

다중 에이전트 기반 워크플로우 모델링 시스템에 관한 연구

A Study on Multi Agent-Based Workflow Modeling System

김 학 성* 김 광 훈** 백 수 기***
Hak-Seong Kim Kwang-Hoon Kim Su-Ki Paik

요 약

본 논문에서는 워크플로우 관리 시스템의 빌드타임 기능에 속하는 모델링 시스템을 다중 에이전트 기반으로 구현 하였다. 본 논문에서 제안된 다중 에이전트 기반 워크플로우 모델링 시스템은 클라이언트의 경량을 위하여 자바를 이용하였으며, 프로세스 정의에 필요한 세션, 조직, 관련데이터, 응용프로그램 속성들을 별도의 에이전트로 구성하였다. 또한, 프로세스 정의를 위한 모델은 ICN(Information Control Net)을 이용하여 작성된 모델의 문법(syntax)을 검증하였으며, 타 워크플로우 엔진과의 연동을 위하여 WfMC에서 정의된 인터페이스 형태인 WPDL로 import/export할 수 있는 기능을 제공한다.

Abstract

Workflow Management Systems(WFMSs) is a software system that supports the specification and execution of business processes. In this paper, we proposed Multi Agent Based Workflow Modeling System which was implemented by Java application. The proposed workflow modeling system is divided into four agents; Session, Organization, Relevant Data, invoked Application. We adapted ICN(Information Control Net) to check workflow model syntax. And the proposed modeling system provide the function to import/export WPDL which was defined in WfMC.

1. 서 론

조직의 업무 프로세스는 그 프로세스의 목표를 달성하기 위해 필요한 여러 단위 업무로 구성되고, 이와 같은 단위업무는 정의된 비즈니스 규칙에 따라 진행된다. 이와 같은 조직내의 업무 프로세스는 다음과 같은 두 가지 관점에서 크게 변하고 있다. 첫 번째 변화는 조직의 내부적인 것으로 고객의 다양한 요구사항을 수용하기 위하여 차별화된 업무 프로세스를 제공하여야 한다는 점이다. 이러한 이유로 인하여 조직 또는 기업들은 CRM(Customer Relation Management)등을 구축하기 위

하여 워크플로우를 도입하고 있다[1]. 두 번째 변화는 기업 외부와의 관계에서 발생하는 변화로서, 기업간 제휴나 협력 업체와의 긴밀한 협조 체제 구축 등 지리적, 조직적으로 기업간의 긴밀한 공동작업이 기업 경쟁력 제고의 발판으로 부각되고 있다는 것이다. 이러한 두 가지 변화는 조직에서 처리해야 할 업무 프로세스를 복잡하고 규모가 커지게 만들고 있고, 업무 프로세스 처리의 정확도와 신속도가 경쟁력을 결정하는 핵심 요인으로 인식되고 있다. 최근 B2C/B2B로 대변되는 전자상거래 및 전자시장의 활성화가 급속하게 확장됨에 따라 워크플로우를 기반으로 하는 기업 내부 업무흐름의 자동화를 의미하는 B2E의 구축을 가속화시키고 있으며, 워크플로우 기술 및 시스템은 결국 기업 또는 조직체 내의 모든 업무 처리 절차들을 통합 관리하는 인프라구조로 인식되고 있다. 워크플로우 관리 시스템은 일반적으로 빌드타

* 정 회 원 : 동남보건대학 컴퓨터응용과 교수
amang@dongnam.ac.kr

** 중신회원 : 경기대학교 정보과학부 교수
kwang@kyonggi.ac.kr

*** 비 회 원 : 경기대학교 정보과학부 교수
skpaik@kuic.kyonggi.ac.kr

임과 런타임으로 구분할 수 있으며, 전자는 메타 모델에 따라 사용자가 업무 프로세스 구조를 정의할 수 있는 프로세스 정의 도구 부분을 의미하고, 후자는 프로세스의 생성, 프로세스의 탐색 또는 프로세스의 제어와 같이 빌드타임에서 정의된 프로세스를 수행하는 기능을 담당한다[2,3].

빌드타임의 기능을 수행하는 기존의 워크플로우 모델링 시스템은 플로우 차트 스테이트 차트 또는 페트리-넷을 이용하여 워크플로우 모델을 작성하였으나 작성된 모델의 검증을 위해서는 정형적인 형태로 변환하는 과정이 필요하다[4].

