

# 조직변화에 유연한 지능형 워크플로우 자동화 시스템: K-WFMS

이 하 빈\*, 박 성 주\*\*

## K-WFMS: An Intelligent Workflow Management System for Changing Organization

Lee, Ha Bin\*, Park, Sung Joo\*\*

In this paper, an adaptive workflow management system, called K-WFMS, is proposed. The K-WFMS integrates database system and knowledge-based system to automate business processes that are executed with complex and various business rules such as task scheduling, role resolution, and exception handling rules. The K-WFMS is adaptable in the sense that it allows its users to change workflow schema in the course of workflow execution as well as it provides rule-based modeling constructs to handle predictable exceptions during workflow modeling. The overall architecture and implementation of K-WFMS are explained, and the change propagation mechanism to maintain validity of workflow model is suggested.

---

\* BText Technologies 지능형 시스템 연구소

\*\* 한국과학기술원 테크노경영대학원

## I. 서론

1990년대에 등장한 워크플로우 관리 시스템(WFMS: workflow management system)은 기업의 비즈니스 프로세스 자동화를 위한 핵심 기술로서 많은 성공사례가 제시되었다[Fisher, 1999]. 그러나 최근 워크플로우 관리 시스템이 비즈니스 프로세스에 대하여 너무 틀에 구속적인 모형화 및 자동화를 강요하므로써, 현실의 작업현장에서 발생하는 다양한 형태의 예외상황 등을 해결하기에는 한계를 지니므로 워크플로우 실행시 발생할 수 있는 예외상황을 처리하고 사용자의 자율성을 허용해 줄 수 있는 적응적인 WFMS(adaptive WFMS)에 대하여 많은 연구들이 이루어지고 있다[Casati et al., 1998; Reichart and Dadam, 1998; Klein and Dellaocas, 2000]. 이러한 적응적인 WFMS에 대한 연구는 크게 워크플로우 모형화 시점에서 워크플로우 실행시 발생할 수 있는 예외상황들을 처리할 수 있도록 해줌으로써 WFMS의 사전적인 유연성(prior flexibility)을 확보하기 위한 연구와 모형화 시점에서 예측이 불가능한 예외상황을 처리할 수 있도록 해 줌으로써 WFMS의 사후적 유연성(posterior flexibility)을 높여주기 위한 연구 두 가지로 구분해 볼 수 있다[Joeris, 1999].

그러나, 지금까지의 적응적 WFMS에 대한 연구는 대부분 대상으로 하는 비즈니스 프로세스를 단순히 수행되어야 하는 과업의 순서화된 집합으로 가정하므로 다양한 조직의 정책이 존재하고 조직구성원 간의 이해관계가 상충되는 조직에서 활용되기에는 한계를 지니고 있다. 특히, 분권화 되고 상충되는 이해관계를 지닌 사용자 계층이 존재하는 조직에서 운영되는 비즈니스 프로세스는 관련된 규정들이 복잡하고 다양하며, 특히 예외상황과 같은 특수한 경우를 처리하기 위한 규칙들이 복잡하게 얽혀있

다. 따라서, 이러한 비즈니스 프로세스를 자동화하기 위한 워크플로우 시스템은 비즈니스 규칙을 효율적이고 효과적으로 관리할 수 있는 장치의 마련이 시급하다.

이러한 동기로 인해 그 동안 비즈니스 프로세스에서의 비즈니스 규칙들을 분석하고 관리하기 위한 연구들이 꾸준히 이루어져 왔으며 [Appleton, 1984][Moriarty, 1993], 최근에는 워크플로우를 비즈니스 규칙의 집합으로 해석한 KWM(Knowledge-based Workflow Model)이 제시되었다[Lee, Kim, and Park, 1999]. 그러나, 이러한 연구들은 모형화 관점에서 비즈니스 규칙을 다루고 있을 뿐, 이러한 모형에 기반하여 실제로 운영될 수 있는 WFMS의 구현방안 및 시스템 차원에서의 예외상황 처리 방안을 제시한 연구가 아직 미진한 실정이다.

본 논문에서는 다양하고 복잡한 비즈니스 규칙에 의해 그 실행이 제약 받는 비즈니스 프로세스 자동화를 위한 적응적인 WFMS인 K-WFMS(Knowledge-based WFMS)에 대하여 소개한다. K-WFMS의 특징은 크게 두 가지로 요약된다.

첫째, 비즈니스 규칙을 기본적인 모형화 요소로 채택한 KWM에 기반하여 개발되어 다양하고 복잡한 예외규칙 등을 지닌 비즈니스 프로세스 자동화에 장점을 지닌다. 또한, 인공지능의 추론기관에 기반한 워크플로우 엔진 개발 접근법을 제시한다. 즉, K-WFMS는 대용량의 워크플로우 인스턴스 정보를 저장하기 위한 데이터베이스 시스템(database system)과 비즈니스 규칙에 기반한 과업 순서화(task scheduling) 및 업무 담당자 매칭(role resolution) 등의 구현을 위한 지식기반 시스템(knowledge-based system)을 통합함으로써 인공지능 기법이 워크플로우 엔진 개발에 효과적으로 활용될 수 있음을 보인다.

둘째, K-WFMS를 통하여 추론엔진에 기반한

워크플로우 시스템이 예외상황 처리 및 워크플로우 모형의 변화관리에 장점을 지님을 보인다. WFMS의 사전적 유연성과 관련하여서는 기반이 되는 워크플로우 모형인 KWM 을 이용하여 다양하고 복잡한 형태의 비즈니스 규칙을 표현하는 것이 가능하므로, 사전적으로 예측이 가능한 예외상황을 정의하고 사전에 대비할 수 있다. 기존의 워크플로우 시스템들은 이러한 다양하고 복잡한 예측 가능한 예외상황을 사전에 표현하고 처리하는데 있어 부적절한 것으로 알려져 있다[Klein and Dellarocas, 2000]. 사후적 유연성을 높이기 위하여 K-WFMS는 워크플로우 스키마 정보를 사전에 컴파일(compile)하여 실행하는 방식이 아니라 실행시점에 해석(interpretation)하는 방식을 선택함으로써, 실행 도중에 워크플로우 스키마를 변경하는 것을 허용한다. 또한, 워크플로우 스키마(workflow schema)상에 변화가 일어났을 경우, 워크플로우 모형의 유효성(validity)을 위반하지 않음을 보장하기 위한 변화관리 장치들을 제공한다.

