협상기의 생명주기

쓰레드 단위로 생성된 태스크 협상기는 (그림 8)과 같은 생명주기를 가진다.

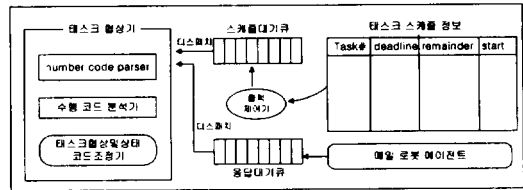


(그림 8) 태스크 협상기의 생명주기

create는 새로운 쓰레드가 할당되어 실행되는 프로세스단계를 말하며, start는 데이터 플로우 태스크가

서비스 제공을 위해 자신의 서비스 동작을 시작하는 과정이며, suspend는 태스크 협상기의 실행이 자체의 원인 또는 수행 상태 코드의 수정, 상대방 메시지 응답에 대한 delay 요청 등에 의해 잠시 멈추는 상태이다.

active는 suspend 상태에서 동작을 시작하는 과정이며, wait는 특정 이벤트에 대해 태스크 협상기의 상태를 Waiting 상태로 변환하는 과정이다. wake는 Waiting 상태에서부터 태스크 협상기 활동을 재개하는 것으로 태스크 협상기 자체에 의해서가 아니라 사용자의 확인 메시지에 의해 변환된다. delete는 태스크 협상기의 실행을 정지시키고 데이터 흐름 제어는 수행 종료로 인해 삭제되는 상태이다.



(그림 9) 스케줄 정보 추출 과정

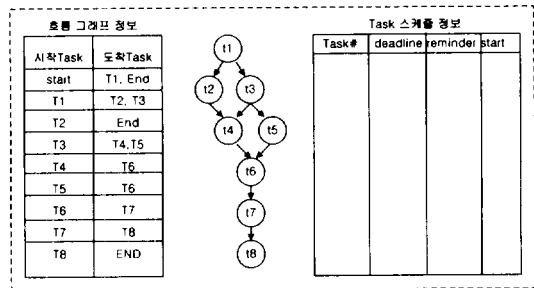
3.3.5 태스크 협상기

태스크 협상기는 (그림 10)과 같이 태스크 협상을 위한 흐름 그래프 정보를 구성하며, 흐름 그래프 정보는 시작 태스크와 도착 태스크 도표로 구성된다.

3.3.3 DFCS(Data Flow Control Script) 처리기

메일 브라우저에서 작성된 DFCS는 데이터 플로우 엔진에서 수행 처리되기 위해 내부 표현으로 변환하며, 변환된 내부 표현은 데이터베이스에 저장되어 시스템 종료나 모니터링 정보를 제공한다.

DFCS 처리기에서 처리되는 테이블은 프로세스 테이블, 프로세스 상태 테이블, 스케줄 리스트 테이블, 스케줄 상태 테이블로 구성되며, 스케줄 정보를 처리하기 위해 작업 번호, 스케줄 번호, 응답코드등을 조합하여 데이터 흐름을 제어할 태스크 넘버를 할당하며, 스케줄 리스트 테이블에 정보를 등록한다.



(그림 10) 프로세스 흐름 정보 추출

프로세스 흐름 그래프를 구성하는 연결 가지 테이블은 프로세스 노드간의 관계 정보에 대한 분석이 가능하며, 태스크 협상기가 연결 가지 테이블을 참조하여 시간 스케줄링에 따라 자동적으로 메일 로봇 에이전트를 통하여 메일 문서를 라우팅하게 된다.

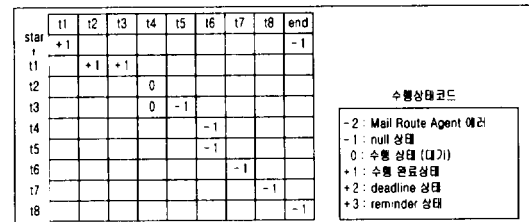
(그림 11)은 태스크 협상기가 참조하는 협상 테이블을 나타낸 것이다.