따라서 본 논문에서는 워크플로우 기술의 한 부분인 업무 프로세스를 정의하는 모델링 부분에 있어 다중 에이전트를 이용하여 모델링이 가능하도록 지원하는 워크플로우 모델링 시스템을 구현하였고 프로세스 정의를 위한 모델로서 ICN을 사용하여 ICN에서 제공되는 정형기법을 이용하여 작성된 모델의 변환과정 없이 작성된 모델을 검증할 수 있도록 구현하였다. 또한 작성된 모델의 호환성을 위하여 WPDL을 제공하여, 작성된 모델이 WfMC의 표준을 따르는 모든 워크플로우 엔진에서 사용할 수 있도록 구현하였다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서는 본 논문의 관련연구로서 에이전트의 개념에 관한 내용을 기술 하였고, 3장에서는 워크플로우 모델링과 프로세스 정의를 위한 모델인 ICN에 대하여 살펴보고, 4장에서는 다중 에이전트 기반 모델링 시스템의 설계 및 구현에 관하여 기술하고 마지막으로 5장에서 결론 및 향후 연구과제에 대하여 기술하겠다.

2. 에이전트

에이전트란 사전적으로 대리인, 대행자란 뜻을 가지며, 컴퓨터 분야에서는 사용자를 대신하여 사용자가 원하는 어떤 일을 수행해주는 프로그램이라고 할 수 있다. 아직까지 에이전트에 대한 정확한 정의는 없다. 에이전트를 연구하는 그룹마다 다르게 정의하기 때문에 에이전트를 명확히 정의하기

는 어려운 일이다. 따라서 각 연구 그룹들이 정의한 에이전트의 정의를 설명하면 다음과 같다[5,6,7].

2.1 Pattie Maes Agent (MIT univ.)

자율적인 에이전트(autonomous agent)는 어떤 복잡한 동적 환경에 살며 이 환경을 자율적으로 감지하면서 동작하는 컴퓨터 시스템이며 그렇게 함으로써 그들이 고안되게 됐던 목적이나 일들을 수행토록 한다.

2.2 IBM Agent

Intelligent agent는 사용자를 대신하여 연산들을 수행하는 소프트웨어 개체들이거나 또는 약간의 독립성을 내지는 자율성을 가지고 있는 프로그램이다. 이를 위해서 지식이나 사용자의 목적 및 요구를 담을 수 있는 표현들이 채택된다.

2.3 The MuBot Agent

에이전트라는 단어는 두개의 다른 개념을 표현하는데 사용되어진다. 첫째는 autonomous 한 실행을 할 수는 에이전트의 능력이 되며 두 번째는 논리에 기반 수행 능력을 가진다는 것이다.

2.4 The Maes Agent

Autonomous한 에이전트란 복잡하며, 동적인 환경에 적응하며 그 환경에서 자율적으로 인지하면서, 행동하여 처음 설계되었던 목적들을 인지하고, 수행하는 계산적 시스템이다.

위의 정의들에서 나타났듯이 autonomous 내지는 intelligence는 에이전트의 특징 중 중요한 특징 중 하나가 될 수 있다. 이 중에서 autonomous는 에이전트와 일반 프로그램을 구별짓는 제일 중요한 특징이다. 에이전트는 autonomous을 가짐으로써 사용자의 지시, 혹은 추가적인 지시 없이 스스로 목적을

달성하기 위해 일련의 작업을 수행할 수 있게 된다.

3. 워크플로우 모델링

모델이란 실세계를 추상화해 놓은 것을 말한다. 추상화를 하는 과정은 중요한 요소들을 뽑아 내어 중요하지 않은 부분들을 제거하여 간단히 하는 과정을 말한다. 따라서 전산분야에서는 복잡한 시스템을 수학적으로 분석하고 시스템과 사용자들간의 의사전달 및 시스템 자체의 시뮬레이션 연구를 위해서 이해하고 개발하는 데 있어서 추상화가 유용하게 이용되고 있다.