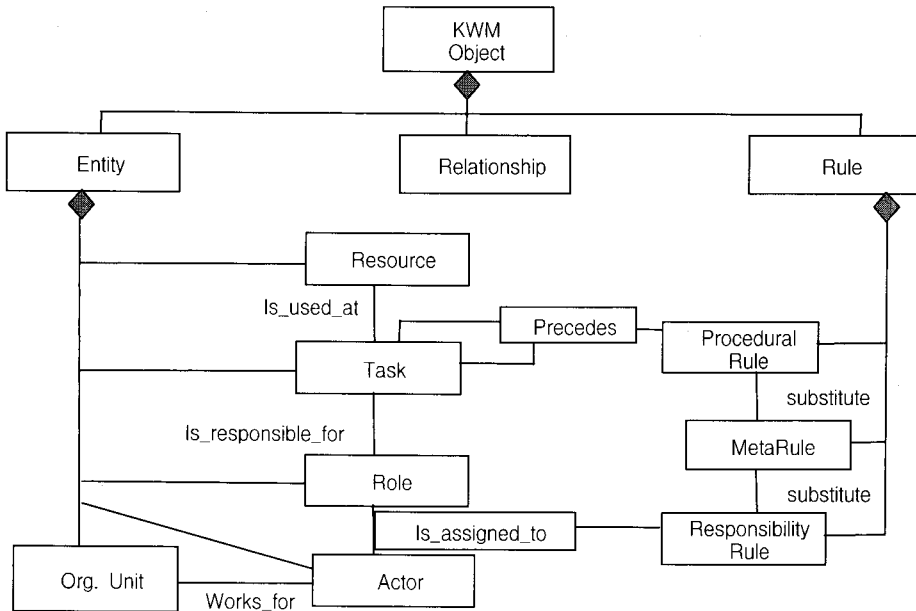
본 논문의 구성은 다음과 같다. 2절에서는 K-WFMS의 기반모형인 KWM을 통한 워크플로우 지식표현 방안에 대하여 간단히 설명하고, 워크플로우 모형화 시점에서 예측가능한 예외상황을 처리하기 위한 방안에 대하여 설명한다. 3절에서는 K-WFMS에서의 데이터베이스 시스템과 지식기반 시스템의 통합을 통한 추론 엔진에 기반한 워크플로우 엔진 개발 방안에 대하여, 그리고 4절에서는 K-WFMS의 사후적 유연성을 향상시키기 위한 동적인 워크플로우 스키마 변화관리방안에 대하여 설명한다. 5절에서는 기존의 워크플로우 자동화 시스템 개발을 위한 접근법 및 워크플로우 스키마 변화관리 방안에 대하여 살펴보고, 본 논문에서의 접근법과의 차이점을 설명한다. 마지막으로 6절에서 본 논문의 의의를 정리한다.

## II. K-WFMS에서의 사전적 예외상황 처리

본 논문에서는 모형화 시점에서 예측가능한 예외상황을 사전적 예외상황으로, 예측불가능한 예외상황을 사후적 예외상황으로 정의한다. 이러한 예외상황들을 얼마나 잘 처리할 수 있는가는 사용하고자 하는 워크플로우 모형이 제공하는 모형화 요소의 특성에 크게 의존한다. 일반적으로 예외상황을 처리하기 위해서는 해당 예외상황 발생의 인식, 처리 루틴(routine)의 파악, 그리고 처리 루틴 적용 후 결과의 감지 및 파생효과 분석의 3단계를 따르게 된다 [Klein and Dellarocas, 2000]. 따라서, 예외상황 처리에 효과적인 워크플로우 모형은 이러한 요소들을 충분히 표현할 수 있어야 한다. 본 절에서는, K-WFMS의 기반이 되는 워크플로우 모형인 KWM에 대하여 간단히 설명하고, 예제를 통하여 사전적 예외상황 처리 방안에 대하여 설명한다.

KWM에서는 워크플로우를 조직의 상황적 요소(organizational context)와 관련하여 정의한다. 이를 위하여 조직의 상황적 요소를 가장 잘 표현한다고 할 수 있는 비즈니스 규칙(business rule)을 워크플로우 표현을 위한 기본 요소로 사용한다. 이러한 비즈니스 규칙의 집합으로서 워크플로우를 보는 관점은 자연스럽게 지식기반의 워크플로우 모형화를 가능하게 한다. 비즈니스 프로세스 관련 비즈니스 규칙의 집합체로 해석되는 워크플로우는 상호 관련된 규칙들의 연쇄적 유발(chaining)에 의하여 워크플로우 실행이 통제된다. 따라서, 워크플로우를 정의하기 위한 비즈니스 규칙 유형(business rule type)의 선택은 자동화하기 위한 대상 워크플로우의 다양성과 정밀성을 위해 중요한 요인이다.

조직적인 상황정보 표현을 고려한 워크플로



<그림 1> 워크플로우 지식표현을 위한 KWM의 메타모형

우 모형화를 위해서는 조직구조와 같은 구조적인 지식 및 프로세스 운영과 관련된 운영지식을 모두 표현할 수 있어야 한다. KWM에서는 프레임(frame)을 사용하여 워크플로우와 관련된 양 측면의 지식을 표현한다. 워크플로우 지식표현을 위한 KWM의 메타모형은 <그림 1>에서 보는 바와 같다. 최상위 수준에서 KWM 프레임은 크게 개체(entity), 관계성(relation-ship), 그리고 규칙(rule) 3가지로 구성된다. 개체 및 관계성 프레임은 워크플로우 및 조직의 구조적인 지식을 표현하기 위한 것이며, 규칙 프레임은 조직의 정책에 맞추어 워크플로우 내에서의 과업 스케줄링, 업무 담당자 선정, 문서 라우팅과 같은 운영적인 지식을 표현하기 위하여 사용된다.

한편, 규칙 프레임은 다시 3개의 하위 프레임으로 분류될 수 있다. 먼저, 절차적 규칙(procedural rule) 프레임은 워크플로우를 구성하는 과업들간의 수행 순서를 표현하기 위하여 사용된다. 워크플로우 관리 연합(WfMC: Work-

flow Management Coalition)에서는 과업들간의 라우팅 요소(routing construct)로서 직렬 라우팅(sequential routing), 병렬 라우팅(parallel routing), 조건부 라우팅(conditional routing), 그리고 순환 라우팅(iterative routing) 4가지가 필요함을 제시하고 있다[WfMC, 1996]. KWM에서의 절차적 규칙들의 집합은 이러한 4가지 라우팅 요소를 표현하는데 사용된다. 절차적 규칙이 과업들간의 수행 순서를 제어한다면, 책임성 규칙은 특정 과업을 담당할 담당자(actor)를 역할(role)을 통하여 결정하는 규칙을 표현하기 위하여 사용된다. 이러한 규칙에 의한 역할 및 담당자 대응방법(role resolution)은 사전에 특정한 역할에 대한 담당자를 지정할 수 없는, 즉 동적인 역할 및 행위자 대응이 필요한 워크플로우 모형화 및 실행에 유용하게 사용될 수 있다. 마지막으로, 메타 규칙(meta-rule)은 절차적 혹은 책임성 규칙들간의 대체 관계(substitute relationship)를 표현하기 위하여 사용된다. 이러한 메타규칙은 워크플로우

운영에 있어서의 여러가지 예외상황을 처리하는데 유용하게 사용될 수 있다. 즉, 정상적인 상황에 적용되어야 할 규칙과 특별한 워크플로우 인스턴스에만 적용되어야 할 규칙을 별도로 정의해 놓고, 이들간의 선택적인 적용 메커니즘을 메타규칙으로 표현함으로써 다양한 형태의 예외상황을 처리할 수 있다.

