

DB에이전트를 이용한 전자상거래 워크플로우 모델링 도구 설계

오종태*

Electronic Commerce Workflow Modeling Tool Design Using Database Agent

Jong-Tae Oh *

요약

모델링 도구는 그룹 단위의 실시간 협업 지원을 가능하게 하는 워크플로우 비즈니스 프로세스를 정의한다. 오늘날 대부분의 모델러는 비즈니스 프로세스 정의과정에서 동시에 여러 사용자가 작업을 수행해야 할 사안이 많이 발생하지만 동시작업 지원을 지원하지는 못하는데 미흡하다. 본 논문에서는 여러 사용자가 동시에 작업이 가능하게 하기 위해 확장 ICN을 이용한 CDN을 생성하는 알고리즘과 아키텍처를 제안하고 데이터베이스 에이전트를 이용하여 에디터를 설계한다. 에디터를 통하여 정의된 워크플로우 비즈니스 프로세스 모델들의 집합은 데이터베이스에 저장되고, WfMC에서 표준화된 워크플로우 프로세스 정의 언어(WPDL)로 변환되게 된다. 이 방법은 프로세스 정의 과정에서 발생하는 작업 지연 비용을 최소화하여 워크플로우 처리성능을 향상 시킬 수 있다.

Abstract

Modeller is defined as the office work procedures(business processes), and that is systemically based upon the real-time collaborative operations by a set of actors, which is called group. This paper describes the design of the ICN editor that is operable under the real-time collaborative computing environment. We use the database agent that enables the ICN editor to operate among multiple actors(group) through the event-driven collaboration platform. Consequently, a set of workflow and business processes defined through this editor is not only stored onto database but also transformed into the format of the workflow process definition language(WPDL) that is a standardized workflow description and specification language proposed by the workflow management coalition(WfMC). This method can improve the performance of workflow processing by minimizing the workflow execution cost occurred during workflow definition.

▶ Keyword : 워크플로우, 프로세스, 모델러, ICN, PDN

* 청운대학교 컴퓨터학과

※ 본 연구는 2003학년도 청운대학교 학술연구조성비 지원에 의하여 연구되었음.

그러나 액티비티간의 제어흐름 및 실행 중 제어의 통제 변경을 반영하지 못 한다.

본 논문에서는 협동 작업을 통하여 조직의 업무 프로세스를 설계할 수 있는 워크플로우 모델링 도구를 설계하였다.

I. 서론

정보화 사회에서 사람들 간의 상호 활동 및 관계는 과거 10년간 큰 차이가 없지만, 그들을 지원하기 위한 수단이나 방법 면에서는 많은 차이가 있다.

즉, 컴퓨터 기술과 전자통신 기술의 급진적인 발전과 이들 간의 기술적 수렴은 바로 전자적인 작업환경(Electronic Workplace)이라고 하는 새롭고도 매우 효율적인 상호 작용 지원 수단 및 방법을 제공하기 시작했다. 전자적인 작업 환경이란 전 조직체적 통합 시스템으로 정보 처리 활동과 정보 통신 활동의 통합을 통해 조직체 구성원들 간의 상호 활동 및 관계를 정의하고 지원하는 개선된 형태의 조직 활동 지원 수단 및 방법이다(1).

이러한 전자적 작업환경을 구현하기 위한 종합적인 연구 분야로서 컴퓨터통신 분야뿐만 아니라 사회학 분야나 언어학 분야, 경영학 분야 등의 다각적인 협력 관계를 통해 성공적으로 완성될 수 있는 매우 다중적인 연구 분야라 할 수 있다. 워크플로우는 그룹의 작업 및 활동이 어떻게 이루어 지는가를 분석하고 컴퓨터를 비롯한 첨단 기술을 이용하여 어떻게 그룹 활동을 효과적으로 지원할 수 있는가를 연구하는 광범위한 분야이다.