3.3.4 클럭 제어기(Clock Controller)

클럭 제어기는 DFCS 처리기가 생성한 스케줄 리스트에서 지정된 시간에 수행될 스케줄 리스트를 추출하여 스케줄 대기 큐에 저장하는 역할을 수행하며 태스크 협상기와 상호 협조적으로 동작한다. 태스크 협상기에서 수행된 스케줄 중에서 데드라인 시간을 가지는 태스크는 데드라인 정보가 스케줄 상태 테이블과 스케줄 리스트 테이블을 이용하여 클럭 제어기의 스케줄 대기 큐에 등록된다.

(그림 9)는 스케줄 정보 추출과 메일 로봇 에이전트 사용자의 응답 메시지를 추출하여 각각의 대기큐에 저장하는 개념도이다.

클럭 제어기는 태스크 스케줄 정보 중에서 태스크 시작 시간을 추출하여 스케줄 대기큐에 저장시키며, 메일 로봇 에이전트는 응답 대기큐에 저장시켜 태스크 협상기가 각각의 대기큐에서 정보를 추출하여 제어 상태를 협상하게 된다.

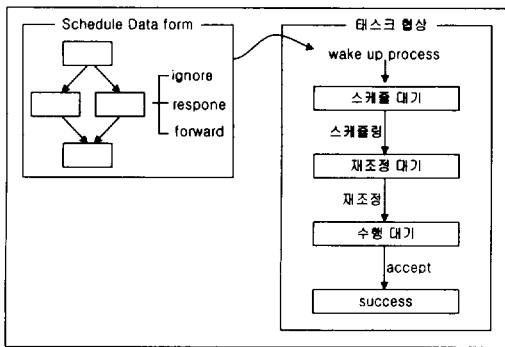


(그림 11) 태스크 협상 테이블

태스크 협상 테이블은 각각의 데이터 흐름 제어 작업마다 하나의 테이블을 관리하고 있으며 문서의 흐름 태스크마다 상호 응답 절차에 따른 수행 상태를 체크

하게 된다. 각 태스크의 상태정보가 -1일 때는 Null 상태를 의미하며, 수행은 0, 수행 완료는 +1, deadline은 +2, remainder 상태는 +3으로 처리하여 제어 협상기에서 AND 조건의 태스크를 처리 할 경우에 열(row)을 기준으로 1인 상태일 때만 행(column)으로 태스크 제어기가 이동되어 계속적인 작업을 하게 된다. 만약 OR 조건의 태스크 처리 관계일 때는 non-blocking 관계를 의미하며 현 태스크의 수행 상태가 0이며 수행 완료 상태인 1이 아니라도 계속적인 태스크 처리를 하게된다.

태스크 처리시 진행 절차간의 상호 응답이 ignore, response, forward 조건의 코드를 보내게 될 때 태스크 협상기를 통하여 태스크 상태 정보를 변경한다. 태스크 상태 정보를 변경하기 위한 절차는 (그림 12)와 같다.



(그림 12) 태스크 코드 변경

태스크 상태 코드 변경은 중복 처리를 막기 위해 스케줄 진행 상태를 대기 상태(wait)로 처리한 뒤 협상 상태 진행을 멈추게 한 다음 태스크 진행 협상에 따라 상태 테이블 내용을 변경하도록 한다. 코드 변경 후 태스크 협상기 제어 상태를 wake로 변환하여 계속적으로 진행하게된다.

태스크 협상기의 알고리즘은 (그림 13)과 같다.

```

struct task { char *start; char *target;
             char *semantic; char *message_no;
             char *task_id; date *c_date; int split;
             char *serial; } schedule, response; . . . .
event_mess = life_cycle_check();
do {
    switch event_mess {
        case "init", "done" : init_life_cycle();
        case "active" :
    
```

```

do { dispatch_schedule_queue(&schedule)
      number_code_parser(&schedule);
      execute_code_analyzer(&schedule);
      if (find_response_queue(schedule->serial))
      { number_code_parser(&response);
        execute_code_analyzer(&response);
        switch response->semantic { . . . }
      }
      else {
        switch schedule->semantic {
            case "reminder" : reminder_mess(&schedule);
            case "deadline" : deadline_mess(&schedule);
            case "mkmsg" : send_mess(&schedule);
        }
        remark_negotiation(schedule);
        'split :1 OR조건 split :2 AND조건
      })while((schedule->split == 1)||schedule->split == 2))
      . . . . .
    } while((event_mess=life_cycle_check())!="delete")
    
```

(그림 13) 태스크 협상기 알고리즘

3.3.6 상호동작 시나리오

데이터 플로우 시스템이 제공하는 그래픽 프로세스 모델링 도구와 데이터 플로우 엔진의 상호 동작 알고리즘은 다음과 같다.

