

협업 워크플로우에서의 인적오류 제어를 위한 하이브리드 모델링 도구

(A Hybrid Modeling Tool for Human Error Control in Collaborative Workflow)

이 상 영[†] 유 철 중^{**} 장 옥 배^{**}
(Sang-Young Lee) (Cheol-Jung Yoo) (Ok-Bae Chang)

요 약 협업 워크플로우 관점에서 비즈니스 프로세스는 기업 내외의 애플리케이션들과 인적 자원들과의 유기적인 결합을 통하여 협업 프로세스를 신속하고 유연하게 수행할 수 있도록 지원하여야 한다. 현재 기업 업무에 대한 시스템 의존도가 높아지고는 있지만, 인간은 여전히 핵심적인 역할을 차지한다. 즉 워크플로우 모델링에는 이러한 인간의 역할이 강조되어야 하며 인간 의사결정 과정 자체를 분석하여 인적오류를 제어할 수 있는 구조가 반드시 필요하다. 또한 액티비티간의 협업을 통하여 빠르고 효율적인 커뮤니케이션이 구축되어 궁극적으로는 액티비티들이 목표 프로세스에 연류되어 결과물에 대한 품질이 향상되도록 하여야 한다. 이에 본 논문에서는 협업 워크플로우에서 발생할 수 있는 인적오류를 제어하기 위해 GEMS(Generic Error Modeling System) 모델을 적용한다. 또한 하이브리드 기반의 모델링 방법을 통한 인적오류 제어 방법을 제시한다. 아울러 이러한 기반에서 협업 워크플로우를 모델링할 수 있는 도구를 설계하고 구현한다. 이와 같은 모델링 방법을 사용하면 인간 자체의 특성을 고려함으로써 인적오류를 제어할 수 있는 워크플로우 모델링이 될 수 있도록 지원할 수 있다.

키워드 : 비즈니스 프로세스, 워크플로우 모델링, 인적오류, 하이브리드

Abstract Business process should support the execution of collaboration process with agility and flexibility through the integration of enterprise inner or outer applications and human resources from the collaborative workflow view. Although the dependency of enterprise activities to the automated system has been increasing, human role is as important as ever. In the workflow modelling this human role is emphasized and the structure to control human error by analysing decision-making itself is needed. Also, through the collaboration of activities agile and effective communication should be constructed, eventually by the combination and coordination of activities to the aimed process the product quality should be improved. This paper classifies human errors can be occurred in collaborative workflow by applying GEMS(Generic Error Modelling System) to control them, and suggests human error control method through hybrid based modelling as well. On this base collaborative workflow modeling tool is designed and implemented. Using this modelling methodology it is possible to workflow modeling could be supported considering human characteristics has a tendency of human error to be controlled.

Key words : business process, workflow modeling, human error, hybrid

1. 서 론

· 본 논문은 2003년도 한국과학재단 지역대학우수과학자 지원사업에 의해 연구되었음(과제번호 : R05-2003-000-12236-0)

† 학생회원 : 전북대학교 전산통계학과
sylee230@chonbuk.ac.kr

** 종신회원 : 전북대학교 전자정보공학부 교수
cjyoo@chonbuk.ac.kr
okjang@chonbuk.ac.kr

논문접수 : 2003년 8월 27일

심사완료 : 2003년 12월 24일

전자상거래는 이제 기업 입장에서 선택 사항이 아닌 살아남기 위한 필수 사항으로 인식되고 있다. 전자상거래를 수행하는 대부분의 기업들은 독자적으로 재화와 용역을 제공하는 것이 아니라, 고객과 공급업체, 비즈니스 파트너들과 서로 협력하고 정보를 공유하는 일련의 비즈니스 프로세스(business process)를 수행한다[1].

비즈니스 프로세스는 작업 순서들의 집합 형태인 프로세스를 기업이나 조직의 목표를 달성하기 위한 프로세스의 형태로 발전시킨 것이다. 프로세스를 소프트웨어

공학 관점에서 보면 서비스를 제공하거나 프로덕트(product)를 만들 때 그것이 소프트웨어 개발이나 보고서 작성 등의 작업(task)을 수행하기 위하여 따르는 일련의 과정을 말한다. 즉 작업 순서의 집합(a set of ordered tasks)을 프로세스라 하고, 기업 입장에서 어떤 목표를 달성하는 결과를 얻기 위하여 수행하는 작업, 제약 조건 및 자원을 포함하는 일련의 과정을 비즈니스 프로세스라 한다[2]. 이러한 비즈니스 프로세스는 조직의 일련의 활동들에 대해 일관성과 구조를 강요하여 효율성을 높일 수 있기 때문에 매우 중요하다[3].

비즈니스 프로세스를 기반으로 하는 워크플로우(workflow)는 목표를 달성하는 과정을 수행하기 위하여 발생하는 일련의 작업들의 흐름을 뜻한다. 즉 기업의 비즈니스 목표를 성취하기 위하여 사전에 정의된 규칙(rule)에 따라 조직 구성원들 사이에 이루어지는 문서와 정보 또는 작업 절차를 자동화시키는 관점을 말한다. 이러한 워크플로우를 통하여 기업의 비즈니스 프로세스 성능을 향상시킬 수 있다. 또한 워크플로우 관리 시스템(WfMS: Workflow Management System)은 이러한 워크플로우를 형성하기 위하여 준비된 작업들의 수행을 지시, 조정, 감시하는 것으로서 프로세스 정의를 해석하고 워크플로우 참여자(participant)들과의 상호작용을 통하여 해당 애플리케이션을 효과적으로 실행할 수 있게 도와준다[4]. 그리고 기업들이 상호간의 거래 문서, 정보 및 거래 과정 등을 비즈니스 프로세스로 설계하여, 협업 프로세스를 전자적으로 수행하고 진행 상황을 파악 및 통제할 수 있는 방법 등을 제공한다[5-7].

또한 비즈니스 프로세스의 생명은 유연하고 적응성이 있는 지속적인 개선을 통해 실제 환경과의 차이를 줄이는 것이 관건이다[8]. 더욱이 기업간 협업 워크플로우 관점에서 비즈니스 프로세스는 기업 내외의 애플리케이션과 인적 자원들과의 유기적인 결합을 통하여, 협업 프로세스를 신속하고 유연하게 수행할 수 있도록 지원해야 한다[9]. 현재 기업의 업무가 점점 자동화되고 시스템에 의존하고 있지만, 인간은 여전히 핵심적인 역할을 차지한다. 그 결과 인간의 개입을 요구하는 워크플로우 시스템은 비즈니스 프로세스 관리(BPM: Business Process Management)에서도 중심 요소가 된다. 이러한 관점에서 워크플로우 시스템은 인적요소의 역할에 맞게 적절한 업무를 할당하고 전달하는 기능을 수행하는 것으로 인식한다. 즉 인간의 의사결정 과정에서 발생 가능한 인적오류를 제어할 수 있는 방법이 요구된다. 보통 인간간의 협업을 통하여 빠르고 효율적인 커뮤니케이션(communication)이 구축되며 궁극적으로는 구성원들이 목표 프로세스에 연류되어 결과물에 대한 품질이 향상된다. 또한 직접 모여서 작업해야 되는 것 등의 제

거를 통해 전반적인 비용이 절감되고 문제를 신속히 해결할 수 있는 등의 시간을 절감할 수 있다. 대개의 협업의 경우 보통 목표지향적인 활동을 하는 것이 일반적인데 이러한 경우 관련된 부분들은 목표를 중심으로 기여하게 된다. 아울러 효율적인 비즈니스 프로세스의 관리를 위해서는 프로세스 참여자들간의 지속적인 정보의 흐름이 필요하고 작업 흐름의 유연한 통합, 협업 작업의 지원을 위한 정보의 공유 및 적합한 표준이 요구된다[10,11].

본 논문에서는 상호 커뮤니케이션을 강조하는 하이브리드(hybrid) 기반의 모델링 방법을 사용한다. 또한 목표지향적인 워크플로우 시스템에서의 협업 수행시 인적오류를 제어할 수 있도록 GEMS(Generic Error Modeling System)모델을 분류하고 특성을 제시한다. 또한 GEMS 모델을 하이브리드 기반의 모델링 방법에 적용한다. 아울러 이러한 기반에서 비즈니스 프로세스를 모델링할 수 있는 도구를 XPDL을 기반으로 설계하고 구현한다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 논문과 관련된 기존의 연구에 대하여 알아보고, 3장에서 협업 워크플로우 모델링에서의 인적오류 제어를 위한 GEMS 모델 및 하이브리드 기반의 모델링 방법 적용에 대해 제시한다. 4장에서 이러한 협업 워크플로우 모델링 도구에 대한 설계를 하고 5장에서는 협업 워크플로우 모델링 도구에 대한 구현 및 평가를 한다. 마지막 6장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

비즈니스 프로세스의 중요성이 1960년대에 처음 언급된 이래 지난 십수년동안 기업의 실제 중요한 점을 파악하고 이를 개선하는데 유용한 모델링 방법들과 도구들이 연구되어 왔다[9,12]. 비즈니스 프로세스는 우선 기업 통합에 있어서 핵심 요소라는 관점에서 출발한다. 비즈니스 프로세스에 따라 관련 기업들이 비즈니스의 수행에 필요한 정보를 교환하고 B2B 인터페이스 및 프로세스를 통합을 하는 것이다[13]. 즉, 기업의 비즈니스를 수행하기 위하여 비즈니스 데이터를 포함하는 조직화된 비즈니스 이벤트의 교환을 통하여 기업과 거래 당사자들과 전자적으로 연결하는 것이다. 현재 이와 관련된 분야에서 통합을 위한 방법론 및 중요요소 등에 대한 많은 연구가 수행되고 있다[8]. 더욱이 최근 들어 비즈니스 프로세스의 정의 자체에 대한 연구에서부터 다양하면서도 세밀하고 체계적인 방향으로 진행되고 있다[14]. 또한 이러한 기업의 협업을 워크플로우로 통합하기 위해서는 적합한 표준과 기반 구조가 필요하다. 이와 관련된 표준에는 WfMC(Workflow Management Coali-

tion) 같은 워크플로우 관련 단체들이 표준화를 선도하며 제안하는 프로세스 정의 표준인 XPDL(XML Process Definition Language), Wf-XML(Workflow-XML) 등이 있다. 또한 최근 각광받고 있는 웹 서비스를 중심으로 하는 비즈니스 프로세스 표준에는 WSCL(Web Service Choreography Language), BPML(Business Process Modeling Language) 및 BPEL4WS(Business Process Execution Language for Web Service) 등이 있다. 이 중 XPDL은 공동된 프로세스 형식 정의의 교환을 위하여 WfMC에서 제안하는 XML 기반의 워크플로우 프로세스 정의 언어이다[11].