기존 워크플로우 시스템의 경우에 한 사람만이 조직 내 업무 프로세스를 디자인하는 형태였다. 그러나 빠르게 발전하는 기업의 구조와 복잡하고 다양하게 변화는 기업의 규모에서 모든 업무 프로세스들을 한 사람이 모델링하고 관리하기는 어렵게 되었다. 이러한 프로세스 모델링 작업을 여러 사람이 동시에 실시간으로 함께 진행할 수 있다면 효과적이고 안정적인 결과를 얻을 수 있을 것이다.

업무프로세스를 모델링하기 위한 연구들이 있어왔다. PN(Petri Net)은 사건을 트랜지션(Transition)으로 언급한다. 트랜지션이 발생하기 위해선 플레이스(Places)에 저장된 조건이 만족해야 하며 이러한 사건을 가시적으로 표현 가능하여 이해하기가 편리하지만 동기적인 사건에 대한 처리가 어렵다(2).

ICN(Information Control Net)은 액티비티, OR노드, AND노드, 선으로 노드들 간의 선후관계로 프로세스를 표현한다(2, 3).

II. 워크플로우

워크플로우 기술은 크게 일련의 단위 업무들로 구성되는 워크플로우를 정의하고 분석하는 워크플로우 모델 분야와 각 업무의 실행과 그들 간의 흐름을 제어하는 워크플로우 관리 시스템 분야로 나뉘어 진다. 특히, 워크플로우 관리 시스템은 어느 조직체내에서나 조직체 간에 구성되어 있는 사무 업무 프로시저 또는 사무 업무 프로세스(Business Process)를 기반으로 하는 업무의 흐름을 제어하고 자동화시키기 위한 기술이다. 워크플로우의 사무 업무 프로세스는 일련의 물리적 또는 논리적 단위의 단위 업무(Activity)들과 이들의 수행을 담당하는 역할(Role) 및 참여자(Participant), 그리고 단위 업무들 간에 전달되는 문서 또는 정보들로 구성된다(4, 5).

일반의 사무 업무 프로세스는 많은 단위 업무들로 구성되어 있다. 사무 업무 프로세스의 실행을 위해 각각의 단위 업무는 응용프로그램 및 사용자 그리고 규칙들로 연결된다. 이러한 사무 업무 프로세스 안에서 단위 업무 간에 정의되는 실행 순서의 흐름을 제어 흐름(Control flow)이라 한다. 그리고 각각의 단위 업무들은 "OR 또는 AND"노드를 통해 선택적 또는 병렬적 실행 제어 순서 관계를 갖는다. 본 논문에서는 이와 같은 단위 업무 간에 실행 순서에 대한 제어 흐름을 분석하여 각 단위 업무 간에 존재하는 실행 제어의 존성 관계를 분석하는 도구를 설계하였다.

1. 빌드타임 구조

비즈니스 프로세스는 하나 또는 그 이상의 분석, 모델링과 시스템 정의 기술을 이용함으로써 실제계로부터 공식적으로 컴퓨터가 수행 가능한 표현으로 변환되어 프로세스 모델, 프로세스 템플릿(template), 프로세스 정의라고 불린다. 프로세스 정의는 인위적인 정의와 워크플로우의 정의를 포함하는 프로세스의 컴퓨터적인 표현이다.

이러한 프로세스 정의는 이산되어 있는 많은 단위업무 과정들과 컴퓨터와 인간과의 상호작용에 있어서의 "OR 또는 AND" 동작과 다양한 단위업무 과정을 통해서 진행되는 프로세스의 진행을 통제하는 규칙들로 구성된다. 또한 실행 시 단위업무에 필요한 프로그램 간에 주고받는 데이터 수행을 맡게 되는 참여자나 물들도 기업 내 조직도를 바탕으로 정의를 해야 한다(6).

2. 빌드타임 기능

빌드타임 기능은 크게 다음과 같이 나누어 볼 수 있다.