1. 문서의 흐름을 정의할 수 있는 그래픽 편집 기능과 아이콘을 이용하여 흐름을 설계한다.
2. 데이터 흐름의 모델인 아이콘을 이용하여 시각적으로 흐름 관계를 설정한다. 작업에 대한 지시 사항은 email 폼으로 작성하며 스케줄에 관련된 deadline 시간, remainder 시간, start 시간을 지정한다. 각각의 email 폼의 관계는 AND 조건과 OR조건으로 수행 관계를 설정하며, END 아이콘을 이용하여 데이터 흐름의 설계를 마무리시킨다.
3. 작성된 문서의 흐름은 스크립트 제너레이터에 의해 제어코드 언어를 생성하게 되며 수행 관계의 모델은 그래픽 프로세스 모델링 도구에 의해 오류를 검증한 상태가 된다.
4. 생성되어진 스크립트는 메일 브라우저에 의해 HTTP 프로토콜을 이용하여 데이터 플로우 엔진이 있는 웹 서버에 문서의 흐름이 정의된 스크립트 언어를 데이터 베이스에 등록한다.
5. 데이터 베이스에 등록된 작업은 데이터 플로우 엔진에서 디스패치 하게되며 데이터 플로우 스크립트 어휘 분석기에 의하여 토큰열을 파싱하고 구문 분석기로 의미를 분석하게 되며 데이터 플로우 실행

기는 구문 분석기로 생성한 6개의 테이블을 이용하여 상태 정보 및 수행 정보를 참조하게 된다. 스케줄 된 정보는 예약 대기 큐에 등록되어 시간별 리스트를 구성한다.

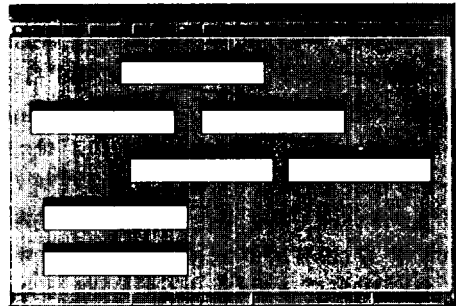
6. 데이터 플로우 실행기는 작업 처리의 수행 상태와 상태 정보를 참조하며 작업 스케줄 정보는 클럭 컨트롤러에 의해 이벤트를 발생 시켜 흐름을 제어한다. 태스크 협상기 모듈에서는 수행 진척 작업과 상호 응답을 처리하고 있으며 AND조건과 OR 조건을 처리하는 방식을 지원한다.
7. 데이터 플로우 실행기가 작업 스케줄이 있는 작업 리스트 큐에서 정보를 받아 문서에 대한 흐름을 제어하여 당사자에게 메일을 라우팅 한다. 메일 로봇 에이전트는 SMTP 프로토콜과 POP3 프로토콜을 지원하여 작업을 수행하게 된다. 사용자에게 의하여 메일 체크 시간을 지정하며, 문서 하달은 SMTP 프로토콜을 이용하여 태스크의 넘버를 subject 앞부분에 marking하여 작업 태스크를 분류하며, 수행에 대한 응답은 POP3 프로토콜에 의해 응답을 받아 낸다. 공용된 메일 스플은 한 계정을 중심으로 작업을 통괄한다.
8. 클럭 컨트롤러에 의해 타이밍 서비스를 수행하며 스케줄에 등록된 작업은 이벤트를 발생시켜 데이터 플로우 실행기를 수행하게 된다. 수행시 태스크의 문맥 교환이 일어나며 수행 정보와 상태 정보를 데이터 베이스에서 불러들인다.