<그림 2>는 국내의 한 대학(KAIST)의 출장 승인 프로세스에 대하여 KWM을 이용하여 표현한 일부분을 보여주고 있다. 프레임 "from-supervisor-to-auditor"는 절차적 규칙의 인스턴스이며, 출장 기일이 7일 이상인 경우에는 감사를 거쳐야 함을 나타낸다. 프레임 "find-traveler's-supervisor"는 책임성 규칙의 인스턴스로서, 특정한 워크플로우 인스턴스에서 "Super-

visor"라는 역할을 담당할 사람을 찾기 위한 규칙을 표현하고 있다. 마지막으로, 관심있게 볼 프레임으로 "exception-for-supervisor"는 메타규칙의 인스턴스이다. 이 프레임은 "SOURCE-RULE"과 "TARGET-RULE" 두개의 슬롯(slot)을 지니고 있는데, 전자의 경우에는 정상적인 상황에서 적용되어야 하는 규칙을 나타내고, 후자의 경우에는 "CONDITION" 슬롯에 명세된 조건을 만족하는 워크플로우 인스턴스에 대하여 선택적으로 적용되어야 하는 예외 규칙을 포함한다. 그림에서는 출장자가 산하 연구소장인 경우에는 조직도 상의 상위부서장이 원장이지만, 부원장이 대신 상위부서장 승인을 수행하도록 지시하고 있다. 이와 같이, KWM에서 메타규칙은 절차적 혹은 책임성 규칙들 간의

<pre> Frame from-supervisor-to-auditor IS_INSTANCE_OF : procedural-rule ; PROCESS : business_trip_approval ; DESCRIPTION : "If a supervisor approve subordinator's trip request and the trip duration exceeds 6 days, the inspector should inspect the trip purpose"; PRE_TASK : approve_subordinator's_trip ; PRE_TASK_STATE : "completed" NEXT_TASK : inspect_trip_purpose ; CONDITION : (Trip_Request_Form (duration ?dur)) (test (&gt;= ?dur 7)) END-FRAME         </pre>	<pre> Frame find-traveler s-supervisor IS_INSTANCE_OF : responsibility-rule ; PROCESS : business_trip_approval ; DESCRIPTION : "Traveler's supervisor is one who works for the Department with manager position which the traveler belongs to"; ROLE : Supervisor ; ACTOR : WorkFor.actor_id ; CONDITION : (Traveler (Department ?dept-id)) (Department (dept_id ?dept-id) (mnger_pos ?m-pos)) (WorkFor (dept_id ?dept-id)(actor_id ?supervisor_id) (position ?m-pos)); END-FRAME         </pre>
<pre> Frame exception-for-supervisor IS_INSTANCE_OF : meta-rule ; PROCESS : business_trip_approval ; DESCRIPTION : "If a traveler is a director of an affiliated research institute, the supervisor is the vice President although his formal supervisor is the President".; SOURCE_RULE : find-traveler's-supervisor ; TARGET_RULE : find-vice-president ; CONDITION : (Traveler (T_Id ?t-id)) (WorkFor (actor_id ?t-id) (position "director-of-affiliated-research-institute)) END-FRAME         </pre>	

<그림 2> 메타규칙을 이용한 사전적 예외상황의 처리

선택적 수행 혹은 갈등해소(conflict resolution) 등의 통제 역할을 수행함으로써, 여러가지 예측할 수 있는 예외상황들을 처리하는데 유용하게 사용된다.

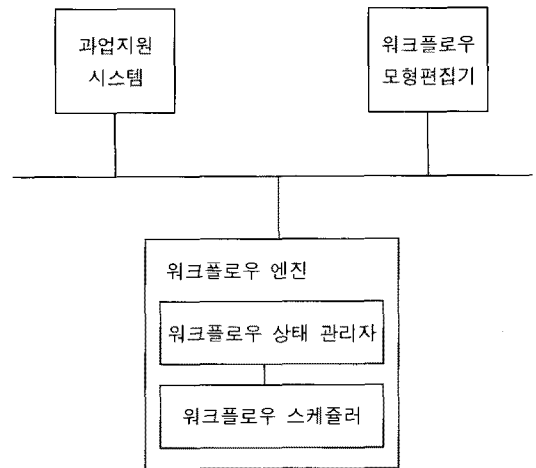
### Ⅲ. K-WFMS : 지식기반 지능형 워크플로우 자동화 시스템

#### 3.1 K-WFMS 아키텍처

K-WFMS는 <그림 3>에서 보이는 바와 같이 클라이언트-서버 구조(client-server architecture)를 이루고 있다. 클라이언트 부문에는 조직 구성원 들의 할당된 과업 수행 지원 및 처리한 과업의 후속상태 확인을 위한 과업지원시스템(Task Support System)과 워크플로우 모형화 작업을 위한 워크플로우 모형편집기(workflow model editor)로 구성된다. 과업지원시스템은 워크플로우에 참여하는 업무수행자의 효과적이고 효율적인 업무수행을 지원하기 위한 각종 자료수집, 조회, 업무상태 확인 등의 기능을 제공한다. 워크플로우 모형 편집기는 그래픽 아이콘(graphic icon)을 기반으로 자동화하고자 하는 대상 워크플로우 정보를 명세할 수 있도록 지원해준다. 워크플로우 모형 편집기의 중요한 기능 중의 하나는 사용자가 그래픽 모형화를 한 결과를 KWM의 프레임으로 자동 번역해 주는 것이다. 이는 사용자가 규칙 프레임들을 직접 입력함으로써 발생할 수 있는 명세오류나 규칙간의 상충을 미연에 방지해 준다.

워크플로우 엔진은 K-WFMS의 중앙서버에 위치한다. 워크플로우 엔진의 주된 기능은 생성된 워크플로우 인스턴스(workflow instance)들에 대한 과업의 순서화(task ordering), 문서 및 자원(resource)의 라우팅(routing), 역할과 실제 담당자 간의 매핑(mapping), 비즈니스 규칙

검사, 예외적인 워크플로우 인스턴스에 대한 특별한 처리 등이다. 각 워크플로우 인스턴스들에 대한 이력관리(history mangement) 또한 워크플로우 엔진의 주요한 기능으로서, 워크플로우 인스턴스가 진행됨에 따라 관련된 모든 정보들을 데이터베이스에 저장하고, 과업관리 시스템의 요청에 따라서 요구정보에 대한 수정 혹은 추출(retrieval) 서비스를 제공한다.



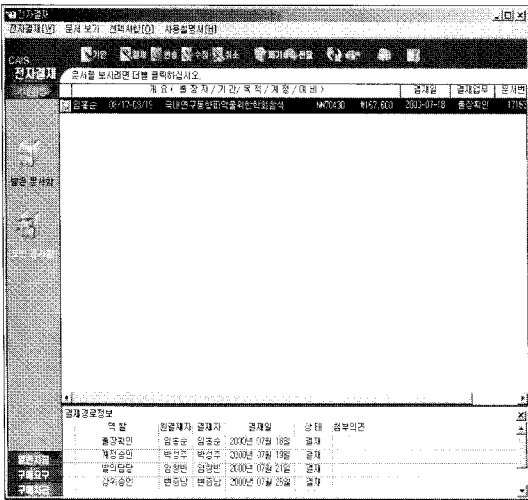
<그림 3> K-WFMS의 전체적인 아키텍처

K-WFMS 과업 관리 시스템은 파워빌더(Power-builder)로 구현이 되었고, 워크플로우 모형화 편집기는 C++로 구현이 되었다.

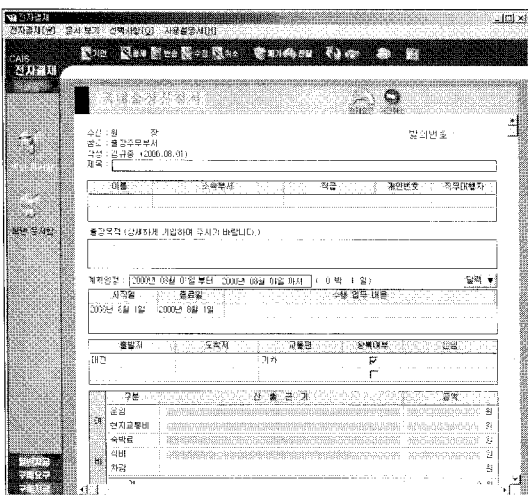
워크플로우 상태 정보는 관계형 데이터베이스인 Sybase에 저장되며, 워크플로우 상태 관리자는 스토어드 프로시저(stored procedure)로, 그리고 워크플로우 스케줄러는 Sybase사의 OpenServer Library와 전문가 시스템 셸(shell)인 CLIPS(C Language Integrated Production System)을 이용하여 구현되었다.