워크플로우 모델은 조직의 모습을 할당 업무, 업무자, 단위업무와 자료저장소 등의 기호로 표현을 한다. 따라서 이와 같은 워크플로우 모델은 컴퓨터로 추상화되어 모델링 되어지면, 프로세스의 생성, 변화 및 시뮬레이션 작업이 가능하게 된다. 이러한 이유로 모델링을 가능하게 하는 도구는 반드시 워크플로우 전문가가 아니라도 이해할 수 있고, 작업이 쉽도록 개발되어야 한다. 또한 모델링 도구는 수학적 의미를 내포하기 때문에 분석이 가능해야 한다. 이러한 모델링 도구가 만들어졌을 때 그 도구는 유용하게 사용될 수 있을 뿐만 아니라 워크플로우 수행 중에 발생할 수 있는 예외 상황과 다양한 변화에 대처 가능하며, 일반적인 정보의 공유도 가능하게 된다[8,9,10].

3.1 워크플로우 모델의 표현

조직은 일련의 기본적인 원리와 프로시저 안에서 동작한다. 몇몇 조직들에서는 그 조직체 내에서 어떻게 조직들이 동작하는 지를 기술하는 확대된 성격을 지닌 프로시저 설명서가 있기도 하다. 그러나 또 다른 조직들에서는 이러한 프로시저를 나타내는 설명서를 다음과 같은 이유로 쓰지 않는다. 첫 번째, 이러한 설명서가 조직의 모든 구성원들이 이해할 수 있도록 프로시저를 표현하는데 쉽지 않다는 것이다. 두 번째, 프로시저

는 비즈니스 활동이 점차적으로 복잡 다양해질수록 프로시저 역시 매우 복잡하게 변하게 된다. 이것은 BPR(Business Process Reengineering)을 수행할 때 심각한 문제를 유발한다. 세 번째는 이렇게 기술된 내용들이 조직의 구성원들로 하여금 따라야만 하고 효과를 내야만 하도록 규정하는 일종의 강요처럼 작용할 수도 있기 때문이다[11].

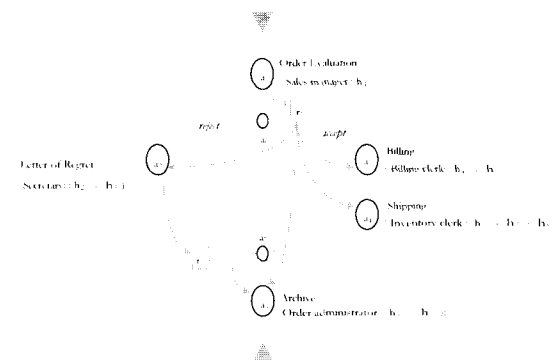
3.1.1 기본(Basic) ICN

ICN은 사무실의 개념을 일련의 관련된 프로시저의 집합으로 정의하며 이러한 프로시저는 선후관계가 존재하는 단위 업무들의 집합으로 표현된다. ICN은 그림형태로 프로시저, 단위업무, 저장소, 선후관계를 나타내는 제어흐름과 데이터 흐름을 표현한다[9,10].

3.1.2 기본 ICN 그래프 형태의 구성 및 특성

ICN제어흐름 그래프는 큰 원으로 표현되는 일련의 단위업무와 작고 빈 원으로 표현되는 OR노드, 작고 채워진 원으로 표현되는 AND노드, 그리고 이러한 노드들을 연결하는 선으로 구성된다. 화살표는 실선과 점선으로 표현되는데 이들은 노드들 간의 선후관계 및 자료저장소와의 입/출력을 표현한다[7,8].

다음 그림 1은 주문 처리과정 업무 프로세스를 ICN을 이용하여 표현한 것이다.



(그림 1) 주문처리 관계를 나타내는 워크플로우[12]

여기서 전체 프로시저의 이름은 주문처리 프로세스이며 각 과정을 담당하는 단위업무는 Order Evaluation(a₁), Letter of Regret(a₂), Billing(a₃), Shipping(a₄), Archive(a₅)이다. 그리고 a₆, a₇은 OR 분기를 나타내는 노드이며 r₁, r₂는 저장소를 나타낸다. 또한 직선형태로 된 실선은 제어의 흐름을 나타내고 곡선형태의 실선은 정보의 흐름을 나타낸다[9].