첫째, 비즈니스 업무 프로세스 정의 기능(Modeler), 프로그램 정의 기능(Application), 데이터 정의 기능(Repository), 그리고 조직 정의 기능(Organization)이다 [7].

〈그림 1〉은 DB 에이전트와 빌드타임과 연동된 것을 보여준다.

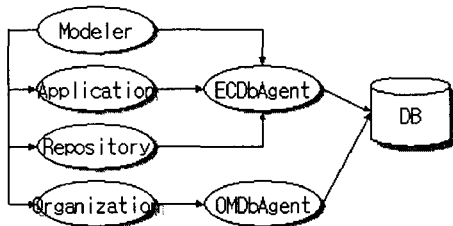


fig 1. DB 에이전트와 빌드타임 연결관계

모델러는 비즈니스 업무 프로세스를 모델링하고, 업무 프로세스를 구성하는 각 작업에 대한 정의 및 작업들 간의 연결 관계를 지정한다. 응용 모듈은 응용 프로그램을 등록하는 기능을 가지고 있으며, 새로운 응용 프로그램을 등록, 수정, 삭제한다. 레파지토리 모듈은 데이터 타입 및 데이터 구조를 등록하는 기능을 가지고 있으며, 새로운 타입 및 구조를 등록, 수정, 삭제한다.

조직 모듈은 책임자 및 수행자, 역할, 조직도를 등록, 수정, 삭제하는 기능을 가지고 있다. 이러한 4개의 빌드타임 모듈은 DB의 호출과 저장을 두개의 DB 에이전트 (ECDBAgent, OMDbAgent)를 통해서만 DB에 접근한다.

III. DB 에이전트

1. 아키텍처

아키텍처는 객체지향 분산 엔터프라이즈 애플리케이션을 위한 컴포넌트 아키텍처이다. 그림 2은 EJB 컨테이너(Container)의 모델을 보여주는 것이다.

엔터프라이즈 빈을 구현하기 위해서는 객체, 홈 인터페이스(Home interface), 빈(bean)이 필요하다.[8] 객체는 빈이 어떤 일을 수행하기 위해 외부 세계에 제공하는 빈의 비즈니스 메소드를 정의한다. 그리고 클라이언트의 모든 메소드 호출을 가로채서 배치 디스크립터(Deployment Descriptor) 설정에 기반한 빈에 대해서 트랜잭션, 상태관리, 영속성, 보안 서비스를 이행한다. 홈 인터페이스는 새로운 빈을 생성, 제거하고, 찾아내는 등의 빈의 라이프 사이클 메소드를 정의한다. 빈 클래스(Bean class)는 실제로 빈의 비즈니스 메소드를 구현한다.

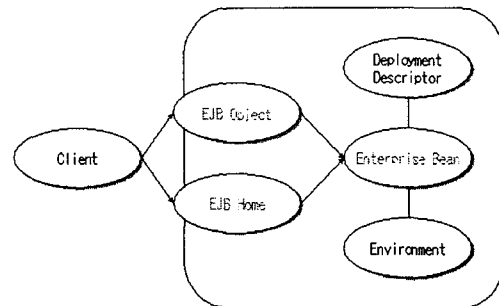


fig 2. 엔터프라이즈 빈 모델

2. 자원 관리

대규모 시스템에서는 수천에서 수백만 개까지의 객체를 동시에 사용하게 된다. 서버에서는 이러한 문제점의 해결 방안으로 인스턴스풀링(Instance Pooling)과 스와핑의 두 방법을 사용한다[9].

서버는 클라이언트에 의해 사용되거나 또는 사용되지 않는 과정을 반복하면서 동작하게 되는데, 빈은 클라이언트의 요청이 오고 서비스를 제공하면서 상태를 전이하게 된다.