<그림 4>는 K-WFMS 과업지원시스템(전자결재 시스템)의 일부를 보여주고 있다. 과업담당자는 자신에게 할당된 과업(문서)들의 리스트를 확인할 수 있으며 (상위 윈도우), 처리

하고 난 후의 대상 과업(문서)의 현 상태를 추적할 수 있다 (하위 윈도우). 한편, <그림 5>는 2장에서 설명한 출장 승인 프로세스에서 출장 신청서를 기안하는 화면이다. 사용자는 과업 지원 시스템에서 제공하는 스마트 문서(smart form)를 이용하여 입력 오류 없이 출장신청서를 작성하여 제출할 수 있다.



<그림 4> K-WFMS 클라이언트 1: 결재경로 정보 화면



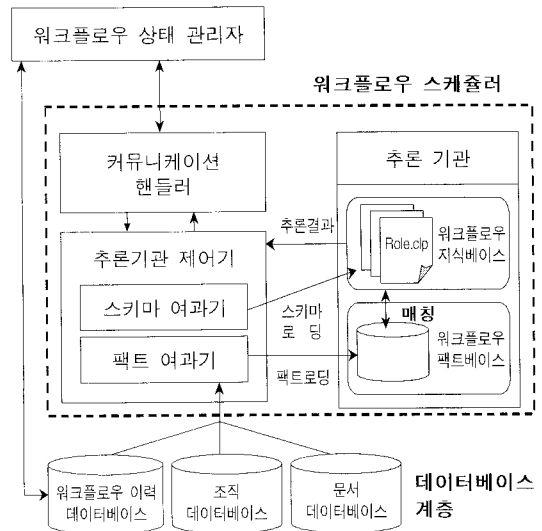
<그림 5> K-WFMS 클라이언트 2: 출장승인 프로세스 기안 화면

### 3.2 규칙기반 워크플로우 엔진의 구현

워크플로우 엔진의 과업 스케줄링 및 워크플로우 상태정보 관리 기능은 워크플로우 상태 관리자(workflow status manager)와 워크플로우 스케줄러(workflow scheduler)에 분담되며 이들간의 상호협동에 의하여 수행된다.

#### 3.2.1 워크플로우 상태 관리자

워크플로우 상태 관리자는 각 워크플로우 인스턴스의 이력(history) 정보를 수정하고 관리하는 기능을 담당한다. 워크플로우 클라이언트 시스템으로부터 특정 워크플로우 인스턴스의 상태변화 이벤트(event)를 접수하게 되면, 이러한 정보에 기반하여 워크플로우 상태 관리자는 워크플로우 인스턴스의 상태 정보를 수정한다. 접수된 이벤트가 특정 과업의 "완료"인 경우, 워크플로우 상태 관리자는 다음단계에 수행되어야 할 과업 및 과업 담당자를 찾기 위하여 워크플로우 스케줄러의 도움을 받는다. 워크플로우 스케줄러는 KWM으로 정의된 워크플로우 지식을 추론하여 다음 단계에서 수행



<그림 6> 워크플로우 스케줄러의 구성

되어야 할 과업 및 과업 담당자를 선정하고 결과를 워크플로우 상태 관리자에게 전달한다. 워크플로우 스케줄러로부터 다음단계의 과업 및 과업담당자가 전달되면, 워크플로우 상태 관리자는 해당 과업담당자의 업무할당테이블에 선정된 과업정보를 입력한다.

### 3.2.2 워크플로우 스케줄러

워크플로우 스케줄러가 워크플로우 상태 관리자로부터 스케줄링 요청을 받아 다음 과업 및 담당자를 찾는 과정이 <그림 6>에서 보여지고 있다. 앞의 KWM을 통하여 표현된 워크플로우 지식은 추론기관에서의 실행을 위하여 모두 CLIPS 언어 요소(language constructor)로 변환된다. KWM으로 표현되는 워크플로우 지식(스키마)의 평가(evaluation) 작업을 위해서는 대상 워크플로우 인스턴스와 관련된 조직 정보, 문서정보, 과거 이력정보 등이 제공되어야 한다. 이러한 정보들은 조직의 데이터베이스 시스템에 존재하므로, 추론기관으로 처리될 수 있는 팩트(fact)로 변환되어야 한다. 이러한 데이터의 추론기관의 내부 팩트로의 변환작업은 팩트 여과기 (fact filterer)에 의하여 처리된다. 팩트 여과기의 역할은 조직의 데이터베이스로부터 특정 워크플로우 인스턴스를 위한 데이터를 추출하여 추론기관의 팩트베이스에 제공하는 것이다. <그림 7>에서는 <그림 2>에서 정의된 KWM 규칙프레임을 CLIPS 규칙으로 변환한 예를 보여주고 있다.

먼저, 커뮤니케이션 핸들러(communication handler)는 워크플로우 상태 관리자와의 통신을 담당한다. 상태 관리자로부터의 스케줄링 요청 메시지를 접수하여 해석하고, 추론기관으로부터의 결과를 해석하여 다시 상태관리자에게 전달하는 역할을 한다.

추론기관 제어기는 상태 관리자로부터 요청에 따라 해당 워크플로우 인스턴스의 스케줄링

```
(defrule rp8
  (Task (task_id "Approve_Subordinator's_Trip")(state "Completed"))
  (WF_DomTripMaster (Duration ?dur))
  (test (>= ?dur 7))
  =>
  (assert (Task (pretask_id "Approve_Subordinator's_Trip")
    (task_id "Inspect_Trip_Purpose")(state "Initiated"))))
(defrule find-supervisor
  (Task (pretask_id ?pr_task)(task_id "Approve_Subordinator's_Trip")
  (state "completed"))
  (WF_DomTripTraveller (T_Id ?t_id)
  (T_Department ?dept))
  (Department (dept_id ?dept)
  (mnger_pos ?m-pos))
  (WorkFor (dept_id ?dept)
  (actor_id ?supervisor_id)(position ?m-pos))
  =>
  (assert (ActorMapping (pretask_no ?pr_task)(task_id "Approve_Subordinator's_Trip")
    (actor_id ?supervisor_id))))
```

<그림 7> 절차 규칙과 책임성 규칙 프레임을 CLIPS 규칙요소로 변환한 예.

을 위해 새로운 쓰레드(thread)를 생성하고 추론기관의 지식베이스(knowledge base)와 팩트베이스(fact base)의 내용을 구성한다. 추론기관 내부의 지식베이스와 팩트베이스의 구성은 스키마 여과기와 팩트 여과기에 의하여 수행된다. 스키마 여과기는 현재의 추론기관에 현재 진행중인 워크플로우의 스키마에 해당하는 규칙 프레임을 지식베이스에 입력시키고, 팩트여과기(Fact filterer)를 통하여 추론을 위해 필요한 fact를 데이터베이스에서 추출하여 팩트베이스에 저장시킨다. 팩트여과기는 지식기반시스템과 데이터베이스 간의 느슨한 형태의 통합(loosely coupled integration)을 구현한다. 즉, 추론이 시작되기 전에 사전에 필요한 데이터를



정하고 팩트베이스에 입력한 후에 추론을 시작하는 형태이다. 이러한 느슨한 형태의 통합은 시스템의 성능(performance) 측면에서 장점을 갖지만, 특정 워크플로우 인스턴스 스케줄링을 위해 필요한 데이터의 범위를 사전에 정의하기 어려운 문제가 존재한다. 이 문제를 해결하기 위하여 K-WFMS의 팩트 여과기는 워크플로우 유형별로 조직 개체들과의 관계성(relationship)에 기반하여 참여범위를 이끌어낸다. 이러한 사전에 정의된 참여범위에 의하여 팩트여과기는 특정 워크플로우 인스턴스 별로 데이터 추출을 위한 SQL문을 생성하고 이의 결과를 팩트파일로 변환하여 팩트베이스에 로드(load)한다.