2.1 비즈니스 프로세스 및 워크플로우 모델링

보통 비즈니스 프로세스는 그림 1에서 보는 바와 같이 하나 또는 그 이상의 협업이 존재할 수 있으며 각 협업은 하나 또는 그 이상의 거래로 구성된다[15].

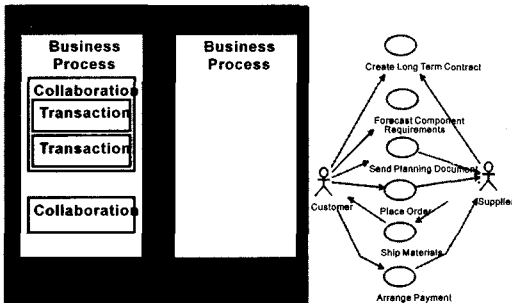


그림 1 비즈니스 프로세스, 협업 및 거래의 개념적 관점

이러한 비즈니스 프로세스에 대한 모델링 방법은 그동안 여러 도메인에서 제안되었고, 비즈니스 프로세스에 적합한 표기법들과 도구들이 연구되어 왔다. 이러한 연구들을 통해 다양한 비즈니스 프로세스 모델링 방법들이 제안되었다[12,16]. 일반적으로 비즈니스 프로세스 모델링에 대한 접근 방법은 정보, 조직, 절차 및 행위 등의 네 가지 관점으로 분류될 수 있다[17]. 이러한 접근 방법 중 절차 관점이나 행위 관점이 지금까지 비즈니스 프로세스 모델링에서의 주된 관점으로 논의되고 연구되어 왔다. 왜냐하면 정보 관점은 너무 정적이어서 프로세스 자체나 이의 동적인 측면을 파악하기가 어려운 단점이 있고, 조직 관점은 너무 많은 변수들이 존재하여 이를 프로세스 자체로 표현하기에 부적절한 단점이 존재하기 때문이다. 한편 절차나 행위 관점에서 비즈니스 프로세스 모델링에 접근하는 가장 중요한 문제는 시간 중심의 동적인 측면을 반영하는 것이다. 이러한 동적인 측면을 표현하는 관점들은 입출력 흐름(input/output flow), 에이전트(agent), 상태 흐름(state flow) 및 워크플로우 관점 등 크게 네 가지로 분류할 수 있다[18]. 먼

저 입출력 흐름 관점은 행위에 따라 소비되거나 생산되거나 변화되는 수동적인 참여자들에 초점이 맞추어져 있다. 이러한 흐름은 다이어그램으로 표현되는데, 여기서 행위는 노드로 표현된다. 이러한 관점의 대표적인 메타 모델로는 IDEF0(Integration Definition for Function Modeling)가 있다. 그리고 에이전트 관점은 에이전트가 자신의 일을 위해서 실행하는 순서에 초점을 둔다. 이런 종류의 흐름을 표현하는 전형적인 표기법은 RAD(Role-Activity Diagram)와 UML의 협력 다이어그램(collaboration diagram) 등이 있다. 또한 실세계 프로세스에 발생하는 변화에 초점을 맞춘 상태 흐름 관점은 일반적으로 상태 전이 다이어그램(state-transition diagram)으로 표현되는데, 상태 흐름을 부분적으로 표현할 수 있다. 마지막으로 본 논문에서 대상으로 하는 워크플로우 관점은 행위의 시간상 순서에 초점을 맞춘다.

보통 워크플로우 관점에서의 흐름은 다이어그램으로 표현되는데, 여기서는 전이(transition)가 화살표로 표현된다. 그리고 각 노드들은 하나 이상의 행위들에 대한 결과 수행을 의미한다. 이러한 종류의 흐름을 표현하기 위한 전형적인 표기법은 IDEF3, UML의 액티비티 다이어그램(activity diagram), 페트리 넷트(petri-net) 등이 있다. 이 중에서 액티비티 다이어그램과 페트리 넷트가 워크플로우 모델링에 널리 이용되고 있다[12].

일반적으로 워크플로우에서 비즈니스 프로세스를 묘사하기 위한 모델링 방법은 크게 세 가지로 구분될 수 있다[10]. 먼저 비즈니스 프로세스의 목적을 고객 만족 증진에 두고 여러 액션 모델과의 커뮤니케이션이라는 개념에서 출발하는 커뮤니케이션 기반의 방법이 있다. 이 방법은 고객과 행위자간에 커뮤니케이션을 중심으로 워크플로우 상의 모든 액션을 수행한다. 그러나 이 방법은 복잡한 작업과 참여자간의 관계 묘사가 난해한 단점이 있다. 그리고 인간간의 행위를 모델링하기 보다 작업 자체를 모델링하는데 초점을 둔 액티비티 기반의 방법이 있다. 이러한 방법은 프로세스에 연관된 작업을 중심으로 수행되는데 고객 만족과 같은 목적은 달성하기 어려운 단점이 존재한다. 마지막으로 커뮤니케이션 기반과 액티비티 기반의 방법을 혼합한 하이브리드 기반의 방법이 있다. 이 방법은 현재 연구가 진행되고 있고 아직까지 발표된 유용한 도구나 방법은 없다. 하이브리드 기반의 방법은 작업 중심의 워크플로우 시스템에서 인간의 역할을 중요시하여 실제적인 프로세스를 이룩하고 예외사항이나 오류를 줄일 수 있는 적합한 방법이다. 따라서 본 논문에서는 이러한 하이브리드 기반의 모델링 방법을 적용한다.

2.2 인적오류

인간 의사 결정시의 오류를 제어하기 위한 인지 심리

에 대한 연구로는 인간 정보처리 각 단계에서 생길 수 있는 오류를 분류하고 이에 대한 방지를 다루는 방법론에 대한 연구들이 주로 수행되어 왔다[19]. 이러한 연구 중 가장 폭 넓게 인정받고 있는 정보처리 모델을 사용한 인적오류 분류 체계로는 Rasmussen의 분류 방법(SRK 모델)이 있는데 그는 표 1에서 보는 바와 같이 세 가지로 인적오류를 분류하였다[20]. 먼저 기술 기반(skill-based) 오류는 무의식적 행동 관계 및 저장된 행동 양식에 의해 제어되며 관계적 상황에서 숙련된 작업자에게 적절한 것으로 위험을 인식하지 못하고 감각과 운동에 의한 오류로 주로 실행 오류이다. 그리고 규칙 기반(rule-based) 오류는 환경을 잘 알고 지식도 많으나 잘못된 작업 방법 등에 의한 오류이다. 마지막 지식 기반(knowledge-based) 오류는 목표와 관련하여 실행을 계획해야 하는 독특하고 친숙하지 않은 상황에서 발생한다. 주로 문제공간에서의 자원의 한계나 불완전하고 부정확한 지적 모델에 의해 발생한다[21].

이러한 Rasmussen의 SRK 모델에 대한 응용 연구로 이를 보다 객관화시키고 실제 적용시 용이하도록 수정한 GEMS 모델이 있다[22]. GEMS 모델은 목표지향적인 활동에 접합한 모델로 본 연구에서는 협업 워크플로우 모델링에 수정하여 적용한다.

한편 워크플로우 모델링 분야에서도 오류의 효율적인 제어가 WFMC의 성공을 위한 중요한 요소임을 인식하고 이에 대한 연구가 진행되어 왔다[23]. 그러나 이와 같은 워크플로우 시스템에서의 오류 제어에 관한 기존 연구는 주로 시스템적인 접근을 통하여 실패 발생시 대처 방법에 대한 연구가 주로 수행되어 왔다[24,25]. 즉 이러한 연구들은 워크플로우 시스템의 환경이나 특성을 기반으로 하는 예외처리 규칙 수준의 연구나 도구 사용 측면에서의 편의성 측면의 연구로, 근원적인 인지적 모델을 기반으로 하는 사려 깊은 연구나 어떤 모델링 도구의 구현 방법을 제시하는 실제적인 연구는 아니다.

표 1 Rasmussen의 인적오류 분류 체계(SRK 모델)

분류	특성
기술 기반 오류 (Skill-based Error)	- 최근에 사용한 방법을 사용 - 자주 사용하는 방법을 사용 - 잘못된 특성을 공유 등
규칙 기반 오류 (Rule-based Error)	- 작업 선행 요소를 잘못 사용 - 지식 활용 오류 - 부적합한 방법 사용 - 너무 단순화시킴 등
지식 기반 오류 (Knowledge-based Error)	- 작업 부하 과다 - 주제 모호 - 유추에 의한 추론 - 추론 논리 오류 - 불완전한 정신 모델 - 부정확한 정신 모델 등

이에 본 논문에서는 하이브리드 개념의 모델링 방법을 통하여 목표지향적인 워크플로우 시스템에서의 협업 수행시 궁극적으로 인적오류를 제어할 수 있도록 GEMS 모델을 통한 제어 방법을 제시한다.

3. 인적오류 제어를 위한 GEMS 모델 및 하이브리드 기반의 모델링 방법 적용

3.1 인적오류 제어를 위한 GEMS 모델 적용

소프트웨어 측면에서 워크플로우 방향은 두 가지 다른 기원에서 출발하였다. 즉 인간 기반의 비즈니스 프로세스(people-based business process)와 규칙 기반의 자동 프로세스(rules-based automation process)이다 [9]. 먼저 인간 기반 워크플로우 프로세스는 팀이나 그룹에 속한 사람들이 쉽게 협업하고 작업 흐름을 원활하고 효율적으로 하는 것이 목적이다. 이와 같은 협업을 위한 도구에 있어 거의 모든 부분들이 인간에 의하여 결정되고 진행시키므로 인적오류가 차지하는 비중은 매우 크다. 이에 본 논문에서는 이러한 인적오류를 제어하기 위하여 가장 대표적인 인지 모델인 Rasmussen의 SRK 모델을 적용한다. 특히 SRK 모델을 일반화시키고 적용을 용이하게 하기 위하여 GEMS 모델을 사용하여 워크플로우 시스템에 적용한다. 즉 워크플로우 모델링시 적용하여 모델링 방법과 인적오류간의 관계에 대한 지침을 제공하고 이에 대한 제어 방법을 제시한다. 우선 GEMS 모델에서의 기본 오류 타입을 표 2에서와 같이 인지 단계에 따라 세 가지로 구분한다.