컨테이너에서는 빈의 인스턴스를 생성하고 객체와 연결하고, 더 이상 사용되지 않는 인스턴스를 삭제하는 등의 과정을 거쳐 빈의 상태를 변화시킨다. 컴포넌트는 크게 엔티티 빈(Entity Bean)과 세션 빈(Session Bean)으로 나누어지는데, DB 에이전트는 스테이트풀(Stateful) 세션 빈을 사용한다. 그림 3는 스테이트풀 세션 빈의 라이프 사이클을 보여준다. <그림 3>에서 처음 상태는 DB 에이전트의 인스턴스가 생성되지 않은 것이다. 그리고 클라이언트의 요청이 생기면 메소드 준비 상태로 전이된다. 그 다음 상태는 빈 인스턴스가 클라이언트로부터 메소드를 서비스하고 있지 않는 비활동 기간인데, 리소스를 절약하기 위해 컨테이너는 빈의 대화 상태를 보존한 채 빈의 인스턴스를 메모리로부터 제거하여 비활성화 시킬 수 있다[6].

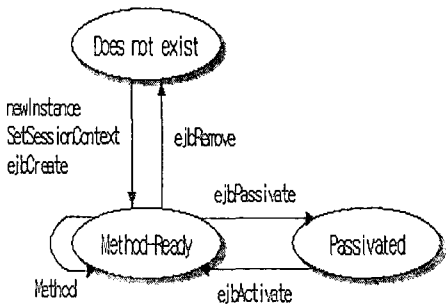


fig 3. DB 에이전트의 라이프 사이클

3. DB 에이전트의 클래스 구성

DB 에이전트의 클래스 구성은 <그림 4>와 같다. 객체와 홈 인터페이스는 java.rmi.Remote를 계승한 javax.ejb.EJBObject를 상속 받는다. 그리고 빌드타임의 구성 테이블들을 두 개의 엔터프라이즈 빈의 헬퍼(Helper) 클래스로 구현하였다.

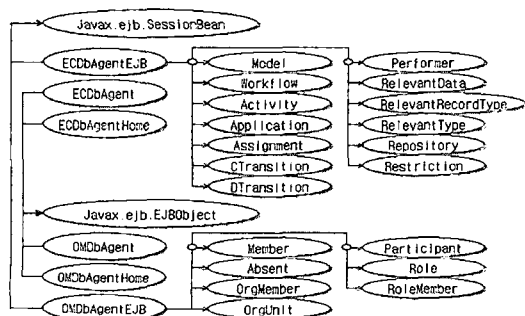


fig 4. DB 에이전트의 클래스 다이어그램

4. DB 에이전트의 테이블 구성

테이블 종류는 <그림 4>에서와 같이 ECDBAgent에 Model, Workflow등 13개, OMDBAgent 7개로 구성되어 있으며 그 중 Workflow의 테이블 구조는 표 1과 같다.

표 1. DB 에이전트의 Workflow 테이블

ECDefWorkflow		
model_id	varchar2(63)	모델 식별자
id	varchar2(63)	프로세스 식별자(PK)
name	varchar2(63)	프로세스 이름
description	varchar2(1024)	설명
priority	number	우선 순위
limit	varchar2(10)	예상 수행 시간
duration_unit	number	수행 시간 단위
author	varchar2(63)	작성자
version	varchar2(20)	버전
created	varchar2(19)	작성일자
responsible	varchar2(63)	실행 시간 책임자
valid_from	varchar2(19)	유효 기간 시작 일자
valid_to	varchar2(19)	유효 기간 만료 일자
status	number	프로세스 정의 상태

<그림 5>은 ECDBAgent의 내부 동작 과정을 설명하고 있다. 클라이언트는 서버로 연결하고 특정 EJB Home (ECDBAgentHome, OM-DbAgentHome)을 얻기 위해 JNDI를 사용한다.