## IV. K-WFMS에서의 동적인 변화관리

### 4.1 동적인 변화의 유형

사전에 예측할 수 없는 예외상황이 특정 워크플로우 인스턴스의 실행 중간에 발생하는 경우에는 향후의 해당 워크플로우 인스턴스의 진행 경로가 사전에 정의된 워크플로우 스키마에서 벗어나도록 허용해 주어야 한다. 이는 동적인 워크플로우 스키마의 변화를 의미하며, 기존의 대부분의 WFMS는 워크플로우 인스턴스의 생성 시점에 워크플로우 모형을 컴파일하여 향후 실행경로를 결정하므로 실행시점에서의 동적인 변경이 어렵다[Bogia and Kaplan, 1995] [Joeris, 1999].

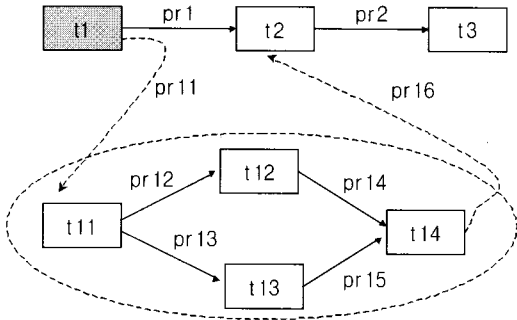
워크플로우 스키마에 대하여 효과적인 동적인 변화관리를 위해서는 크게 두 가지 기능이 필요하다. 첫 번째는 사전에 정의된 워크플로우 스키마의 변경없이 특정한 워크플로우 인스턴스에 대하여 사용자가 자신의 환경에 맞게 워크플로우 스키마를 변경하도록 허용하는 것이다. 이를 통하여 사전에 워크플로우 스키마를 정의할 수 없을 경우, 일단 대강의 워크플로우

스키마(rough workflow schema)를 가지고 워크플로우를 시작한 후에 실행시점에서 각 워크플로우 인스턴스 별로 워크플로우 스키마를 완성해 나가는 것이 가능하다. 두 번째는 모든 워크플로우 인스턴스들이 시작되어 실행되고 있는 중간에 이들이 공통으로 참조하는 워크플로우 스키마를 변화시킬 수 있도록 허용하는 것이다.

### 4.2 K-WFMS에서의 동적인 변화관리

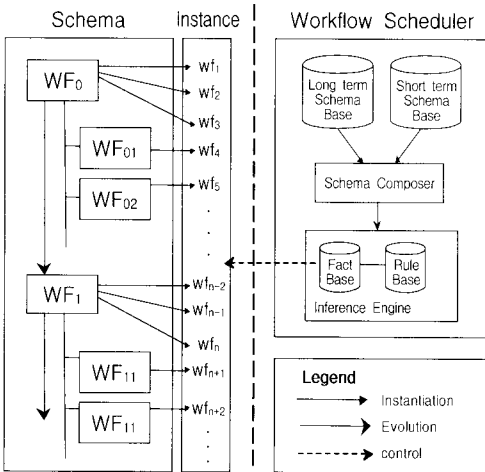
K-WFMS는 워크플로우 인스턴스의 경로선택에 있어서 생성시점에 모든 경로를 워크플로우 모형의 컴파일링을 통해 일괄 결정하는 방식이 아니고, 각 진행 단계마다 관련된 규칙 프레임을 해석하여 새로운 경로를 선정하는 방식이므로 동적인 변화 관리에 장점을 지닌다. 즉, 워크플로우 인스턴스가 특정 과업을 완료하고 다음 단계에서 수행되어야 할 과업을 선정함에 있어, 사전에 정의된 규칙 프레임을 새로운 환경에 맞는 규칙 프레임으로 대체하여 경로를 다르게 가져갈 수 있는 것이다. <그림 8>에서는 이러한 동적인 변화의 예를 보여주고 있다. 그림에서는 현재 과업 t1이 완료되고 다음 단계의 과업을 선정하는 상황에서, 사전에 정의된 과업 t2를 수행하기 전에 새로운 서브 워크플로우를 수행하도록 절차적 규칙집합 PR이 PR'로 변경되었음을 보이고 있다.

한편, K-WFMS에서는 앞에서 말한 두 가지 형태의 동적인 변경을 지원하기 위하여 워크플로우 스키마 정보를 저장하는 지식베이스를 <그림 9>과 같이 구성하고 있다. 그림 중앙의 굵은 점선을 기준으로 왼쪽은 워크플로우 스키마와 인스턴스간의 관계를 표현하고 있다. 워크플로우 스키마간에는 진화(evolution)관계와 주종(slave-master)관계 두 가지가 존재한다. 진화 관계는 하나의 워크플로우 스키마가 새로운 버전으로 진화하는 관계를 보여준다. 즉, 워크



PR = {pr1, pr2}  
 PR = {pr11, pr12, pr13, pr14, pr15, pr16, pr2}

<그림 8> 워크플로우 스키마의 동적인 변경 (PR -> PR')



<그림 9> 워크플로우 스키마, 인스턴스, 그리고 엔진과의 관계

플로우 스키마가 WF<sub>0</sub>에서 WF<sub>1</sub>으로 진화관계에 의하여 변경되는 경우에는 새로운 워크플로우 인스턴스는 WF<sub>1</sub>에 정의된 규칙에 기반하여 라우팅이 이루어진다 (<그림 9>에서 워크플로우 인스턴스 wf<sub>1</sub>, wf<sub>2</sub>, wf<sub>3</sub>, wf<sub>n-2</sub>, wf<sub>n-1</sub>, 그리고 wf<sub>n</sub> 등이 해당된다). 한편, 주종 관계는 특정 워크플로우 인스턴스를 위하여 기존의 워크플로우 스키마가 변경되는 경우에 유용하다. 즉, 워크플로우 스키마 WF<sub>01</sub>은 WF<sub>0</sub>와 주종 관계를 이루며, 이 스키마는 특정 워크플로우 인

스턴스인 wf<sub>4</sub>를 라우팅하기 위해서만 사용된다. K-WFMS에서는 진화를 위한 워크플로우 스키마 변경은 시스템 관리자에게만, 그리고 주종 관계를 통한 워크플로우 스키마 변경은 권한을 지닌 특수 사용자들에게만 허용이 된다.

<그림 9>에서 가운데 굵은 점선을 기준으로 오른쪽 쪽에서는 <그림 6>에서 보인 워크플로우 엔진의 구조 중에서 워크플로우 지식베이스에 대한 부분을 자세하게 보여주고 있다. 워크플로우 지식베이스는 크게 장기적 워크플로우 스키마베이스(long term schema base)와 단기적 워크플로우 스키마베이스(short term schema base) 두 가지로 구분하여 관리한다. 전자에는 진화관계를 통하여 변경되는 워크플로우 스키마 정보를, 후자에는 주종관계를 통하여 변경되는 워크플로우 스키마 정보를 저장한다. 워크플로우 실행 시점에서 워크플로우 스케줄링 요구가 전달되는 경우에는, 스키마 구성기(schema composer)가 해당 인스턴스에 적용되어야 하는 스키마 정보를 선택적으로 추출하여 추론기관에 입력함으로써 다음 단계에 실행될 과업 및 담당자를 결정한다.