표 2에서 보는 바와 같이 계획, 저장 및 실행을 수행하는 인지 단계에 따라 기본적인 오류 타입을 과오(mistakes), 간과(lapses) 및 실수(slips)로 분류하였다. 먼저 계획 단계에서는 목표 달성과 관계되는 프로세스를 수행할 때의 기본적인 오류인 과오로 분류하였고, 처한 환경의 복잡도에 따라 다시 지식 및 규칙 기반으로 분류하였다. 그리고 저장 및 실행 단계에서는 일상적인 행동 수행 동안에 발생하는 오류로 기술 기반 오류인 간과 및 실수로 분류하였다. 이러한 세 가지 유형인 지식, 규칙 및 기술 기반 수준 간에는 다음 그림 2와 같은 동적인 변화가 발생한다.

이와 같은 GEMS 모델을 워크플로우 시스템에 적용

표 2 인지 단계에 따른 기본 오류 타입 및 유형

인지 단계	기본 오류 타입	환경 및 오류 유형	
계획	과오(mistakes)	복잡	지식 기반
		단순	규칙 기반
저장 및 실행	간과(lapses) & 실수(slips)	기술 기반	

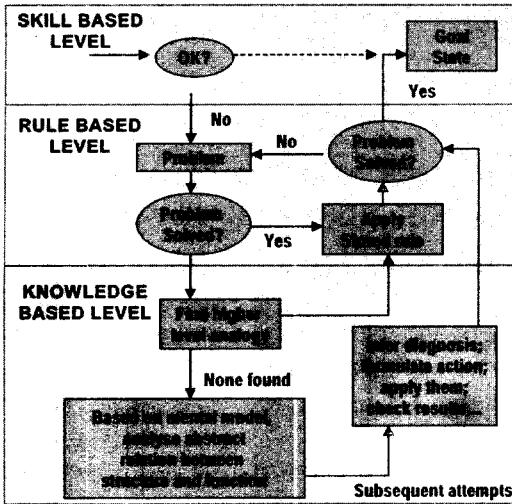


그림 2 GEMS 모델에서의 흐름도

하면 다음과 같다. 먼저 기술 기반 오류는 일반적으로 반복 수행시에 발생하는 일련의 행동 절차 중 하나를 생략하는 등의 간과나 실수에 해당하는 것이다. 이러한 경우를 제어하기 위한 방법은 워크플로우 시스템 자체에서 자동화 시키는 방법도 고려할 수 있지만 궁극적으로는 인터페이스 측면에 관련된 경우가 가장 많고 이를 개선하는 것이 가장 효율적이므로 본 논문에서는 대상으로 하지 않는다. 그리고 규칙 기반 오류는 익숙한 환경에서 사려 깊은 행위 중의 과오를 말하는 것으로 이러한 수준의 오류는 올바른 규칙을 가지고 적합한 실행을 하게 되면 제어 가능하다. 즉 워크플로우 시스템 자체가 가진 규칙에 따라 순서대로 진행되는 특성으로 제어 가능하고 특히 규정된 기호, 패턴 및 적합한 규칙을 통하여 접근하면 제어 가능하다. 마지막으로 지식 기반 오류는 어떤 노하우(now-how)나 규칙이 유용하지 못한 환경에서 즉 익숙하지 않은 상황에서 발생하는 오류를 말한다. 특히 본 논문에서는 워크플로우 특성을 고려하여 동적으로 변화하는 환경이나 복잡한 작업 영역에서 환경이 익숙하지 못할 때 발생하는 오류 유형으로 규정하고 GEMS 모델에서 제시하는 다음과 같은 두 가지로 구분한다. 즉 발생요인을 자원의 한계와 지적 모델의 불완전성 및 부정확성으로 구분한다. 이러한 오류의 발생 빈도는 타 수준에 비해 상대적으로 적으나 감지 및 제어하기 힘들고 오류 발생시 파급되는 영향효과가 큰 특징이 있다. 따라서 워크플로우 시스템에서 이러한 오류를 줄이려면 기본적으로 지식 기반 오류의 형태를 근원적으로 규정하고 상호간에 커뮤니케이션을 통한 대처 방안을 강구해야 한다.

3.2 하이브리드 기반의 협업 워크플로우 모델링을 통한 인적오류 제어

협업 워크플로우는 한 조직 내에서 발생하는 다양한 비즈니스 업무 흐름을 정의하고 수행하기 위한 효율적인 작업 환경을 제공한다. 그리고 각 부분들간의 협업을 통하여 지향하는 목표를 중심으로 모든 작업들이 이루어지는 특성을 가진다. 이는 관련 부분들간의 상호 협업을 강조하고 커뮤니케이션을 통한 목표 달성을 지향하는 하이브리드 방법에서의 전략과 일맥상통한다. 여기서는 인적요소가 차지하는 비중이 간과되어서는 안되며 특히 협업 워크플로우 시스템에서는 상대적으로 중요한 의미를 가진다.

3.2.1 하이브리드 기반의 모델링 방법

비즈니스 프로세스 모델링에는 자동적인 개념의 작업과 수동적인 개념이 있다. 자동적인 개념의 작업은 컴퓨터 시스템에 의해 자동적으로 수행되는 작업이고 수동적인 작업은 인간이 개입된 작업을 말한다. 즉 워크플로우 시스템의 참여자를 크게 두 가지로 구분하면 인간 혹은 자동화된 프로세스이다. 그림 3은 이러한 비즈니스 프로세스에 관련된 기본 개념들간의 관계를 보여준다 [11].

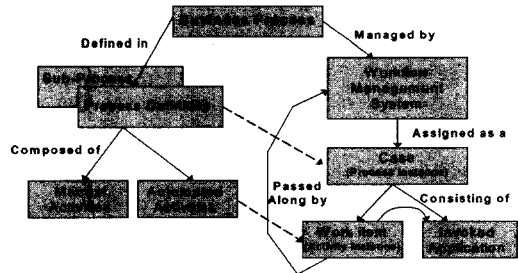


그림 3 비즈니스 프로세스 관련 개념도

그림 3에서 보는 바와 같이 비즈니스 프로세스는 세부적으로 서브 프로세스들로 구분될 수 있고, 이러한 프로세스들은 액티비티들로 이루어져 있다. 또한 액티비티들은 다시 인간이 개입하는 수동(manual) 액티비티와 인간 개입이 불필요한 자동(automated) 액티비티로 구분된다. 그리고 비즈니스 프로세스를 자동화시키는 등의 작업을 위하여 워크플로우 관리 시스템(WFMS)이 사용되며 여기서는 프로세스 인스턴스를 생성시키기 위해 프로세스를 설정한다.

본 논문에서 제시하는 하이브리드 기반의 워크플로우 모델링 방법은 이러한 개념들을 바탕으로 기본적으로 그림 4와 같은 계층 구조를 가진다.

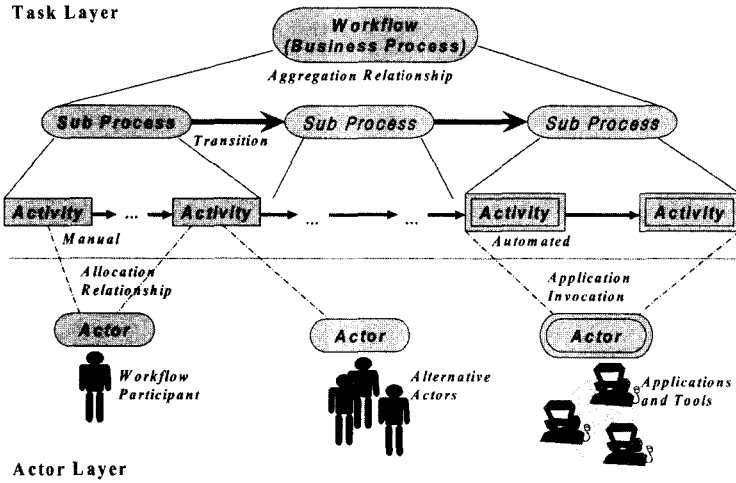


그림 4 하이브리드 기반의 워크플로우 모델링 구조

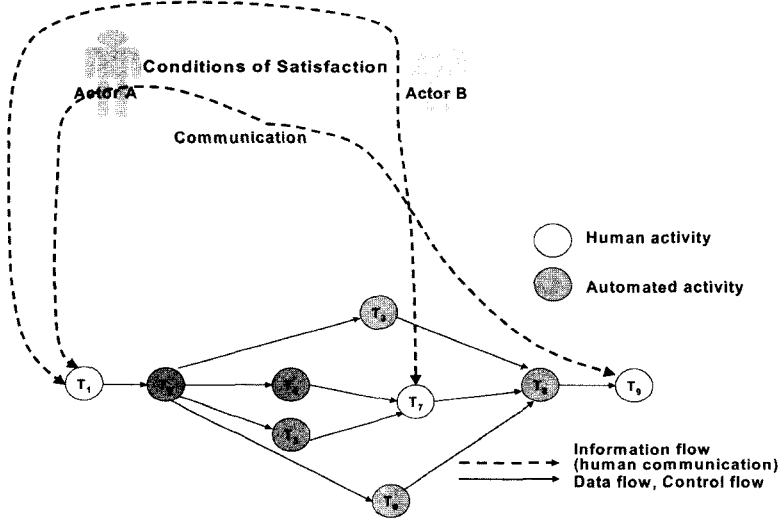


그림 5 하이브리드 기반의 워크플로우 모델링

그림 4에서 보는 바와 같이 레이어는 작업 레이어 (task layer) 및 액터 레이어(actor layer)로 구분된다. 먼저 작업 레이어는 워크플로우의 비즈니스 프로세스 영역이며 여기의 비즈니스 프로세스는 여러 개의 프로세스들로 구성되어 있다. 그리고 액터 레이어는 액터들의 영역이며 여기의 액터들은 프로세스 안의 액티비티들에 포함된다. 아울러 액터는 두 가지 형태로 구분되는데 먼저 인적 액터는 수동 혹은 반자동 액티비티를 수행하는 주체이고, 자동 액터는 컴퓨터 시스템에 의해 자동으로 수행되는 액티비티를 담당한다. 그리고 아래 그

림 5는 적용하고자 하는 하이브리드 기반의 방법을 나타낸다.