4. 구현 및 평가

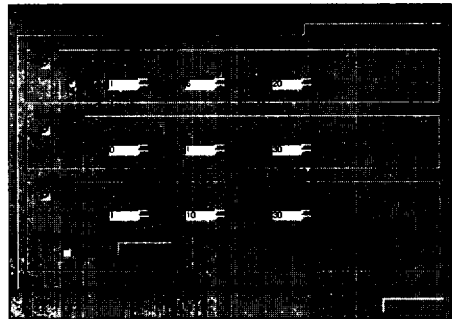
시스템 구조는 서버측 시스템으로 Microsoft Windows NT서버 기반으로 IIS 4.0웹 서버, MS-SQL6.5를 데이터 베이스로 사용하였다. 클라이언트측 시스템은 Windows 95 기반으로 제안한 메일 브라우저를 구현하였으며, 데이터 흐름제어 스크립트는 ASP 프로그램으로 웹서버에 등록하도록 하였으며 VC++로 메일 브라우저와 데이터 플로우 엔진을 구현하였다.

(그림 14)는 메일 브라우저로 데이터 흐름제어 스케줄 정보를 작성하는 과정을 나타낸 것으로 그래픽 인터페이스 방식으로 흐름 정보가 구성된다.

메뉴판에 있는 아이콘으로 각각의 흐름 정보 아이콘을 생성할 수 있으며 태스크 단위로 의미를 부여하게 된다. 각 태스크 정보의 내용은 (그림 15)와 같이 설정할 수 있다.



(그림 14) 데이터 흐름제어 스케줄 작성



(그림 15) 아이콘의 등록정보

각각의 등록 정보는 시작 시간, 데드라인 시간, 독촉 시간으로 구성되며 기한을 명시할 수 있으며, 전송 받을 메시지와 내용은 메일 주소를 명시하여 전달되도록 하였다.

(그림 16)은 메일 브라우저에서 그래픽 인터페이스 모델링 도구로 문서 내용 작성시 생성된 스크립트 내용이다.

```

<dataflow>
begin
[start t0] to [t1 done none]
[start t1] to [t2 mkmsg response] and [t3 mkmsg response]

.....
end
<schedule>
t1:
{
start: 19990301 06:20
deadline: 19990302 10:10
remainder: 19990302 11:30
} .....
<message>
t1:
{

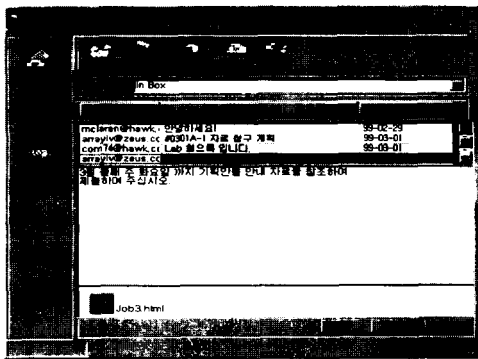
```

```
to: arrayiv@zeus.kyungnam.ac.kr
subject: =?EUC-KR?B?M7/5wNrBpiCxuMDUULDoyLk=?=
cc: mclaren@zeus.kyungnam.ac.kr
MIME-Version: 1.0
Content-Type: text/plain;
Content-Transfer-Encoding: quoted-printable
.....
} .....
```

(그림 16) 데이터 플로우 스크립트

스크립트 언어는 <dataflow>, <schedule>, <message>로 나누어지고, <dataflow>는 데이터 플로우 태스크 정보로 구성되며 begin, end구조를 가진다. <schedule>은 각 태스크의 속성을 가지며, <message> 실제 전송될 메시지로서 '헤더' 부분과 '바디' 부분으로 구성된다.

(그림 17)은 데이터 플로우 엔진의 메일 로봇 에이전트로부터 전송받은 메시지 내용을 나타내고 있다. 각각의 메시지는 데이터 흐름제어 스크립트 등록 날짜와 연번 그리고 태스크 번호로 메시지 번호가 구성되도록 하였으며 응답 코드는 메시지 구성번호 끝에 추가된다.