5. DB 에이전트의 동작

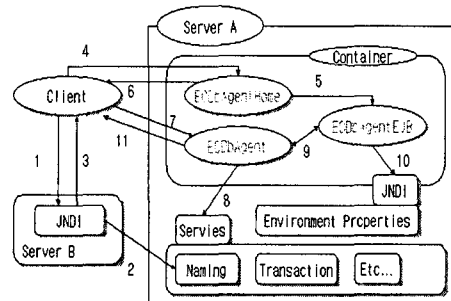


fig 5. DB 에이전트의 내부 동작 과정

- (1) ~ (3) InitialContext의 생성자를 통해 프로퍼티에 지정된 서버를 찾고 lookup() 메소드는 JNDI 서비스에서 ECDBAgent로 바인딩된 ECDB-AgentHome 인터페이스의 레퍼런스를 얻어온다. 다음으로 narrow()를 통해 ECDB-AgentHome 인터페이스를 사용할 수 있게 된다.
- (4) 클라이언트는 ECDBAgentHome 오브젝트를 통해 ECDBAgentEJB 생성을 요청한다.

- (5) ECDbAgentHome은 ECDbAgentEJB를 생성한다.
또한 클라이언트의 요청이 있을때는 ECDbAgent-EJB를 제거할 수 있다.
- (6) ECDbAgent에 대한 레퍼런스가 클라이언트에게 반환된다.
- (7) 클라이언트는 ECDbAgent를 통해 ECDbAgentEJB를 호출한다.
- (8) ECDbAgent는 자신에게 요청된 모든 메소드를 Transaction, Persistence, Security로 덮는다.
- (9) ECDbAgent는 메소드 호출을 ECDbAgent로 전달하고, 그에 대한 반환 값(return value)을 받는다.
- (10) 필요하다면 JNDI를 통해 배치 디스크립터(Deployment Descriptor)에 있는 프로퍼티에 접근한다.
- (11) ECDbAgent는 비즈니스 호출에 의해 획득한 값을 클라이언트에게 반환한다.

IV. 워크플로우 모델

워크플로우 모델이란 조직이나 작업그룹에서의 업무환경과 업무 프로세스를 적절히 표현한 것이다. 워크플로우 모델은 조직의 모습을 할당업무, 업무자, 역할, 단위 업무와 자료저장소의 모습으로 표현된다[2].

1. ICN

ICN(Information Control Net)은 사무실의 개념을 일련의 관련된 프로세스의 집합으로 정의하며 이러한 프로세스는 전후관계가 존재하는 단위 업무들의 집합으로 표현된다. ICN은 그림 형태로 프로세스, 단위업무, 저장소 전후관계를 나타내는 제어흐름과 데이터흐름을 표현한다. ICN 제어흐름 그래프는 큰 원으로 표현되는 일련의 액티비티와 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 [2,3].

2. 정형적인 ICN 모델

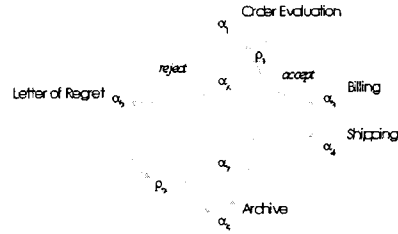


fig 6. 주문처리 관계를 나타내는 워크플로우

기본 ICN은 4개의 구성 요소인 프로세스, 단위업무, 선 후관계와 자료저장소로 구성되며 A를 일련의 단위 업무들의 집합이라 하고, R을 일련의 자료 저장소의 집합이라 할 때 수식적인 표현과 정의는 <그림 7>과 같다.