그림 5에서 보는 바와 같이 하이브리드 기반의 워크플로우 모델링 방법은 액티비티 기반 모델링 방법과 커뮤니케이션 기반 모델링 방법을 혼합한 개념이다. 이러한 구조에서 실제 진행은 상호간에 커뮤니케이션의 수행과 작업 흐름을 동시에 진행시켜야 한다. 특히 인적 액터가 담당하는 액티비티 작업의 경우 작업의 흐름에서 액티비티에 참여하는 액터들간의 커뮤니케이션이 반드시 필요하다. 일반적으로 비즈니스 프로세스 모델링에

서는 모델과 실제과의 괴리를 줄이는 것이 관건이다. 이러한 하이브리드 기반의 모델링 전략의 특징은 일반적인 작업의 흐름에 의한 업무 진행이 아닌 인적요소를 중시하는 상호 커뮤니케이션에 의한 협업을 강조한다. 또한 적시에 적합한 인프라를 이용해 목표를 달성시키는 것을 목표로 한다. 즉 작업 레이어 및 액터 레이어에 의한 액티비티 기반의 작업 흐름과 커뮤니케이션 기반의 정보 흐름을 분리함으로써, 작업의 흐름을 유지하면서 커뮤니케이션이 원활히 이루어지도록 하여 보다 실제적인 협업이 가능하도록 하는 장점을 가진다. 또한 이와 같은 하이브리드 기반의 모델링 방법의 구현에 있어 적절한 커뮤니케이션을 위한 경로의 설정 및 운용을 작업 흐름에 따른 액티비티의 진행에 결부시켜 구현하는 것이 관건이다.

3.2.2 인적오류 제어 방법

인간이 개입된 작업의 경우 협업시 필요할 때 마다 다른 액티비티간에 정보 교환을 통한 커뮤니케이션을 수행한다. 이러한 방법은 기업이 처한 환경 자체가 동적으로 환경이 변화하는 경우 더욱 효율적이고 실제적인 프로세스를 구축할 수 있다. 예를 들어 기업의 효율적인 공급 관리를 대상으로 한다면 해당 부분 및 액티비티간의 정보 공유를 통한 주문 및 물류 프로세스의 통합적 관리가 요구된다. 기존의 모델링 방법에 의한 프로세스의 경우 분화된 기능적 부문에서 각 기능의 경계에서 권한과 책임이 분리되고 정보 전달이 제한 받거나 지연되어 리드타임(lead time)의 증가 및 주문 충족 실패 등의 문제가 발생한다. 따라서 종래의 기능적 조직의 한계를 극복하여 프로세스를 효과적으로 관리하기 위해서는 기능간의 경계를 극복하여 기능간에 정보를 공유 및 전달하는 등의 통합 관리할 수 있는 비즈니스 프로세스가 필수적이다. 따라서 제시하는 하이브리드 방법을 통하여

인적오류를 제어할 수 있는 모델링을 수행한다.

인적오류를 제어하기 위한 하이브리드 기반의 모델링 전략을 다음과 같이 네 가지로 제시한다.

전략 1 : 목표를 중심으로 변화하는 환경에 대처 가능한 동적인 비즈니스 프로세스의 수행을 지향한다.

전략 2 : 작업 레이어 및 액터 레이어에 의한 작업(액티비티 기반) 흐름과 정보(커뮤니케이션 기반) 흐름을 분리함으로써 작업의 흐름을 진행시키면서 커뮤니케이션이 이루어지도록 하여 보다 실제적인 협업이 가능하도록 한다.

전략 3 : 인간이 개입된 액티비티(수동 혹은 반자동 액티비티)의 경우 커뮤니케이션을 통하여 오류를 제어하는 등의 효율성을 기할 수 있도록 커뮤니케이션 패턴을 첨가해서 사용한다.

전략 4 : 자동화된 액티비티의 경우 미리 규정되고 유효한 워크플로우 패턴을 사용하여 작업의 흐름에 따라 목표를 달성시키는 최적의 경로를 설정한다.

이를 바탕으로 인적오류 분류에 따른 제어 방법을 제시하면 다음 표 3과 같다.

• 기술 기반 오류 유형에 대한 제어

사용자가 인터페이스를 사용할 때 흔히 범할 수 있는 오류의 대부분이 기술 기반 오류에 속한다. 이 경우에는 인터페이스 측면에서 사용자 편의를 제공하는 등의 방법으로 오류를 최소화할 수 있다. 또한 협업 워크플로우 시스템에서는 가능한 수동 작업을 자동화된 작업으로 전환하는 등의 방법을 적용할 수 있다.

• 규칙 기반 오류 유형에 대한 제어

정해진 규칙 및 관습이라 여기는 절차 및 양태를 어김으로써 발생하는 오류로 적용 방법에 대한 불완전한 정의 및 잘못된 규칙에 의한 오류가 대부분을 차지한다. 이와 같은 오류는 근본적으로 워크플로우 자체의 시간

표 3 인적오류 분류 및 제어

협업 작업 수행 수준		인적오류 제어	
기술 기반 (slips & lapses)	slips	일상적인 행동 절차 수행동안 주의를 하지 않음으로써 발생하는 오류(attention failures)	가능한 수동 작업을 자동화된 작업으로 전환하고 사용자 인터페이스 측면의 고려
	lapses	순간 부주의를 함으로써 발생하는 오류(memory failures)	
규칙 기반 (mistakes)	대다수의 사람들이 규칙 및 관습이라고 여기는 절차 및 양태를 어김으로써 발생하는 오류		약속된 비즈니스 프로세스 표기법 및 유효한 워크플로우 패턴 등을 적용하여 제어
지식 기반 (mistakes)	<ul style="list-style-type: none"> • 요인 • 자원의 한계 • 문제 공간에서의 지적모델의 불완전성 및 부정확성 		하이브리드 방법을 적용한 커뮤니케이션 패턴을 첨가해서 작업 수행
	<ul style="list-style-type: none"> • 문제해결 유형(커뮤니케이션 패턴) • 정적 구성 • 반응하는 동적 구성 • 멀티 동적 구성 		

및 목표를 중심으로 작업을 체계적으로 진행시키는 특성을 이용하여 제어 가능하다. 또한 표준화되고 통용화된 방법을 적용해야 하는데 예를 들어 UML 같은 표준화된 모델링 언어를 사용하여 사용자, 개발자 및 설계자가 규칙 기반 오류를 최소화할 수 있도록 하여야 한다. 즉 약속된 비즈니스 프로세스 표기법 및 유효한 워크플로우 작업 패턴을 통하여 사용자들이 익숙한 환경에서 작업할 수 있도록 제공하는 방법으로 제어할 수 있다. 예를 들어 다음 그림 6과 같은 프로세스가 있을 경우 이를 잘못된 오류 패턴으로 규정하고 사용을 금지해야 한다.

그림 6에서 보는 바와 같이 이전 분기가 OR-Split로 분기한 다음 액티비티로 진행되고 다시 AND-Join 분기가 발생되면 논리적으로 무효한 오류 패턴임을 알 수 있다.

• 지식 기반 오류 유형에 대한 제어

지식 기반 오류는 인지적 제어가 가장 어려운 부분으로 인간 특성인 지적 능력이나 지식 축적 정도 등의 상태나 주위 환경 등을 종합적으로 고려하여야 한다. 이 경우에는 오류 발생 자체에 대한 예방은 어려우나 발생 가능한 오류를 분류하고 대처하는 노력이 필요하다. 특히 상호간에 정보의 공유를 통한 독자적인 판단에 의한 작업이 수행되지 못하도록 관련 액터나 액티비티간에 커뮤니케이션이 효율적으로 이루어지도록 하여야 한다.

본 논문에서는 앞에서 제시한 요인에 대응하는 지식 기반 오류를 범하는 문제 해결 유형을 세 가지로 구분하고 이를 커뮤니케이션 패턴으로 사용한다. 그림 7은

이러한 패턴을 나타낸다.

a) 정적 구성 패턴

커뮤니케이션 패턴 중 가장 단순한 패턴으로 액티비티간에 단순히 한쪽으로만 정보를 주는 피드백이 필요 없는 유형이다. 여기서는 문제 해결에 필요한 정보 전달을 통해 커뮤니케이션이 이루어진다. 예를 들어 변경 사항에 대해 관련 부문들에 전달하고 인지시켜야 하는 경우에 해당한다. 즉 다른 액티비티로부터 정보를 제공받거나 제공하는 경우에 해당하고 주로 자원의 한계에 의해 발생할 수 있는 관련 부문들과의 정보 교류 부재로 인한 오류를 제어할 수 있다

b) 반응하는 동적 구성 패턴

문제 해결에 꼭 필요한 정보를 응답받거나 정보를 교환하면서 협상하는 유형으로 반드시 피드백이 이루어진다. 질의하거나 정보 교환을 필요로 하는 액티비티의 경우에 해당되고 예를 들어 작업을 수행하는데 다른 부문에 정보 제공을 요구하거나 커뮤니케이션을 필요로 하는 경우에 해당된다. 주로 문제 공간에서의 커뮤니케이션의 비효율성에 기인하는 지적 모델에 대한 불완전성 및 부정확성을 제어할 수 있다.

c) 멀티 동적 구성 패턴

반응하는 동적 구성에 외부적인 환경 반응을 합친 개념으로 작업의 외적인 요소까지 고려하여야 하는 유형이다. 예를 들어 다른 부문에서 정보를 받고 관련된 부문에 커뮤니케이션이 이루어지는 경우에 해당된다. 즉 외부의 변화에 의해 커뮤니케이션 내용 및 결과가 변화하는 경우에 해당되며 주로 문제 공간에서의 지적 모델

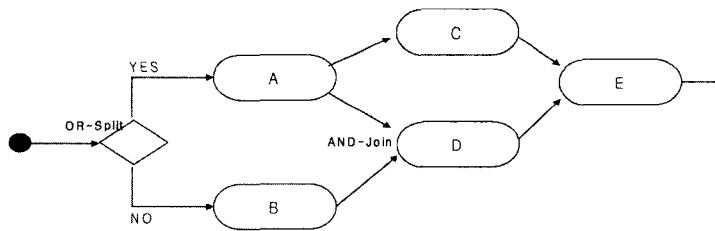


그림 6 잘못된 워크플로우 패턴의 예

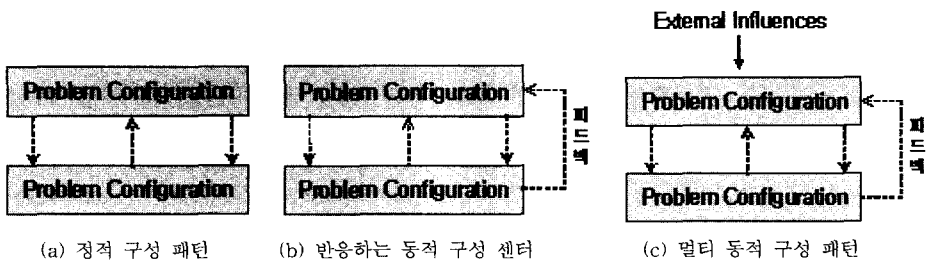


그림 7 커뮤니케이션 패턴 유형

에 대한 불완전성 및 부정확성을 제어할 수 있다.