ICN은 정사각형 모양으로 표현되는 저장소 노드를 통해서 저장 속성을 지니고 유지되는 데이터를 표현하고, 작업종류별로 색채 패트라의 개념을 도입하여 그래프 상에서 토론의 흐름을 알 수 있도록 지원한다. 또한 ICN은 OR 노드를 통하여 기존의 모델에서 제공할 수 없었던 선택분기를 지원함으로써 노드 프로시저를 통해서 알맞은 노드로 분기할 수 있도록 하여 단위업무의 비결정성을 해결할 수 있도록 한다[13,14].

3.2 ICN의 명세

예를 들어 A를 일련의 단위 업무들의 집합이라고 하고, R을 일련의 자료 저장소의 집합이라 할 때 기본 ICN 명세를 수식적인 표현은 다음 표 1과 같다[8].

- 1) I는 초기에 입력되는 자료 저장소들의 유한 집합이며 ICN의 실행 전에 어떠한 외부의 프로세스에 의해서 로드되어야 한다고 가정한다.
- 2) O는 마지막으로 출력되는 자료 저장소들의 유한 집합이며 ICN의 실행 후에 어떠한 외부의 프로세스에 의해서 이용되는 정보를 포함하고 있다고 가정한다.
- 3) $\delta = \delta_i \cup \delta_o$ 이때 $\delta_o : A \rightarrow P(A)$ 은 하나의 단위업무에 후행하는 단위 업무들의 집합에 연결하는 관계를 나타내며 $\delta_i : A \rightarrow P(A)$ 은 하나의 단위업무를 선행하는 단위 업무들의 집합에 연결하는 관계를 나타내는 관계이다.
- 4) $\gamma = \gamma_i \cup \gamma_o$ 이때 $\gamma_o : A \rightarrow P(R)$ 은 하나의

(표 1) ICN명세를 위한 수식적인 표현

$\Gamma = (\delta , \gamma , I , O)$ δ : precedence constraint among activities γ : repository input / output requirement of activity I : initial input repository O : final output repository
--

(표 2) 주문처리 과정의 ICN 정형적 표현

$A = \{ a_1, a_2, a_3, a_4, a_5, a_6, a_7 \}$ $R = \{ r_1, r_2 \}$ $I = \{ \}$ $O = \{ r_1, r_2 \}$	
$\delta_i(a_1) = ((\lambda)), \delta_o(a_1) = ((a_6))$	$\gamma_i(a_1) = (\lambda), \gamma_o(a_1) = (r_1)$
$\delta_i(a_2) = (a_6), \delta_o(a_2) = ((a_7))$	$\gamma_i(a_2) = (r_1), \gamma_o(a_2) = (r_2)$
$\delta_i(a_3) = (a_6), \delta_o(a_3) = ((a_4))$	$\gamma_i(a_3) = (r_1), \gamma_o(a_3) = (\lambda)$
$\delta_i(a_4) = (a_3), \delta_o(a_4) = ((a_7))$	$\gamma_i(a_4) = (r_1), \gamma_o(a_4) = (\lambda)$
$\delta_i(a_5) = (a_7), \delta_o(a_5) = (\lambda)$	$\gamma_i(a_5) = (r_1, r_2), \gamma_o(a_5) = (\lambda)$
$\delta_i(a_6) = (a_1), \delta_o(a_6) = ((a_2), (a_3))$	$\gamma_i(a_6) = (\lambda), \gamma_o(a_6) = (\lambda)$
$\delta_i(a_7) = (\psi, a_4), \delta_o(a_7) = ((a_5))$	$\gamma_i(a_7) = (r_2), \gamma_o(a_7) = (\lambda)$

단위업무에 후속하는 출력 자료 저장소들의 집합과 연결하는 것 중 하나이며 $\gamma_i : A \rightarrow P(R)$ 은 하나의 단위업무에 선행하는 입력 자료 저장소들의 집합과 연결하는 관계를 나타내는 것 중 하나이다.

표 2는 그림 1을 ICN의 명세로 표현한 것이다.

4. 설계 및 구현

본 논문에서 구현된 모델링 작업자의 권한에 따라 작업을 할당함으로써 작성될 모델의 일관성 유지를 위해 정의 틀에서 필요한 세션, 조직, 응용 프로그램, 관련 데이터를 별도의 에이전트로 구현하고, 각각의 에이전트는 별도의 관리기능을 가지고 있다. 그림 2는 이 논문에서 제안된 워크플로우 모델링 시스템 구조를 나타내고 있다.