(그림 17) 전송 받은 메시지

본 논문에서 제안한 데이터 흐름 스크립트 코드는 메일 시스템 기반으로 설계되었으며, 문서의 흐름을 시각적으로 설계할 수 있을 뿐만 아니라 태스크 단위로 작업 스케줄을 설정함으로써, 작업의 흐름 설계에 응용할 수 있다. 또한 생성되는 개념 그래프는 태스크들의 속성 및 흐름 관계를 내포하고 있기 때문에 기존의 다양한 흐름관계에 대한 태스크 단위의 추가 설정과 변환이 쉽다. 또한 데이터 플로우 엔진에서 사용

하는 태스크 협상기는 트랜잭션 처리기의 역할로 활용할 수 있을 뿐만 아니라 메일 로봇 에이전트로 데이터의 생성, 보관 및 참조가 가능하며, 참조된 다양한 문서내용은 응용 범위에 따라 메일링 서비스에 이용할 수 있다. 워크플로우 시스템은 데이터베이스 기반으로 운용되지만 제안 시스템의 경우 메일 시스템 플랫폼 기반으로 설계되었으며, 데이터베이스 기반으로 멀티미디어 데이터를 내포한 문서를 보내게 될 경우 다수 사용자에 대하여 유지 보수하여야하는 단점을 가지고 있지만 제안 시스템의 경우는 메일양식 기반으로 MIME처리를 하기 때문에 문자자체에 멀티미디어 데이터를 포함하는 방식으로 처리되기 때문 개별적인 유지 관리가 가능하였다.

<표 1>은 제안 시스템을 기존의 그래픽 인터페이스 도구들과 특성 및 정보 전송 방법등을 비교한 표이다.

<표 1> 기존 도구의 특성 및 전송방법 비교

도구 특성	Twente [3]	METER [9]	FileNet [17]	Flow Mark[19]	제안한 시스템
설계기반	그래프 기반	스크립트 기반	그래프기반	그래프 기반	태스크 개념기반
정보교환	데이터베 이스기반	데이터베 이스기반	전자메일 (Exchange) 시스템&데 이터베이스 기반	데이터베 이스기반	전자메일 (SMTP & POP3) 시스템&데이 터베이스기반
정보저장	중앙집중	중앙집중	중앙집중	중앙집중	분산 & 원본저장
문서처리	트랜잭션	트랜잭션	라우팅	트랜잭션	라우팅
경고 & 태드라인	지원안함	지원안함	지원	지원	지원

5. 결 론

본 논문에서는 사무 업무의 연계에 있어서 핵심 정보 기술인 데이터 흐름 제어와 자동화가 가능한 메일 브라우저 및 데이터 플로우 엔진을 설계 및 구현하였다.

본 논문은 기존 시스템의 정보교환 방법과는 다르게 메일 시스템 기반의 트랜잭션이 가능하도록 구성하였으며, 그래픽 프로세스 모델링 도구는 기존의 메일 시스템에 연동하여 자동적으로 수행될 수 있도록 프로세스 정의 언어를 기반으로 구성하였다. 또한 메일 브라우저는 다양한 프로토콜(HTTP, SMTP, POP3)을 지원하여 인터넷 환경에서 데이터베이스를 접근할 수 있도록 하였으며, 그래픽 프로세스 모델링 도구를 제공하여 사용자가 직접 문서의 흐름을 스케줄링 할 수도

있다. 그래픽 인터페이스 모델링 도구에서는 태스크 단위로 작업의 스케줄이 가능하도록 정의하여 점진적인 절차에 따라 문서의 흐름 제어가 가능하며, 기존의 업무 자동화에 적용을 할 수도 있다. 또한 데이터 플로우 엔진은 메일 로봇 에이전트를 제공하기 때문에 데이터 흐름에 대한 트랜잭션 처리가 가능하며, 태스크 협상기로 진행 절차를 조절할 수 있어 문서흐름에 대한 협상 및 라우팅이 가능하기 때문에 문서 흐름의 자동화가 가능하다.

향후 웹 기반의 문서처리를 위해 다양한 품 양식을 정의할 수 있는 도구와 진행과정을 모니터링하여 통제할 수 있는 시스템으로의 확장이 요구된다.