이와 같이 세 가지 커뮤니케이션 패턴을 사용함으로써 지식 기반 오류의 원인인 자원의 한계나 문제 공간에서의 지적 모델의 불완전성 및 부정확성을 제어한다. 즉 인간이 개입된 수동 액티비티의 경우 반드시 정보 흐름이 포함된 커뮤니케이션 패턴을 첨가해서 사용하게 함으로써 지식 기반 오류를 제어하고 반자동 액티비티나 자동 액티비티의 경우에도 필요한 경우 모델링시 관련된 부문(액터나 액티비티)에 커뮤니케이션 패턴을 가지도록 모델링한다.

3.3 적용 사례

하이브리드 기반의 방법은 목표를 중심으로 하는 결과 지향적인 활동을 절대적으로 요구한다. 따라서 인적 자원의 역할이 중요시되고 인적오류의 제어가 상대적으로 중요한 부분으로 부각된다. 본 논문에서 적용하는 사례는 제조업체에서 원자재 구매를 위한 자재부와 구매부 및 협력업체(공급처)간의 협업 워크플로우를 대상으로 하였다.

그림 8은 이와 같은 워크플로우의 일부를 보여준다.

그림 8에서 보는 바와 같이 사례에서의 프로세스는 다음과 같다. 우선 회사 구매부의 구매 검토를 기점으로 프로세스가 시작된다. 구매 검토는 정해진 기준(예를 들어 원료의 재고량이 안전재고 이하로 내려가면 프로세스가 시작되는 등)에 따라 시작된다. 프로세스가 시작되면 구매 검토후 다음 액티비티인 구매 요청서 작성 부분에서 원료명과 구매량 등을 기입한 요청서를 구매부에 보낸다. 구매부는 구매 요청서를 검토하여 거절하거나 수락할 수 있다. 거절하는 경우 프로세스를 종료하고 거절된 사유를 자재부에 보낸다. 수락하는 경우 견적 요청서를 작성하여 거래처인 공급사에 견적 요청서를 발송한다. 견적 요청서를 접수한 공급사에서는 견적 요청서를 검토하여 견적서를 작성한 후 거래처에 발송한다. 그리고 견적서 접수를 하면 평가해서 공급사를 1차 선정, 최종 선정 단계를 거쳐 발주한다.

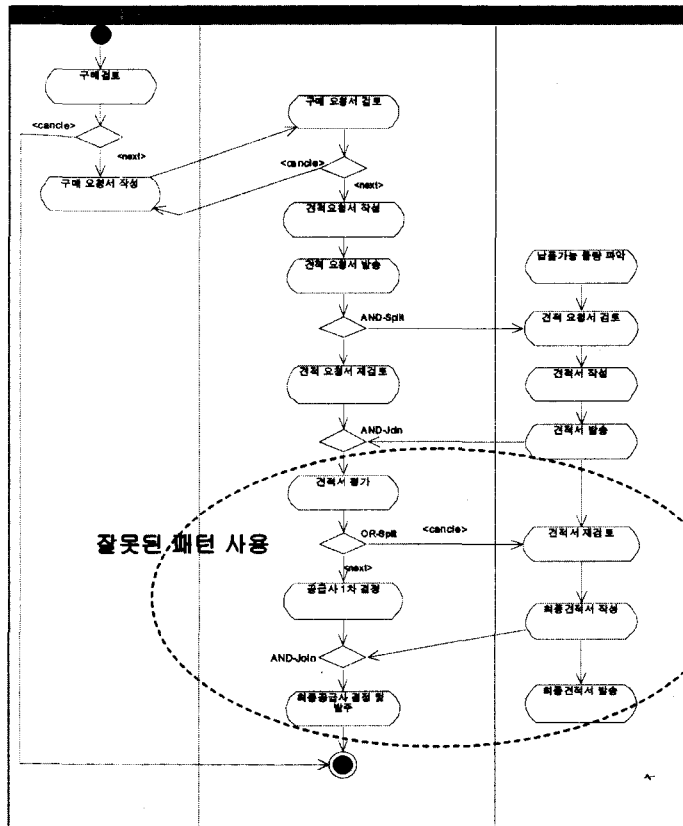


그림 8 워크플로우 사례(적용전)

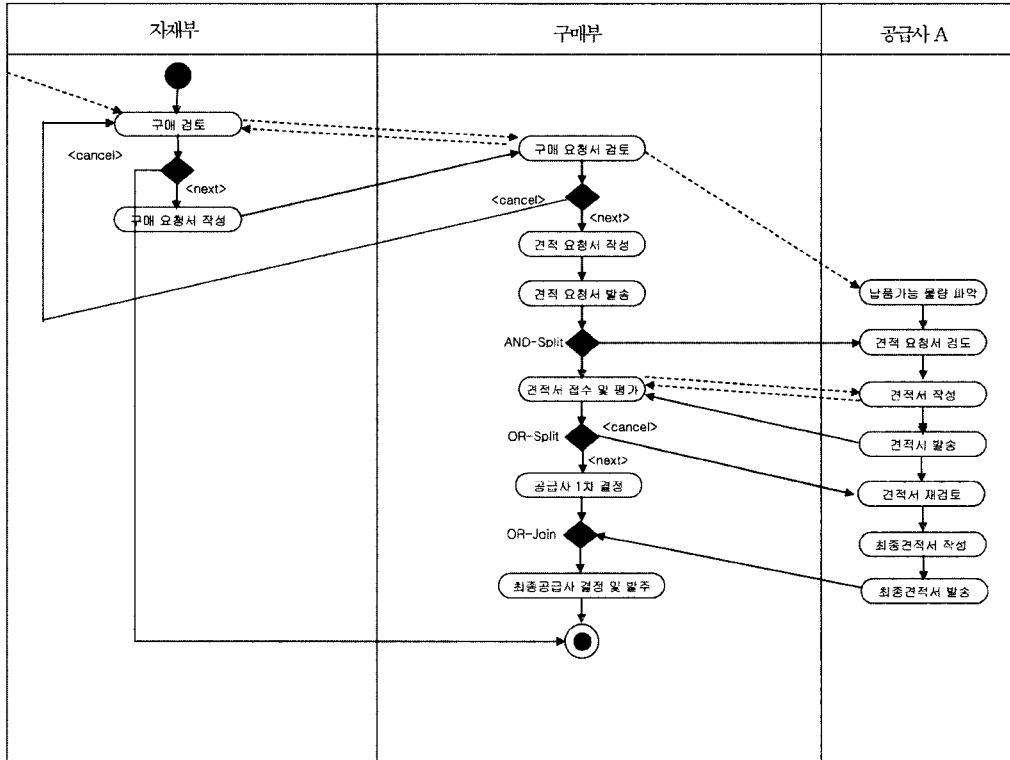


그림 9 하이브리드 기반의 모델링 방법을 적용한 사례

이와 같은 사례를 바탕으로 인적오류 제어를 위해 하이브리드 기반의 모델링 방법을 적용한 결과는 그림 9와 같다.

그림 9에서 보는 바와 같이 먼저 그림 8에서 표시된 액티비티 기반의 진행 관점에서 발생할 수 있는 규칙 기반 오류에 해당하는 잘못된 워크플로우 패턴을 수정하였다. 즉 견적서 평가에서 최종 공급사 결정 및 발주 액티비티 사이의 잘못된 분기 패턴을 재모델링하여 적용하였다. 또한 인간이 개입된 수동 작업의 경우 커뮤니케이션 패턴을 첨가해서 적용하였다. 즉 구매 검토, 구매 요청서 검토, 견적서 접수 및 평가 등의 액티비티는 이에 해당되고 각기 상황에 맞는 커뮤니케이션 패턴을 첨가해서 사용하도록 재모델링하였다. 예를 들어 구매 요청서 검토 액티비티의 경우 반드시 자재부의 구매 검토 액티비티와 커뮤니케이션이 이루어져야 하며 공급사에도 준비할 수 있도록 정보를 주어야 하므로 이에 맞게 모델링하였다. 그리고 기타 다른 프로세스도 이러한 하이브리드 기반의 모델링 방법에 맞추어 재모델링하였다. 이와 같은 사례에서 보는 바와 같이 하이브리드 기반의 모델링 방법을 사용하면 기존의 관련 모델링 도구

에서 제공하지 못하는 체계적인 작업 수행과 원활한 커뮤니케이션이 가능해진다.

4. 협업 워크플로우 모델링 도구 설계

시스템의 동적인 측면을 흐름으로 표현하고 기술하는데 잘 알려져 있는 UML 액티비티 다이어그램을 확장하여 협업 워크플로우 모델링에 적용한다. 또한 공통적인 프로세스 정의 교환을 위한 워크플로우 정의 언어인 XPDL을 사용하여 구현한다.

4.1 전체 구성

WfMC의 표준 워크플로우 규약을 준수하고 특히 WfMC의 참조 모델에 준한 아키텍처를 바탕으로 구현한다. WfMC에서는 기존의 워크플로우 참조 모델에 근거하여 워크플로우에 관한 여러 가지 표준들을 제정하고 있는데, XPDL은 참조 모델의 인터페이스 1에 해당되는 프로세스 정의 모델을 XML로 표현한 것이다. 그림 10은 이와 같은 협업 워크플로우 모델링 도구의 전체 구성도를 보여준다.