세션 에이전트를 통하여 전달된 작업자 정보는 DB 에이전트를 통하여 식별된 작업자의 권한에 따라 각 에이전트를 호출할 수 있다. 호출된 에이전트에서 정의된 기능의 작업을 수행하며 특히, 업무 프로세스를 정의하는 모델러 에이전트에서는



(그림 2) 워크플로우 모델링 시스템 구조

저자스 에이전트 조직 에이전트 응용 프로그래 에이전트를 호출할 수 있어 워크플로우 프로세스 정의에 필요한 요소를 정의할 수 있다. 정의된 모든 정보는 DB 에이전트를 통하여 저장될 수 있으며, 워크플로우 실행 시 모델링 시스템에서 정의된 정보를 바탕으로 워크플로우 엔진은 실행되게 된다.

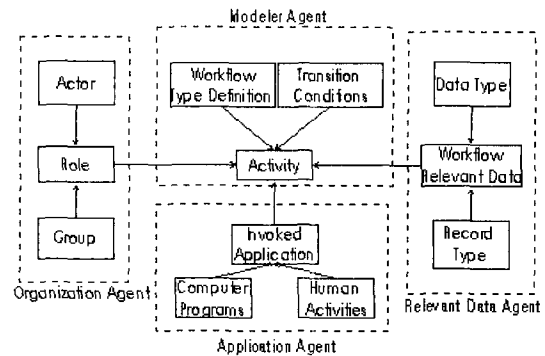
4.1 다중 에이전트 구성도

그림 3은 본 논문에서 제시된 워크플로우 모델에서 사용되는 각 에이전트 구성 요소를 나타내고 있다. 본 논문에 워크플로우 모델링 시스템은 크게 빌트타임 클라이언트와 모델러, 조직, 관련 데이터, 응용프로그램, DB 에이전트로 구성된다. DB 에이전트를 제외한 나머지 에이전트는 별도의 UI를 가지고 있으면, 각각의 UI는 자바로 구현되었기 때문에 배터나 플랫폼에 구애를 받지 않는다. 또한 DB 에이전트 JDBC를 사용하기 때문에 하위 데이터베이스에 대한 고려를 할 필요가 없다.

워크플로우 관리 시스템에서 업무 프로세스는 단위 업무들의 모임이다. 즉 본 논문에서 제시된 각 에이전트는 단위업무에 대하여 각각의 에이전트에서 부여된 다음과 같은 기능을 정의한다.

4.1.1 모델러 에이전트

모델러 에이전트에서 워크플로우 타입 정의와 전이 조건, 단위업무를 정의한다. 워크플로우 타입



(그림 3) 모델링 시스템에서 다중 에이전트 구성도

정의는 기본적으로 워크플로우 프로세스의 이름, 버전 번호, 프로세스의 시작 및 종료 조건, 보안, 컨트롤 데이터 등에 속성이 있고, 전이 조건은 전이 조건 또는 실행 조건을 기술한다. 모델러 에이전트의 기능은 업무 프로세스, 모델, 단위업무를 정의한다. 모델러는 메뉴바, 트리, 테이블, ICN속성 및 프로세스 정의 부분으로 나누어진다.

4.1.2 조직 에이전트

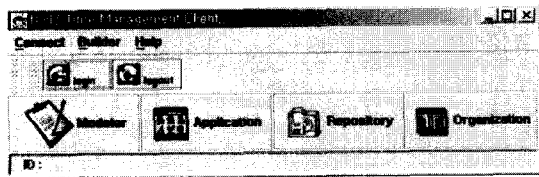
조직은 조직원 및 각각의 조직들에 정보를 정의하며, 모델링 도구에서 단위업무 정의 부분과 프로세스 정의부분에 역할을 할당할 때 관련 데이터를 호출하여 사용된다. 기업내의 물리적 조직 관계를 의미하는 수직조직과 기업내의 역할에 따라 정의되는 수평조직으로 구분된다.

4.1.3 관련 데이터 에이전트

워크플로우에서 Input/Output 처리 기능은 관련 데이터의 처리를 의미한다. 관련 데이터는 워크플로우 엔진이 주어진 프로세스를 수행하는 동안에 소용이 되는 데이터를 의미한다

4.1.4 응용 프로그램 에이전트

응용 프로그램 에이전트는 업무 프로세스 정의 시 단위업무에서 수행되어야 할 프로그램을 정의해 주는 역할을 수행한다.