### 4.3 동적인 변화관리의 유효성 검증

동적인 변화로 인한 워크플로우의 구조적인 변화 이외에 워크플로우의 조건부 라우팅 혹은 역할/담당자 매핑 등 워크플로우 인스턴스 운영에 영향을 미치는 비즈니스 규칙의 변화는 관리가 더욱 어렵다. 즉, 특정 규칙의 변화는 이와 연관된 다른 규칙들에 영향을 미치며 따라서 이러한 관련 규칙을 찾아내고 적절한 수정을 가해야 한다. 또한, 이러한 특정 변화로 인한 후속작업으로 인해 전체 워크플로우 스키마의 유효성(validity)이 훼손되지 않았는지를 검증할 수 있어야 한다.

워크플로우 스키마의 변화관리는 워크플로우 모형 편집기와 워크플로우 엔진과의 유기적

인 상호작용에 의하여 처리된다. K-WFMS의 워크플로우 스키마 DB에는 워크플로우를 표현하고 있는 KWM 프레임 정보뿐만 아니라, 프레임 간의 종속성에 대한 정보를 술어(predicate) 형태로 저장하고 있다. <표 1>에서 보는 바와 같이 KWM 프레임 간의 종속성을 표현하는 술어는 크게 3가지 유형으로 분류된다.

개체간 종속성 술어는 관계성 프레임에 의하여 표현되는 것으로, 특정 자원이 어떤 과업에서 사용되고 있는지, 특정 업무수행자는 어떤 부서에서 어떤 직위를 지니고 있는지 등의 조직의 상황정보(organizational context information)를 표현한다. 이러한 술어는 조직 구성원의 이직 혹은 신규채용, 조직 부서체계의 변동 등의 변화가 있을 때 유용하게 사용된다. 두 번째 술어 유형은 규칙간 종속성을 표현하는 것이다. 이는 실행(firing)되는 시점이 같은

규칙 프레임들간의 종속성을 의미한다. 즉, 워크플로우 진행단계의 특정 시점에서 여러 규칙프레임들 중에서 오직 하나의 규칙프레임만 실행이 되어야 한다거나(XOR-firing), 모든 규칙프레임들이 동시에 실행되어야 한다(AND-firing)는 등의 지식을 표현한다. 또한, 메타규칙 프레임은 다른 규칙프레임들간의 대체관계를 표현하는 특별한 형태의 종속성을 표현하고 있는데, Substitute라는 술어에 의하여 표현된다. 이러한 술어는 워크플로우 스키마 상에서 특정 규칙이 변화되었을 경우, 어떤 규칙들이 영향을 받게 되고, 그 영향의 유형은 어떤 것인지를 파악하는데 사용된다. 마지막으로, 규칙과 개체간의 종속성 술어 유형이 있다. 규칙과 개체간 종속성은 과업개체 프레임간의 스케줄링, 과업담당자 선정을 위한 역할과 업무수행자 간의 매핑에 사용되는 규칙프레임과 과업개체 프레임 혹은 역할개체 프레임 간의 관계를

<표 1> KWM 프레임간의 종속성을 표현하는 프레디케이트

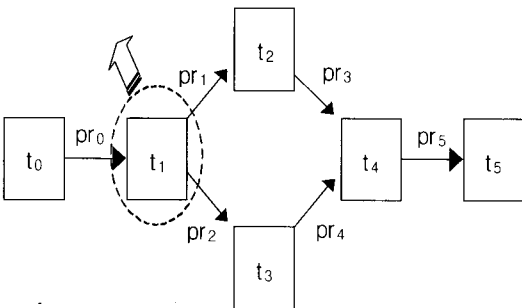
유형	술어	의미
개체간 종속성	$IS\_A(o1, o2)$ $SUBPART\_OF(o1, o2)$ $works\_for(a, u, p)$	개체프레임 $o1$ 이 개체프레임 $o2$ 로부터 상속 개체프레임 $o1$ 이 개체프레임 $o2$ 의 부품 행위자 개체프레임 $a$ 가 조직구성요소 개체프레임 $u$ 에 직위 $p$ 를 가지고 일하고 있음
	$used\_at(re, t)$ $responsible\_for(ro, t)$ ...	자원개체프레임 $re$ 가 과업개체프레임 $t$ 에 사용됨 역할개체프레임 $ro$ 가 과업개체프레임 $t$ 를 수행함 ...
		...
규칙간 종속성	$XOR-firing(r_1, r_2, \dots, r_n)$	규칙프레임 $r_1, r_2, \dots, r_n$ 중에서 하나만 실행될 수 있음
	$AND-firing(r_1, r_2, \dots, r_n)$	규칙프레임 $r_1, r_2, \dots, r_n$ 이 모두 실행되어야 함
	$Substitute(r_1, r_2, rm)$	메타규칙 프레임 $rm$ 이 규칙프레임들의 집합 $r_1$ 을 규칙프레임들의 집합 $r_2$ 로 대체함
규칙과 개체간 종속성	$Splitted(t_1, T_s, Rp_s)$	워크플로우 컨트롤이 과업개체프레임 $t_1$ 이 끝난 후 절차규칙 프레임 집합 $Rp_s$ 에 의하여 과업개체프레임들의 집합 $T_s$ 로 전이됨
	$Joined(t_1, T_j, Rp_j)$	워크플로우 컨트롤이 과업개체프레임들의 집합 $T_j$ 으로부터 절차규칙프레임 집합 $Rp_j$ 에 의하여 과업개체프레임 $t_1$ 으로 전이됨
	$Precedence(t_1, t_2, rp)$	워크플로우 컨트롤이 절차규칙프레임 $rp$ 에 의하여 과업개체프레임 $t_1$ 에서 과업개체프레임 $t_2$ 로 전이됨
	$Role-charge(ro, rr, t_i)$	책임성 규칙프레임 $rr$ 이 역할개체프레임 $ro$ 의 과업개체프레임 $t_i$ 담당자를 찾음

표현한다. 이러한 지식은 워크플로우 스키마에서 특정 과업의 추가 혹은 기존 과업의 삭제로 인해 영향을 받는 과업, 역할 그리고 규칙프레임을 파악하는데 사용된다.