그림 10에서 보는 바와 같이 본 논문에서의 구현 대상은 워크플로우 관리 시스템 전체 구성도에서 비즈니스

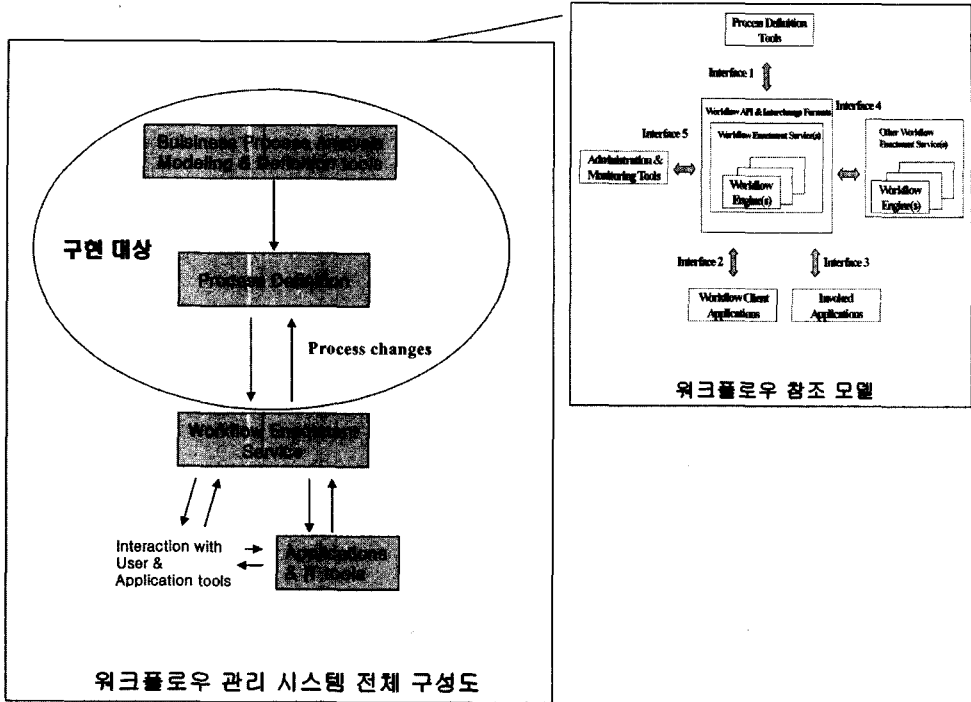


그림 10 협업 워크플로우 모델링 도구의 전체 구성도

스 프로세스를 분석하여 모델링하고 정의하는 도구 및 프로세스를 정의하여 교환이 이루어지는 부분까지를 대상으로 한다. 즉 워크플로우 참조 모델에서 인터페이스 1 부분에 해당되는 부분을 구현한다. 또한 구현하는 협업 워크플로우 모델링 도구는 JHotDraw[26]의 아키텍처를 사용하여 변경 및 확장이 용이하도록 구현한다. 그림 11은 이러한 아키텍처를 보여준다.

그림 11에서 보는 바와 같이 JHotDraw 아키텍처는 javax.swing.JFrame과 javax.swing.JPanel을 중심으로

하는 구조를 가진다.

4.2 UML과 XPDL 매핑

UML의 적용 범위는 소프트웨어 시스템, 하드웨어 시스템 및 실시간 시스템 등 매우 다양하다. 특히 효율적으로 시각화된 표기법을 공유함으로써 비즈니스 분석가 및 시스템 분석가간의 의사소통을 통하여 목표하는 결과를 도출할 수 있는 디팩토(de-facto) 표준이다. 본 논문에서는 이러한 UML을 확장하여 비즈니스 프로세스를 도식화하고 워크플로우 관점에서 이를 적용한다. 또한 프로세스 교환을 위하여 워크플로우 프로세스 정의 언어인 XPDL을 사용하여 정의한다. 아래 표 4는 본 논문에서 적용하는 액티비티 다이어그램 기반 표기 및 XPDL 스키마간의 매핑(mapping)을 나타낸다.

표 4에서 보는 바와 같이 시각적인 표현 방법을 UML을 사용하여 표기하고, XPDL을 사용하여 프로세스 정의를 구현한다. 그리고 이러한 매핑을 기반으로 커뮤니케이션 패턴에 관한 XPDL 스키마를 제시하면 그림 12와 같은데, 그림에서 보는 바와 같이 커뮤니케이션 패턴은 정적 구성, 반응하는 동적 구성 및 멀티 동적 구성으로 구분된다.

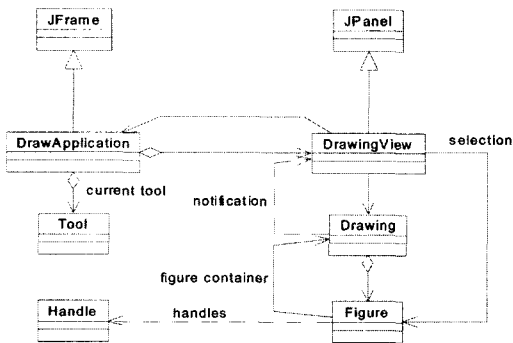



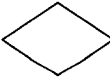


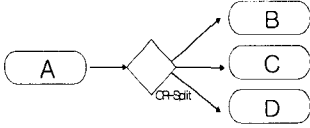
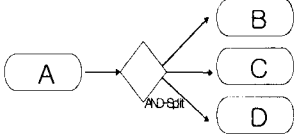
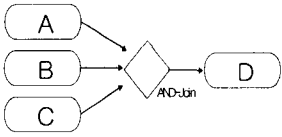
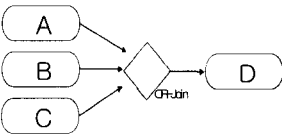
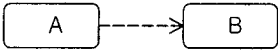
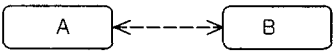
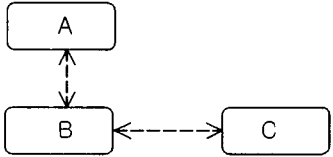


그림 11 모델링 도구의 기본 컴포넌트에 대한 아키텍처

표 4 액티비티 다이어그램 기반 표기와 XPDL 매핑

구성 요소	표기	XPDL 매핑
Initial Node		<code><Activity> <Route/> </Activity></code>
Final Node		<code><Activity> <Route/> </Activity></code>
Activity Node		<code><activity Id="..." Name="..."> <Implementation> ... </Implementation> <TransitionRestrictions> ... </TransitionRestrictions> <ExtendAttributes> ... </ExtendAttributes> </Activity></code>
Decision Node		<code><Activity> <route/> <TransitionRestriction> <Split Type="AND"/> </TransitionRestriction> <Activity> Combined with : <Transition> <Condition> <Transition></code>
Transition		<code><Transition Id="..." From="..." To="..."></code>
Swimlane		<code><Participant Id="..."> <ParticipantType Type="..."> <Description>...</Description> </Participant></code>
Split(OR-Split)		<code><TransitionRestriction> <Split Type="XOR"> <TransitionRef Id="..."/> <TransitionRef Id="..."/> </Split> </TransitionRestriction></code>
Fork(AND-Split)		<code><TransitionRestriction> <Split Type="AND"> <TransitionRef Id="..."/> <TransitionRef Id="..."/> </Split> </TransitionRestriction></code>
Join(AND-Join)		<code><TransitionRestriction> <Join Type="AND"> <TransitionRef Id="..."/> <TransitionRef Id="..."/> </Join> </TransitionRestriction></code>
Merge(OR-Join)		<code><TransitionRestriction> <Join Type="XOR"> <TransitionRef Id="..."/> <TransitionRef Id="..."/> </Join> </TransitionRestriction></code>

<p>communication (정적구성)</p>		<pre><Communication Type="STATIC"> <ParticipantRefs> <ParticipantRef> <Participant Id="..."/> </ParticipantRef> </ParticipantRefs> <Tool Id="..." Type="..."> <Description>...</Description> </Tool> </Communication></pre>
<p>communication (반응하는 동적구성)</p>		<pre><Communication Type="DYNAMIC"> <ParticipantRefs> <ParticipantRef> <Participant Id="..."/> </ParticipantRef> </ParticipantRefs> <Tool Id="..." Type="..."> <Description>...</Description> </Tool> </Communication></pre>
<p>communication (멀티 동적구성)</p>		<pre><Communication Type="MULTI"> <ParticipantRefs> <ParticipantRef Id="..."/> <ParticipantRef Id="..."/> </ParticipantRefs> <Tool Id="..." Type="..."> <Description>...</Description> </Tool> </Communication></pre>

```
<xsd:element name="Communication">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:ParticipantRefs"/>
      <xsd:element ref="xpd:Tool"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Type" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="STATIC"/>
          <xsd:enumeration value="DYNAMIC"/>
          <xsd:enumeration value="MULTI"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="ParticipantRefs">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:Participant" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

그림 12 커뮤니케이션 패턴에 대한 XPDL 스키마

5. 협업 워크플로우 모델링 도구 구현 및 평가

5.1 구현

본 논문에서 제시하는 모델링 도구는 Java로 구현되었다. 개발도구는 JDK 1.4.2 및 JBuilder 9 Enterprise

를 사용하였고 Window 2000 Server를 운영체제로 사용하였다. 또한 XML 1.0, XML 스키마 1.1 및 JDOM 등을 사용하였고 프로세스 정의 언어로 XPDL 1.0을 사용하여 구현하였다. 본 도구에서 사용하는 전체적인 스키마는 그림 13과 같다.

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema targetNamespace="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xpdl="http://www.wfmc.org/2002/XPDL1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:element name="Activities">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:elementref="xpdl:Activity" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Activity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:elementref="xpdl:Description" minOccurs="0" />
        <xsd:elementref="xpdl:Limit" minOccurs="0" />
        <xsd:choice>
          <xsd:elementref="xpdl:Route" />
          <xsd:elementref="xpdl:Implementation" />
          <xsd:elementref="xpdl:BlockActivity" />
        </xsd:choice>
        <xsd:elementref="xpdl:Performer" minOccurs="0" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="WorkflowProcesses">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:elementref="xpdl:WorkflowProcess" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="WorkingTime" type="xsd:string" />
  <xsd:element name="XPDLVersion" type="xsd:string" />
  <xsd:element name="Xpression">
    <xsd:complexType mixed="true">
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:anyprocessContents="lax" minOccurs="0" maxOccurs="unbounded" />
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

그림 13 전체적인 XPDL 스키마

위와 같은 스키마를 바탕으로 사례를 적용하여 모델링 도구를 실행한 결과 화면은 다음 그림 14와 같다. 그림 14에서 보는 바와 같이 분기 패턴을 사용한 경우와 커뮤니케이션 패턴 중 동적 구성 패턴을 사용한 경우의 실행 결과를 XPDL 인스턴스로 보여준다. 사례는 3장에서 제시한 사례를 사용하였으며 커뮤니케이션 패턴의 경우 구매 요청서를 검토하는 액티비티의 경우를, 분기 패턴의 경우 견적서 접수 액티비티와 공급사 1차 결정 액티비티 사이의 분기에 적용한 결과를 제시한다.

그리고 이러한 패턴 적용 시 올바르게 유효한 패턴만을 사용할 수 있도록 구현하였는데 예를 들어 규칙 기반 오류를 제어 할 수 있도록 하는 분기 패턴에서의 잘못된 패턴을 제어하는 알고리즘은 그림 15와 같다.

그림 15에서 보는 바와 같이 이 알고리즘은 잘못된 분기 패턴을 제어하는 알고리즘으로 입력값으로 프로세스를 구성하는 액티비티들의 집합을 가지고, 출력값으로 잘못된 패턴을 가지는 경우에 true, 유효한 패턴을 가지는 경우 false를 출력하도록 구성되어 있다.

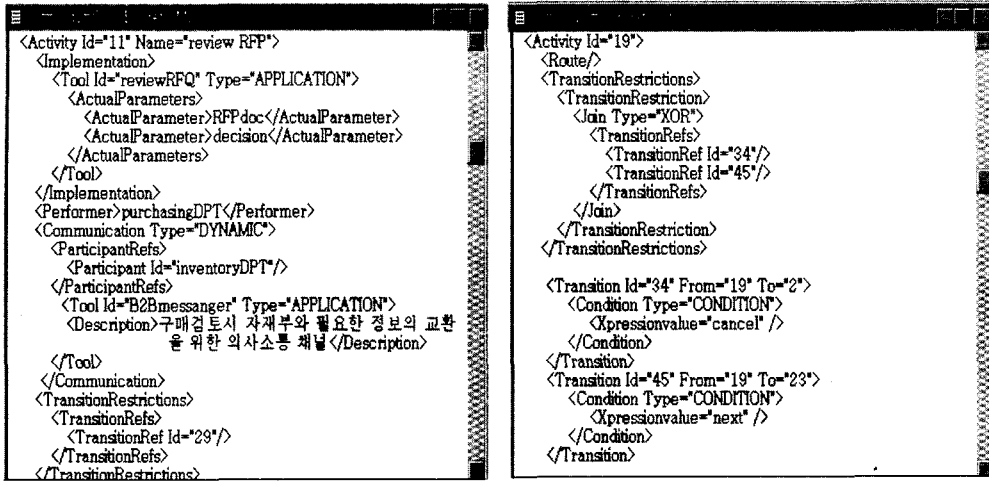


그림 14 커뮤니케이션 패턴(동적 구성) 및 분기 패턴을 사용한 경우의 XPDL 인스턴스