참 고 문 헌

- [1] D. Georgakopoulos and M.Hornick, "An Overview of Workflow Management : From Process Modeling to Workflow Automation Infrastructure," *Journal on Distributed and Parallel Database Systems*, Vol.3, No.2, pp.119-153, 1995.
- [2] A. Sheth, "Workflow Automation : Applications, Technology and Research," Tutorial notes of SIGMOD Conference, May, 1995.
- [3] C.Castroianni, Development of a Workflow Design Tool for the Scarabeus Project, Master's thesis, Dept. of Computer Science, Univ. of Twente, 1995.
- [4] Jonathan B.Postel, "Simple Mail Transfer Protocol," RFC 821, 1982.
- [5] David H. Crocker, "Standard for the format of ARPA Internet Text Messages," RFC 822, 1982.
- [6] Moore, K. "MIME(Multipurpose Internet Mail Extensions)" Part Two : \Message Header Extension for Non-ASCII Text, RFC 1522, 1993.
- [7] R. Marshak, "WorkFlow : Applying Automation to Group Process," *Groupware : Technologies and Applications*, Prentice Hall, Inc. Coleman D. and Khanna R. Ed. pp.71-97, 1995.
- [8] K. Abbott, S. Sarin, "Experiences with Workflow Management : Issues for the Next Generation," *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, USA, pp.22-26. Oct. 1994.
- [9] ORBWork : "A Distributed CORBA-based Engine for the METEOR Workflow Management System," University of Georgia, Athens, CA.
- [10] L. Fischer, "THE Workflow Paradigm -The Impact of information Technology on Business Process Reengineering, Future Strategies," Inc., Alameda, CA, 2nd. edition, 1995.
- [11] C.Mohan, "Tutorial : stste of the Art in Workflow Management System Research and Products," IBM Almadan Research Center, Workflow system and Interoperability, Istanbul. Turkey, Agust, 1997.
- [12] David Hollingsworth, "Workflow Management Coalition Specification : The Workflow Reference Model," WfMC, 29, Nov., 1994.
- [13] C.Mohan, G.Alonso, "Exotica : A Research Perspective on Workflow Management System," *Data Engineering Vol.18, No.1*, 1995.
- [14] Drik Wodtke, Jeanine Weissenfels, Gerhard Weikum, Angelica Kots Sittrich, "The Mentor Project : Steps Towards Enterprise-Wide Workflow Management," Technical Report, 1996.
- [15] M. Zisman, "Representation, Specification, and Automation of Office Process, Ph.D.dissertation," Wharton School, Univ. Pennsylvania, 1997.
- [16] C. Ellis, M. Bernal, "Officetalk-D : An Experimental office information system, Proceedigs of the ACM-SIGOA Conference on Office Information System," pp. 281-288, No. 1992.
- [17] W. Fisher, J.Gilbert, "FileNet : A Distributed System Supporting WorkFlo : a Flexible Office Procedures Control Language," *IEEE Computer Society Office Automation Symposium*, pp.247-249, April 1987.
- [18] A. Reuter, F. Schwenkreis, "ConTracts - A Low-Level Mechanism for Building General Purpose Workflow Management System, Bulletin of the Technical Committee of Data Engineering," *IEEE*, Vol.18, , March 1995.
- [19] G. Alonso, C.Mohan, and Gunthor, "Exotica / FMQM : A Persistent Message-Based Architecture for Distributed Workflow Management," *Proc. of the IFIP WG8.1 Working Conference on Information Systems for Decentralized Organizations*, Trondheim, August, 1995.



박규석

e-mail : kspark@hanma.kyungnam.ac.kr
1988년 중앙대학교 전자계산학과
(이학 석/박사)
1990년~91년 미 UCLA 객원교수
1982년~현재 경남대학교 정보통신공학부 교수

1998년~현재 행정자치부 정책자문위원(전자정부분과)
관심분야 : 운영체제, 멀티미디어시스템, 정보통신시스템.



김성후

e-mail : arayiv@zeus.yeungnam.ac.kr
1995년 경남대학교 전산통계학과
(이학사)
1997년 경남대학교 전자계산학과
(공학석사)
1997년~현재 경남대학교 컴퓨터
공학과(박사과정)

1998년~현재 창신대학 전산정보처리과 전임강사
관심분야 : 정보통신시스템, 워크그룹, 분산네트워크