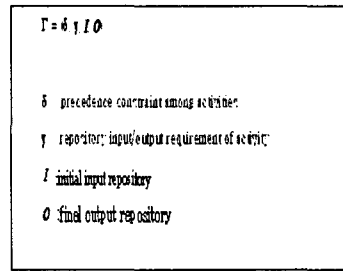


fig 7. ICN 표현

- (1) I는 초기에 입력되는 자료 저장소들의 유한집합이며 ICN의 실행 전에 어떠한 외부의 프로세스에 의해 로드(load) 되어져야 한다고 가정한다.
- (2) O는 마지막으로 출력되는 자료 저장소들의 유한집합이며 ICN의 실행 후에 어떠한 외부의 프로세스에 의해 이용되는 정보를 포함하고 있다고 가정한다.
- (3) $\delta = i \ o$
여기서 $o : A$ (A)은 하나의 액티비티를 후행하는 액티비티들의 집합에 연결하는 관계를 나타내며 $i : A$ (A)은 한 액티비티를 선행하는 액티비티의 집합에 연결하는 관계를 나타내는 관계이다.
- (4) $\gamma = i \ o$
여기서 $o : A$ (R)은 하나의 액티비티를 후속하는 액티비티 집합을 출력자료 저장소의 집합과 연결하는 것 중 하나이며, $i : A$ (R)은 하나의 액티비티를 선행하

는 액티비티 집합들을 입력 자료 저장소의 집합과 연결하는 관계를 나타내는 것 중 하나이다. <그림 7>에 대한 정형적인 ICN은 <그림 8>과 같다.

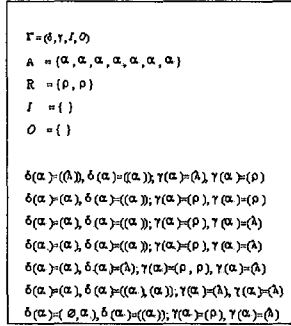


fig 8. 정형적인 ICN 표현

V. 제어 의존성 분석 메커니즘

1. CDN

ICN 모델을 바탕으로 제어 흐름 의존성을 분석하기 위한 CDN(Control Dependent Nets)을 정의한다. 또한 ICN으로부터 CDN을 생성하는 알고리즘도 제시하였다. 워크플로우의 제어 의존성 분석은 각 액티비티들 간에 존재하는 실행 의존 관계를 정의함으로써 구현될 수 있다. 즉, ICN 모델상의 OR 노드와 AND 노드 그리고 Loop 부분이 액티비티들 간의 실행 의존 관계를 형성하는데 중요한 역할을 한다.

CDN을 정의하기 위해 확장된 ICN을 이용한 용어를 재정의 할 수 있다.

$$\text{확장 ICN} = (\Gamma, \pi, \kappa, I, O)$$

(1) 확장 ICN에서 W 는 액티비티 간의 순서를 나타내며, W 의 길이는 |W|이다. |W| = 0 이면 액티비티가 아무 일도 하지 않으며 empty 부른다.

|W| 가 nonempty 이면 시작 액티비티 u 와 마지막 액티비티 v 의 순서는 u v 표현한다.

(2) 확장 ICN 의 우수성

확장 ICN에서는 다음의 조건을 만족해야 한다.

① 는 초기의 액티비티 aI 와 마지막 액티비티인

$$aF \text{ 포함하면 } \delta(aI) = \{\emptyset\}, \delta O(aF) = \{\emptyset\}$$

② 에 모든 액티비트는 aI - aF 순서를 실행한다.

$u \in A$ forward dominates $v \in A$ iff every $v - vF$
 $u \in ; u$ properly forward dominates v
 iff $u \rightarrow v$ and u forward dominates v
 $u \in A$ strongly forward dominates $v \in A$
 iff u forward dominates v 이고 정수 $k \geq 1$ 일 때
 $|W| \geq 1$
 $a \in (A - \{aF\})$ immediate forward dominator ifd(a)
 일 때 $a - aF$ 는 a 의 proper forward dominator

2. 정형적인 CDN 모델

CDN은 프로세스의 각 단계에서의 OR노드 또는 AND 노드를 설계하기 위해 사용된다. CDN을 통해 각각의 액티비티간의 제어 이행 조건(control-transition conditions)을 효율적으로 발생시킬 수 있다.