```

Algorithm invalidPatternCheck(activities)
입력: 프로세스를 구성하는 액티비티들의 집합
출력: 잘못된 패턴을 가지는 경우 true, 유효한 패턴을 가지는 경우 false

// 알고리즘의 시작을 위해 프로세스의 첫번째 액티비티를 가져온다
activity ← activities.getFirstActivity()
while break_condition do
// 액티비티의 타입이 ROUTE이고 라우팅 타입이 XOR_SPLIT인 경우
if activity.type=ROUTE && activity.routeType=XOR_SPLIT then
  // 서브 액티비티를 참조하기 위해 액티비티의 전이가 가리키는 액티비티의 아이덴티티들을 activity_ids에 저장
  activity_ids ← activity.getTransitionRefIds()
  // 배열 액티비티 아이덴티티의 크기만큼 반복
  for i ← 0 to activity_ids.length do
    // 액티비티가 가지는 서브 액티비티들을 액티비티 아이덴티티를 통해 가져온다
    subActivity ← activity.getSubActivity(activity_ids[i])
    // 서브 액티비티의 타입이 ROUTE이고 라우팅 타입이 AND_SPLIT인 경우
    if subActivity.type=ROUTE && subActivity.routeType=AND_SPLIT then
      // 서브 액티비티의 아이덴티티들을 route_id 배열에 저장
      route_id[index++] ← subActivity.id
      // route_id 배열의 크기만큼 반복
    for i ← 0 to route_id.length do
      // route_id 배열의 항목으로부터 라우트 액티비티를 가져온다
      route ← (Route).getActivityFromId(route_id[i])
      // 라우트 액티비티의 전이참조를 통해 서브 노드값들을 2차원 배열 subNodes에 저장
      subNodes[i] ← route.getTransitionRefs()
      // subNodes의 배열 크기만큼 반복
    for i ← 0 to subNodes.length do
      // 각 서브 노드값들을 전체 서브 노드값과 비교하여 같은 액티비티 노드를 가지는 경우 잘못된 패턴으로 판정
      if subNodes[i].hasSameNodeWith(subNodes) then
        return true // 잘못된 패턴
      else do
        return false // 유효한 패턴
  else
    // 그렇지 않은 경우 다음 액티비티를 가져온다
    activitite.getNextActivity()
  continue
    