(그림 4) 빌드타임 클라이언트 메인 화면

4.2 업무 프로세스 정의 예

업무 프로세스 정의하기 위하여 사용자는 다음 그림 4와 같은 Build Time Client 메인 화면으로 사용자는 로그인을 함으로써 사용자의 권한 및 자원 액세스 권한을 인증 받는다. 현재 권한 레벨은 3 단계로 구분하고 있다.

4.2.1 사용자 로그인 권한 레벨

- 1 단계 (Supervisor 유저) : 모든 에이전트를 관리하는 유저로서, 시스템 관리자이다.
- 2 단계 (Administrator 유저) : 각각 에이전트를 관리하는 유저로서, 모델러의 경우 비즈니스 프로세스를 정의하는 유저이다.
- 3 단계 (일반 유저) : Build Time Client 에서 접근 권한이 없는 유저

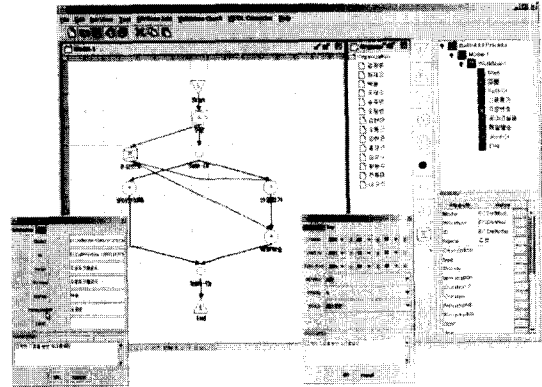
4.2.2 업무 프로세스 정의 과정

사용자 로그인 레벨에 따라 화면에 보여지는 모델링 시스템의 에이전트는 서로 다르게 나타나며, 그림 5는 1단계 유저로 로그인하여 업무 프로세스를 모델링 하는 예를 나타내고 있다.

모델러 에이전트를 이용하여 그래픽 적으로 업무 프로세스를 정의하며, 파란선은 단위업무의 제어 흐름을 의미하며, 빨간선은 데이터의 흐름을 의미한다. 각 에이전트에 정의된 내용은 모델러 에이전트의 메뉴를 통하여 호출하여 이용할 수 있다.

모델러 에이전트의 오른쪽 사다리의 화면에는 현재 정의된 단위업무와 저장소가 나타나고 있다.

모델러 에이전트의 우측 하단의 테이블은 선택된 단위업무의 속성을 보여주는 테이블이다. 테이블



(그림 5) 주문처리 업무 프로세스 정의 예

에서 직접 유저가 각각의 속성을 입력하여 변경할 수 있고 경우에 따라서는 다이얼 박스 창의 내용을 선택하여 속성을 변경할 수 있다.

하단에 보이는 두 개의 다이얼 박스 창은 작성되는 모델의 속성과 워크플로우를 정의하는 창으로 아이디, 이름, 설명, 책임자, 실행자, 유효시간 및 비즈니스 프로세스 상태 값 등을 선택한다.

5. 결 론

본 논문에서는 WfMC 표준 기구의 명세인 Interface 1 부분인 워크플로우 프로세스 정의 모델의 교환 서비스를 위하여 다중 에이전트 기반의 워크플로우 모델링 시스템을 구현하였으며, 작성된 모델의 일관성을 유지하기 위하여 업무 프로세스 정의를 위해 필요한 세션, 응용 프로그램, 데이터, 조직과 같이 각각의 에이전트로 분류하여 사용자 관점에서 GUI 기반의 인터페이스를 이용하여 보다 편리한 비즈니스 프로세스 정의 및 관련 데이터의 관리 및 수정을 할 수 있게 하였다

또한, 작성된 모델의 모든 워크플로우 엔진과의 연동을 위하여 WfMC에서 정의한 WPDLo import/export 할 수 있는 기능을 구현하고, 프로세스 정의를 위한 ICN 모델을 이용하여 작성된 모델을 검증 할 수 있다.