이상의 종속성 술어는 워크플로우 스키마 변화관리를 위한 변화전파규칙(change propagation rule)에서 사용된다. 변화전파규칙은 이벤트-조건-행위(ECA: Event Condition Action) 규칙의 형태를 취하게 되는데, 이벤트절에는 워크플로우 스키마 상에 적용되는 각종 오퍼레이션(operation)을 명세하며, 조건절에서는 해당 오퍼레이션으로 인하여 영향을 받게 되는 프레임들을 찾기위한 조건을 위에서 정의한 종속성 술어를 이용하여 명세한다. 마지막으로, 행위절에서는 이벤트절의 오퍼레이션이 가해진 후에도 전체 워크플로우 스키마의 유효성(validity)이 만족될 수 있도록 하기 위해 조건절에서 찾아진 프레임들에 대하여 가해져야 하는 오퍼레이션을 명세한다.

```

ON : DeleteForkTask (t1)
IF : Precedence(t0, t1, pr0) Splitted(t1, {t2, t3},
    {pr1, pr2}) Responsible_For(r0, t1)
    Role-charge(r0, rr, t1)
THEN : Rp = Rp {pr0}; update pr1.PRE_TASK
    = t0; update pr2.PRE_TASK = t0;
    T = T {t1}; Rr = Rr {rr}; RF
    = RF {(r0, t1)}
    
```



<그림 10> 워크플로우 스키마에서 포크 과업(fork task) 삭제시 변화 전파 규칙

<그림 10>은 워크플로우 스키마에서 포크(fork) 과업 t<sub>1</sub>(즉, 완료 후에 워크플로우 통제권(control)이 여러 과업으로 분할되는 과업)을 삭제할 경우, 정의될 수 있는 변화전파규칙의 예를 보여주고 있다. 이 경우에는, 포크 과업이 삭제되는 경우에는 대상 포크과업의 이전과업 t<sub>0</sub>이 대신 포크과업으로 자동으로 지정되며, 포크과업과 종속성을 갖는 절차적 규칙들(pr<sub>0</sub>, pr<sub>1</sub>, pr<sub>2</sub>)에 대하여 적절한 수정작업이 이루어 지도록 유도된다. 또한, 삭제되는 과업의 역할과 담당자를 매칭하는 책임성 규칙 프레임 r<sub>r</sub>, 그리고 삭제되는 과업에 대한 담당할 역할의 책임성 (r<sub>0</sub>, t<sub>1</sub>)이 삭제된다.

## V. 관련 연구

### 5.1 워크플로우 자동화 시스템 아키텍처

워크플로우 자동화 시스템 개발을 위한 접근법은 크게 데이터베이스 시스템에 기반한 것과 이메일 시스템에 기반한 것 두 가지를 들 수 있다. 데이터베이스 시스템에 기반한 워크플로우 자동화 시스템은 구현이 쉽고, 중앙통제방식이어서 워크플로우 상태추적(workflow tracking)이 쉽다는 장점이 있다. 또한, 데이터베이스 시스템의 복구기능을 이용한 워크플로우 인스턴스의 에러처리가 가능하다는 것도 주요한 장점이라고 할 수 있다[Casati, Ceri, Pernici, and Pozzi, 1996; Eder and Liebhart, 1995; McCarthy and Sarin, 1993; Rusinkiewicz and Shet, 1993]. 반면, 워크플로우 트랜잭션(workflow transaction)이 하나의 중앙서버에 집중됨으로 인해 규모성(scalability)이 낮다는 단점이 있다.

한편, 이-메일 시스템에 기반한 워크플로우 시스템의 경우 기본적으로 분산된 워크

플로우 트랜잭션 처리가 가능하므로 규모성에서 장점을 지닐 수 있다. 반면, 워크플로우 통제권(workflow control)이 분산되어 있으므로, 특정한 워크플로우 인스턴스의 상태추적이 실시간으로 이루어지기가 어려운 단점을 지니게 된다[Alonso et al., 1994; Chang and Scott, 1996; Jennings et al., 1996]. 최근의 에이전트 기반의 워크플로우 자동화 시스템들은 대부분이 이러한 메시지 중심의 접근법의 하나로 볼 수 있다.

본 논문에서 제시되는 K-WFMS는 기본적으로 데이터베이스 시스템에 기반한 워크플로우 자동화 시스템으로 분류해 볼 수 있다. 그러나, 다양한 형태의 비즈니스 규칙을 구현하기 위해서 지식기반시스템을 도입한다. 따라서, 데이터베이스 시스템과 지식기반 시스템의 통합을 통한 지능형 워크플로우 자동화 시스템의 아키텍처를 제시하고 있다.

## 5.2 워크플로우 자동화 시스템에서의 변화 관리

WFMS에서의 변화관리는 최근에 그 중요성이 더욱 부각되어 활발히 진행 중이다[Klein and Dellarocas, 2000; Casati et al., 1998; Reichert and Dadam, 1998; Dellen et al., 1997; Bogia and Kaplan, 1995]. Klein & Dellarocas[2000]는 워크플로우 설계자와 참여자가 예외상황 처리를 효과적으로 할 수 있도록 프로세스 라이브러리와 예외상황 라이브러리를 제시하였다. 많은 조직에서 사용하는 비즈니스 프로세스의 공통적인 요소를 근원적인 프로세스 타입으로 정의하고, 이러한 근원적인 프로세스 간의 계층적인 구조를 제시하였다. 또한, 이러한 근원적인 프로세스 별로 발생할 수 있는 근원적인 예외상황 간의 계층구조를 제시하고, 이러한 예외상황을 처리하는 데 지침이 될

수 있는 지식을 체계화 하였다. 그러나, 이러한 체계를 실제 문제에서 구현할 수 있는 시스템을 제시하지는 않고 있다. Joeris[1999]는 예외상황을 사전적 예외상황과 사후적 예외상황으로 구분하고, 이러한 예외상황처리를 위한 메커니즘을 모형화 시점 및 실행시점에서 가능하도록 하고 있다. 그러나, 이 연구에서는 대상으로 하는 워크플로우를 단순히 과업 간의 순서화된 집합으로 해석하고 있어, 본 연구에서 제시하는 바와 같이 비즈니스 프로세스의 근거가 되는 비즈니스 규칙에 대한 명세 장치를 제시하지 않고 있다. Casati 등[1998]은 워크플로우 스키마(workflow schema) 수정을 위한 프리미티브(primitive)들을 제시하였으며, 이러한 워크플로우 스키마가 변경되었을 경우 실행중인 워크플로우 인스턴스들을 관리하기 위한 방안에 대하여 제시하였다. Reichert와 Dadam[1998]은 구조적 정확성과 일관성을 보장하며 현재 실행중인 워크플로우 구조를 변경시킬 수 있는 완전성과 최소성을 갖는 변화 오퍼레이션(change operation) 집합(ADEPT<sub>flex</sub>)을 제시하였다. Dellen 등[1997]은 소프트웨어 플래닝(planning)과 같이 애드-혹(Ad-hoc)한 특성을 갖는 워크플로우 지원을 위한 시스템으로 CoMo-Kit을 제시하였다. CoMo-Kit 시스템에서는 사전에 정의될 수 없는 워크플로우를 실행 중에 새롭게 재정의 할 수 있도록 지원하는 특성을 가지고 있다.

이상의 변화관리 접근법은 대부분이 워크플로우의 구조적인 지식들의 변화를 그 관리 대상으로 삼고 있다. 즉, 워크플로우 스키마내에서 특정한 과업의 삭제 혹은 새로운 과업의 추가 등과 같은 변화에 초점이 맞추어져 있으며, 예외규칙이나 조직상황 정보에 의한 조직 구성원예의 과업 할당 규칙과 같은 워크플로우 운영과 관련된 변화관리는 언급이 되지 않고 있다.

## VI. 결 론

비즈니스 프로세스 자동화를 위해 제시되고 있는 WFMS들은 워크플로우 모형을 통하여 대상 조직에서 실행되어야 하는 비즈니스 프로세스를 표준화하고 모든 참여자들(workflow participants)이 이러한 틀에 맞추어 업무를 수행하도록 강요하고 있다. 그러나, 실제 작업현장에서 업무를 수행하는 업무 수행자들은 업무수행에 있어서 자율성을 요구하고 있으며, 보다 적응적인 WFMS가 되기 위해서는 업무수행자들의 이러한 자율성을 최대한 보장해 줄 수 있어야 한다.