```

그림 15 잘못된 분기 패턴 제어 알고리즘

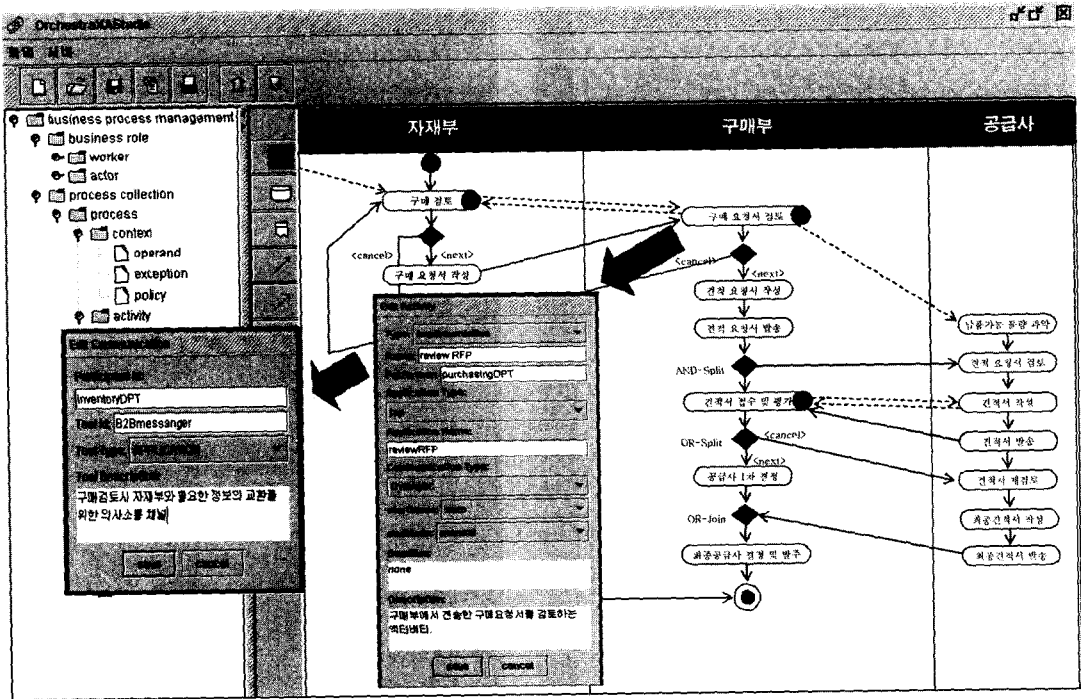


그림 16 협업 워크플로우 모델링 도구

이와 같은 내용을 바탕으로 구현된 협업 워크플로우 모델링 도구는 그림 16과 같다. 그림에서 보는 바와 같이 액티비티 기반의 작업 진행과 커뮤니케이션 기반의 정보 교류가 이루어질 수 있는 하이브리드 기반의 모델링 방법을 적용하여 구현하였다. 또한 인간이 개입된 수동 액티비티의 경우 색으로 구분하였고 범할 수 있는 오류 유형을 표시하였다. 그리고 제시한 두개의 창은 구매부의 구매 요청서 검토 액티비티를 설정하는 부분으로 여기서는 커뮤니케이션 타입, 연결 부분 및 사용 방법 등을 정할 수 있다.

5.2 평가

본 논문에서는 인적오류를 제어하기 위한 구조로 인지 과학 분야에서 가장 대표적인 Rasmussen의 SRK 모델을 기반으로 하는 Reason의 GEMS 모델을 사용하였다. 그리고 이러한 오류 유형을 제어하기 위한 방법으로 워크플로우 자체의 특성 및 패턴을 활용하는 방법을 제시하였다. 더욱이 커뮤니케이션 방법 및 액티비티 방법을 혼합한 하이브리드 방법을 적용하여 효율적으로 인적오류를 제어할 수 있도록 제시하였다. 이와 같은 워크플로우 시스템에서의 오류 제어에 관한 연구에는 대표적으로 ADOME-WFMS[24], WIDE[25] 및 Jablonski[23]의 방법 등이 있다. 우선 ADOME-WFMS는 위

크플로우에서의 오류를 정의하고 분류한 대표적인 방법으로 워크플로우 자체의 특성과 범할 수 있는 오류를 체계적으로 제시하였다. 그러나 이와 같은 방법은 인지적인 모델을 간과하는 시스템에 편향된 접근이고 더욱이 실행시에 발생하는 예외처리에 한정되어 있다. 그리고 WIDS도 마찬가지로 예외처리 관점에서 오류 제어 방법을 제시하고 있다. 마지막 Jablonski의 방법은 변화 관점에서 범할 수 있는 오류를 패턴과 기준을 사용하면 제시하는 아키텍처적인 접근인데 마찬가지로 인지 부분에 대한 뒷받침이 부족하다. 본 논문에서 제안한 방법과 기존의 ADOME-WFMS, WIDE 및 Jablonski의 방법을 비교하면 표 5와 같다.

표 5에서 보는 바와 같이 본 논문의 초점을 중심으로 비교 항목을 인지 관련 부분 및 워크플로우 관련 부분으로 분류하였다. 그리고 인지 관련 부분을 다시 인간 의사결정시의 특성에 기반을 둔 인지적 모델 기반의 오류 정의 및 분류 여부와 오류 제어에 대한 대안 제시 여부 측면으로 구분하였다. 그리고 워크플로우 관련 부분을 목표를 지향하며 작업을 진행시키는 등의 워크플로우 특성을 지원하는 워크플로우 모델링 방법 제공 여부와 프로세스 정의 언어 지원을 통한 모델링 도구 구현 및 구현된 워크플로우 엔진 적용 여부 측면으로 구분하였다.

표 5 다른 방법들과의 비교

GEMS		Labronski		Labronski	
인지 관련	인지적 모델 기반의 오류 정의 및 분류	GEMS 모델 기반의 오류 정의 및 분류	오류 정의 및 분류 방법 없음	예외처리 관점의 실패 정의 및 예외 구분	결합 설계 관점의 실행 오류 정의 및 분류
	오류 제어 방법	모델링 관점의 오류 제어 방법	모델링 관점 및 실행 관점의 예외처리 방법	예외처리 관점의 오류 제어 방법	변화 관점의 오류 제어 방법
워크플로우 관련	워크플로우 특성을 지원하는 모델링 방법	워크플로우 특성을 지원하는 하이브리드 모델링 방법 제시	워크플로우 특성을 지원하는 실행 관점의 모델링 방법 제시	워크플로우 특성을 지원하나 모델링 방법 제시 못함	워크플로우 특성을 지원하는 변화 관점의 모델링 방법 제시
	모델링 도구 구현 및 워크플로우 엔진 적용	XPDL을 지원하는 모델링 도구 구현(엔진에는 적용하지 못함)	프로세스 정의 언어 지원 모델링 도구 미비(엔진에도 적용하지 못함)	프로세스 정의 언어 지원 모델링 도구 미비(엔진에는 설계 측면에 부분적으로 적용)	프로세스 정의 언어 지원 모델링 도구 미비(엔진에도 적용하지 못함)

먼저 기존의 다른 방법들은 주로 실행 측면의 오류를 대상으로 정의하고 분류하였는데 반해 본 논문에서는 인지 모델인 GEMS 모델을 기반으로 오류를 정의하고 분류하였다. 또한 오류 제어 방법에서 본 논문에서는 모델링 관점에서 오류 제어를 제시한 반면, ADOME-WFMS의 방법은 모델링 및 실행 관점에서, 나머지 두 가지 방법은 실행 관점에서의 오류 제어 방법을 제시하고 있다. 그리고 워크플로우 관련된 측면에서 보면 기존의 방법들 대부분이 시간 및 목표를 중심으로 체계적인 작업을 수행하도록 해주는 워크플로우 특성을 지원하지만 프로세스 정의 언어를 지원하거나 구현된 도구 제시에서는 부족하다. 예를 들어 주로 시스템적인 처리 측면에서 엔진실행시의 예외 상황 변화 패턴 등에 대한 처리 방법을 제공하고 있기 때문에 실제 구현 측면에서 중요한 표준 비즈니스 프로세스 정의 언어를 지원하거나 구현된 모델링 도구 제시에서는 부족하다. 반면에 본 도구는 예외처리와 같은 실행 차원의 접근 및 구현된 워크플로우 엔진에의 적용 등에서는 부족하다.

이상의 비교를 종합해 보면 기존의 방법들은 주로 오류를 실행 측면의 오류로 규정하여 적용하면서 이에 해당하는 예외처리나 결합 설계 측면의 오류를 주 대상으로 하고 있기 때문에 인간의사 결정을 바탕으로 하는 인지적 모델을 사용하는 체계적인 접근이 부족하다. 또한 구현 측면에서 보면 프로세스 정의 언어를 지원하거나 구현된 도구 제시도 부족하다. 그러나 본 워크플로우 모델링 도구는 대표적인 인지적 모델을 기반으로 접근하고 있고 표준화된 워크플로우 모델링 언어를 지원하는 등의 장점이 있다. 반면에 실행 관점의 구체적인 워크플로우 엔진에의 적용은 미비하다.

6. 결론 및 향후 연구

기업간 협업 워크플로우 시스템은 기업 내의 환경을

얼마나 잘 반영되느냐가 성공 여부를 결정한다. 즉 자신의 비즈니스 환경과 비즈니스 프로세스를 충분히 이해하고 이에 적합한 워크플로우 시스템 환경을 구축한다면 효율적인 협업을 실현할 수 있을 것이다.

본 논문은 인적오류를 제어하는 방법으로 하이브리드 기반의 모델링 방법을 제시하였다. 그리고 목표지향적인 워크플로우 시스템에서의 협업 수행시 행위 자체가 목표에 부합되도록 하고, 문제 해결이나 의사결정시 사려 깊은 행위를 하도록 유도하여 궁극적으로 인적오류를 제어할 수 있도록 GEMS 모델을 적용하였다. 아울러 워크플로우 시스템에 인적오류 제어를 위해 제안하는 GEMS 모델 및 하이브리드 기반의 모델링 방법을 사용하여 협업 워크플로우를 모델링할 수 있는 도구를 설계하고 구현하였다.

본 논문에서 제시하는 있는 협업 워크플로우 시스템에 인적오류 절감을 위해 제안하는 GEMS 전략을 하이브리드 기반의 방법에 적용하여 기업은 자신의 업무 환경에 적합한 워크플로우 시스템을 위한 비즈니스 프로세스 모델링을 구현할 수 있을 것이다. 이를 통하여 기업이 효율적인 e-비즈니스 환경에 참여하고 IT 경쟁력을 향상하는데 도움이 될 것이다. 향후 연구로는 워크플로우 모델링시에 적용하는 인적오류 제어를 발전시켜 워크플로우 실행시 즉 전체적인 워크플로우 시스템에서 워크플로우 엔진에 적용하는 인적오류 제어 방안에 대한 연구가 필요하다.

참고 문헌

[1] D. Jutla, et al., "Making Business Sense of Electronic Commerce," IEEE Computer, Vol. 32, pp. 67-75, 1999.
 [2] Shari Lawrence Pfleeger, Software Engineering Theory & Practice, Prentice-Hall, 1998.
 [3] Chris Marshall, Enterprise Modeling with UML-

Designing Successful Software Through Business Analysis, Addison-Wesley, 1999.

[4] Layna Fischer, 2003 Workflow Handbook, Workflow Management Coalition, 2003.

[5] 신상철, "기업간 워크플로우 통합 기술 표준 연구보고서," 한국 전산원, 2002.

[6] Akhil Kumar and Leon Zhao, "Workflow Support for Electronic Commerce Applications," Decision Support Systems, Vol. 32, pp. 265-278, 2002.

[7] Lerina Aversano, Gerardo Canfora, Andrea De Lucia, and Pierpaolo Gallucci, "Business Process Reengineering & Worklow Automation: A Technology Transfer Experience," The Journal of Systems and Software, Vol. 63, pp. 29-44, 2002.

[8] C. Bussler, "B2B Protocol Standards and Their Role in Sementic B2B Integration Engines," IEEE Computer Society. Vol. 24, pp. 35-43, 2002.

[9] Ruth Sara Aguilar-Savlen, "Business Process Modelling: Review and Framework," International Journal of Production Economics, Vol. 50, pp. 235-256, 2003.

[10] Gregory Mentzas, Christos Halaris, and Stylianos Kavadias, "Modelling Business Processes with Workflow Systems: An Evaluation of Alternative Approaches," International Journal of Information Management, Vol. 21, pp. 123-135, 2001.

[11] WPMC-TC-1025, Workflow Process Definition Interface-XML Process Definition Language Version 1.0, Workflow Management Coalition, 2002.

[12] Diimitrios Georgakopoulos and Mark Hornickd, "An Overview of Workflow Management: from Process Modeling to Workflow Automation Infrastructure," Distributed and Parallel Databases, Vol. 3, pp. 119-153, 1995.

[13] S. Lawrence and C. L. Giles, "Accessibility of Information on the Web," Nature, Vol. 400, pp. 107-109, 1999.

[14] S. Aissi, P. Malu, and K. Srinivasan, "E-business Process Modeling: The Next Big Step," Computer, Vol. 35, pp. 55-62, 2002.

[15] ebXML Spec., <http://ebXML.org/specs/bpOVER.pdf>

[16] K. T. Phalp, P. Henderson, R. J. Walter and G. Abeyasinghe, "RolEnact: Role Based Enactable Models of Business Processes," Information & Software Technology, Vol. 40, pp. 123-133, 1998.

[17] Kwang-Hoon Kim and Dong-Soo Han, "Performance and Scalability Analysis on Client-Server Workflow Architecture," Proceedings of the 8th ICPADS, pp. 176-186, 2001.

[18] 김형도, 김철한, "웹서비스로서 ebXML 구현방안 연구보고서", 전자거래진흥원, 2002.

[19] R. Parasuraman and V. Riley, "Humans and Automation: Use, Misuse, Disuse, and Abuse," Human Factors, Vol. 39, No. 2, pp. 230-253, 1997.

[20] Rasmussen, J., "Skills, Rules, and Knowledge: Signals, Signs, Symbols, and other Distinctions in Human Performance Models," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 13, pp.

257-267, 1983.

[21] Rasmussen, J., Pejtersen, A. M., and Goodstein, L. P., Cognitive Systems engineering, Wiley, New York, 1994.

[22] Reason, J., Human Error, Cambridge Univ. Press, 1990.

[23] W. M. P. van der Aalst and S. Jablonski, "Dealing with Workflow Change: Identification of Issues and Solutions," International Journal of Computer Systems Science & Engineering, Vol. 5, pp. 267-276, 2000.

[24] K. W. Dickson, Qing Li, and Kamalakar Karlapalem, "Exception Handling with Workflow Evolution in ADOME-WFMS: A Taxnomy and Resolution Techniques," ACM SIG GROUP, Vol. 20, 1999.

[25] Claus Hagen and Gustavo Alonso, "Exception Handling in Workflow Management Systems," IEEE Transactions on Software Engineering, Vol. 26(10), pp. 943-958, 2000.

[26] JHotDraw: Open-Source Project, <http://www.jhotdraw.org/>



이 상 영

1994년 숭실대학교 산업공학과 졸업(공학사). 1996년 전북대학교 대학원 산업공학과 졸업(공학석사). 1999년~현재 전북대학교 대학원 전산통계학과 박사과정 수료. 관심분야는 워크플로우, 전자상거래, HCI, SCM, 멀티미디어 시스템, 모바일 애플리케이션 등



유 철 중

1982년 전북대학교 전산통계학과 졸업(이학사). 1985년 전남대학교 대학원 계산통계학과 졸업(이학석사). 1994년 전북대학교 대학원 전산통계학과 졸업(이학박사). 1982년~1985년 전북대학교 전자계산소 조교. 1985년~1996년 전주기전여자대학 전자계산과 전임강사~부교수. 1997년~현재 전북대학교 자연과학대학 컴퓨터과학과 전임강사~부교수. 관심 분야는 소프트웨어 개발 프로세스, 소프트웨어 품질, 컴포넌트 소프트웨어, 소프트웨어 매트릭스, 소프트웨어 에이전트, GNSS, GIS, 교육공학, 인지과학 등



장 옥 배

1966년 고려대학교 수학과 졸업(이학사) 1973년 고려대학교 교육대학원(교육학석사). 1980년 조지아 주립대(박사과정 수료). 1987년 산타바바라대학교 졸업(Ph.D). 1990년~1991년 영국에딘버러대학교 객원교수. 1980년~현재 전북대학교 공과대학 전자·정보공학부 교수. 관심분야는 소프트웨어공학, 전산교육, 수치해석, 인공지능 등