그러나 현재의 모델링 시스템은 정적 모델링

워크플로우의 실행중에 작성된 변경을 지원하는 것과 같은 동적변경을 지원하지 못하는 것 만을 지원하는 모델링 시스템으로서 향후 워크플로우의 실행 시점에 작성된 모델의 동적변경을 지원하기 위해서는 작성된 모델의 의미를 검증 할 필요가 있고 또한 발생될 수 있는 예외상황에 대하여 처리 메커니즘을 제시해야만 한다.

향후 연구과제로는 워크플로우 모델링에 대한 예외상황의 분류를 통하여 발생될 수 있는 예외상황의 처리 메커니즘을 적용하여 동적 변경을 지원하는 모델링 시스템을 구현하여야 할 것이며, 작성된 모델의 의미적 검증을 위한 연구가 진행되어야 할 것이다.

참 고 문 헌

- [1] 원재강, 김학성, 김광훈, 정관희, 워크플로우 표준화 동향 분석, 인터넷정보학회, 제1권 1호, 2000년 9월.
- [2] Kwang-hoon Kim and Su-ki Paik and Moon-uee Lee, "GROUPWARE : CSCW(Computer) Supported Cooperative Work", 경기대학교 국제경영논문집(제36판)별쇄본, 1995.
- [3] Workflow Management Coalition, "The Workflow Reference Model Document Number TC00-1003 Version1.1", WfMC, Jan,1995.
- [4] 전남대학교, 트랜잭션 워크플로우 모델링 도구 개발 최종연구 보고서, 한국전자 통신 연구소, 1998.
- [5] 박정훈, 에이전트의 소개, <http://islab.hanyang.ac.kr/~jhpark/hhome/agent>.
- [6] 이재호, 에이전트 시스템의 연구 및 개발 동향, 정보과학지 제 18 권, 제 5 호, pp. 4~9, 2000.
- [7] 이경하, Mobile Agent의 구현 <http://dmlab.comeng.chungnam.ac.kr/~bart/>.
- [8] Kwang-Hoon Kim and Su-Ki Paik, Actor-Oriented Workflow Model, The Second Cooperative Database Systems for Advanced Applications, Wollongong Australia, March 1999.
- [9] Clarence A. Ellis, Gary J. Nutt, The Modeling and Analysis of Coordination Systems, University of Colorado/Dept. of Computer Science Technical Report, CU-CS-639-93, Jan. 1993.
- [10] Clarence A. Ellis and Gary J. Nutt, Office Information Systems and Computer Science, Computing Surveys, Vol. 12, No. 1, March 1980.
- [11] Gary J. Nutt, Using Workflow in Contemporary IS Applications, Computing Surveys, CU-CS=663-93, Aug 1993.
- [12] 진훈, 김학성, 김광훈, 백수기, 자바 기반 ICN 모델링 도구의 설계 및 구현, 정보과학회, 제6권 1호, 1999년 7월.
- [13] Clarence A. Ellis, Goal Based Models of Groupware, University of Colorado.

◎ 저자 소개 ◎



김 학 성

1993년 2월 경기대학교 전자계산학과 졸업(학사)
1995년 2월 경기대학교 대학원 전자계산학과 졸업(석사)
2003년 2월 경기대학교 대학원 전자계산학과 졸업예정(박사)
1998년~현재 : 동남보건대학 컴퓨터응용과 교수
관심분야 : Workflow Mining, CSCW, Workflow Exception Handling
E-mail : amang@dongnam.ac.kr



김 광 훈

1984년 경기대학교 전자계산학과 졸업(이학학사)
1986년 중앙대학교 대학원 전자계산학과 졸업(이학석사)
1994년 콜로라도대학교 대학원 컴퓨터과학과 졸업(이학석사)
1998년 콜로라도대학교 대학원 컴퓨터과학과 졸업(이학박사)
1998년~현재 : 경기대학교 정보과학부 교수
관심분야 : Workflow, Groupware, CSCW, Database, Distribute Computing
E-mail : kwang@kyonggi.ac.kr



백 수 기

1972년 연세대학교 토목공학과 졸업(학사)
1979년 동국대학교 대학원 전산통계학과 졸업(석사)
1992년 동국대학교 대학원 전산통계학과 졸업(박사)
1998년~현재 : 경기대학교 정보과학부 교수
관심분야 : Computer Network, Workflow, Network Algorithm
E-mail : skpaik@kuic.kyonggi.ac.kr