본 논문에서 제시한 K-WFMS는 이러한 적응적인 WFMS 구현을 위해 지식기반의 접근법이 유용함을 보이고 있다. 특히, 복잡하고 다양한 비즈니스 규칙에 근거하고 예외상황이 자주 발생하는 비즈니스 프로세스 자동화에 장점을 지닌다. 작업현장에서 발생하는 다양한 예외상황을 사전적 예외상황과 사후적 예외상황으로 구분하여, 사전적 예외상황을 처리하기 위한 방안으로 지식기반의 워크플로우 모형인 KWM

에서는 지식표현 기법으로 많이 사용되는 규칙에 기반한 모형화 요소를 제시하고 있다. 사후적 예외상황을 처리하기 위해서는, 실행시점에 사용자가 워크플로우 스키마를 변경할 수 있도록 워크플로우 경로선정 및 담당자 설정을 컴파일 방식이 아닌 관련 규칙의 해석방식을 이용하고 있다. 이를 위하여, K-WFMS는 데이터베이스 시스템과 지식기반 시스템을 통합한 아키텍처에 근거하여 구현되었다.

한편, K-WFMS에서는 비즈니스 규칙 등 KWM 모형화 요소간의 종속성을 효과적으로 관리함으로써 특정 요소에 대한 변화 오퍼레이션이 발생하였을 경우, 해당 변화로 인해 영향을 받는 다른 요소들을 쉽게 찾아낼 수 있으며, 워크플로우 모형의 유효성을 위반하지 않도록 하기 위한 적절한 대응을 할 수 있도록 지원한다.

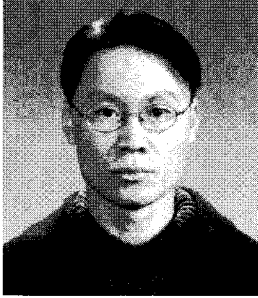
K-WFMS는 현재 한국과학기술원의 캠퍼스 정보 시스템인 CAIS(Campus Advanced Information System)와 통합되어 교육, 연구, 도서, 행정 프로세스 자동화를 위한 기반으로서의 역할을 수행하고 있다[Park, 1998].

## 〈참 고 문 헌〉

- [1] 이하빈, 박성주, 유연한 워크플로우 자동화 시스템을 위한 지식기반 접근법, 한국경영과학회, 춘계학술대회, 계명대학교, 1999.
- [2] Alonso, G., Mohan, C., Guenthoer, R., Agrawal, D., El Abbadi, A., Kamath, M. Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management, *Proc. IFIP WG8.1 Working Conference on Information Systems for Decentralized Organizations*, Trondheim, August 1995.
- [3] Appleton, D.S. Business Rules: The Missing Link," *Datamation*, October, pp. 145-150, 1984.
- [4] Bogia, D.P. and Kaplan, S.M., Flexibility and control for dynamic workflows in the worlds environment. *Proc. of the Conference on Organizational Computing Systems (ACM 1995)*, 1995.
- [5] Casati, F., Ceri, S., Pernici, B., and Pozzi, G., Deriving active rules for workflow enactment, In *Proceedings of DEXA 96*,

- Zurich(CH), 94-110, 1996.
- [6] Casati, F., Ceri, S., Pernici, B., and Pozzi, G., Workflow evolution, *Data and Knowledge Engineering*, 24, 211-238, 1998.
- [7] Chang, J.W. and Scott, C.T., Agent-based workflow: TRP Support environment (TSE), *Computer Networks and ISDN Systems*, 28, 1501-1511, 1996.
- [8] Dellen B., Maurer, F., and Pews, G., Knowledge-based techniques to increase the flexibility of workflow management, *Data and Knowledge Engineering*, 23, 269-295, 1997.
- [9] Eder, J. and Liebhart, W., The workflow activity model: WAMO, *Proceedings of the third international conference on the cooperative information systems*, 87-98, 1995.
- [10] Fisher, Layna, Excellence in Practice Volume II: Innovation and Excellence in Workflow Process and Knowledge Management, *Future Strategies Inc.*, Florida, USA, 2000.
- [11] Lee, Ha Bin, Kim, Jong Woo, and Park, Sung Joo, "KWM: Knowledge-based Workflow Model for Agile Organization," *Journal of Intelligent Information System*, 13 (3), 261-277, 1999.
- [12] Joeris, Gregor., Defining Flexible Workflow Execution Behaviors, *Technical Report*, University of Bremen, 1999.
- [13] Klein, M. and Dellarocas, C.A Knowledge-based Approach to Handling Exceptions in Workflow Modeling. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*. Vol. 9, No. 3-4, pp. 399-412, 2000
- [14] McCarthy, D.R. and Sarin, S.K., Workflow and Transactions in InConcert. *Bulletin of the Technical Committee on Data Engineering*, 16 (2), 1993.
- [15] Moriarty, T. Business Rule Analysis. *Database Programming & Design*, February, pp. 66-69, 1993.
- [16] Park, J.Y. and Park, S.J., IBPM: An integrated systems model for business process reengineering, *Systems Engineering*, 1(3), 159-175, 1998.
- [17] Park, S.J., Development of future management information system for intelligent campus, *Project Report*, KAIST, Korea, 1998.
- [18] Park, S.J., Kim, J.W., Lee, H.B., Cho, K.H., and Kim, G.J., Open workflow automation system in client/server environment. *Proc. of the Korea Society of Management Information Systems (149-158)*. Seoul, South Korea, 1995.
- [19] Reichert, M. and Dadam, P., ADEPT<sub>flex</sub>-Supporting dynamic changes of workflows without losing control, *Journal of Intelligent Information Systems*, 10, 93-129, 1998.
- [20] Rusinkiewicz, M. and Shet, A., On transactional workflows. *Bulletin of the Technical Committee on Data Engineering*, 16 (2), 1993.
- [21] WfMC, *Workflow management coalition terminology and glossary (WfMC-TC-1011)*. Technical Report, Workflow Management Coalition, Brussels, 1996.

◆ 저자소개 ◆



이하빈 (Lee, Ha Bin)

현재 영국 BTextact Technologies (구, BT Lab)의 지능형 시스템 연구실에서 전임연구원으로 재직중이다. 한국항공대학교 항공경영학과에서 이학사(1991), 한국과학기술원(KAIST) 경영학과에서 공학석사 (1993), 한국과학기술원 테크노경영대학원에서 경영정보학 분야로 공학박사(1999)를 취득하였다. 1999년도에는 한국과학기술원 가상기업 및 지식경영연구센터에서 1년간 박사 후 연수과정으로 지냈으며, 2000년부터 배재대학교 전자상거래학과 전임강사로 소속되어 있다. 주요 관심분야는 에이전트 기반의 협동작업, 정보시스템 개발 방법론, 비즈니스 프로세스 공학 등이다.



박성주 (Park, Sung Joo)

현재 한국과학기술원 테크노경영대학원장으로 재직중이며, 한국 시물레이션 학회장을 역임하고 있다. 서울대학교 산업공학과에서 공학사(1973), 한국과학기술원 산업공학과에서 공학석사(1975), 미시간 주립 대학교에서 시스템 공학 공학박사(1978)를 취득하였다. 주요 관심분야는 경영정보시스템, 이비즈니스에서의 에이전트 응용, 지식경영 등이다.

◆ 이 논문은 2000년 10월 11일 접수하여 1차 수정을 거쳐 2001년 8월 4일 게재확정되었다.