제어 이행 조건은 액티비티 간의 제어 흐름 및 실행 중에 제어 흐름 동적 변경을 하기 위해 사용된다. CDN은 = (, c, S, E) 정의할 수 있고, A는 액티비티 집합이고, T는 제어변이 조건들의 집합이다.

$$= i \ o$$

$o : A$ (A) 는 다수의 액티비티가 하나의 액티비티에 의존하고, $i : A$ (A) 는 하나의 액티비티는 다수의 액티비티에게 의존한다. $c = c_i \ c_o$: control-transition conditions
 S 는 한정된 초기의 control-transition conditions
 E 는 한정된 종료된 control-transition conditions

3. CDN 생성 알고리즘

INPUT : An extended ICN OUTPUT : An Control Dependency net

BEGIN

For all $x \in A$ and $y \in A$ in an extended ICN
 IF x is strongly control dependent on y
 //OR or Loop Constructs Get a dependent flow
 ADD x To $o(y)$; ADD y To $i(x)$;

```
//Get control-transition conditions between x and
yADD ci (x) To co (y); ADD ci (x) To ci (x)
ELSE IF fd(y) ∈ x(fd(y)fd(ibd(y))OR idb(y) = )
//Get a dependent flow
ADD x To o (y); ADD y To I (x);
//Get control-transition conditions between y and x
Add ci (x) To co (y); ADD ci (x) To ci (x)
FI
FI
ROF
End
```

확장 ICN을 이용한 CDN을 생성하는 알고리즘을 구현한다.

4. CDN 생성

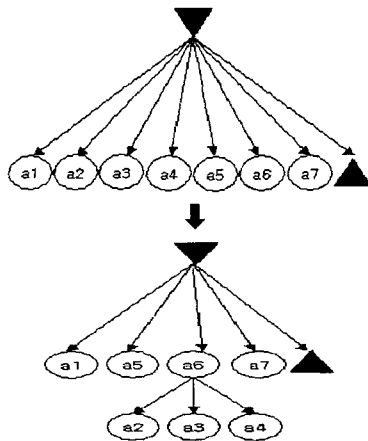


fig 9 CDN 모델

<그림 9>는 <그림 6>의 주문처리 워크플로우 대한 액티비티 간에 제어 의존성의 표현했으며 <그림 10>은 <그림 9>를 기초로 정형화된 CDN 모델을 보여준다.

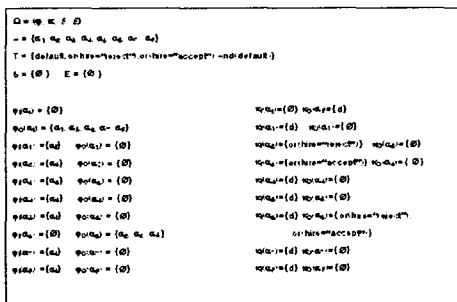


fig 10. 정형화된 CDN 모델

액티비티 1, 5, 6, 7, 출력값은 입력값인 I에 의해 결정되므로 1, 5, 6, 7, 출력값은 입력값에 제어 의존성을 가지고 있다. 액티비티 2, 3, 4는 6 OR 분기에 의해 제어 흐름이 이동함으로 6에 제어가 의존한다.

VI. 워크플로우 모델링 도구 설계

1. 모델링 요구 사항

(1) 시스템 목적

- 낮은 네트워크 부하 요구
- 응용 프로그램 안에서 동시 작업 요구

(2) 사용 목적

- 실시간 협력 작업 지원
- 유연한 What You see is What I See(WYSIWIS)
- 자세한 그룹 인지 정보 제공

(3) 접근방법

- 복제 구조를 선택
- 단일사용자 인터페이스를 다중사용자와 같이 균등하게 분산
- 실시간 협력 작업을 위한 최적의 분산알고리즘
- 요구가 있을 때마다 자동 교체 지원

2. 모델러 설계기준

본 논문에서 실시간으로 그룹작업을 하는 구성원 협동작업을 할 수 있도록 지원하는 저작도구를 설계한다. 협력기능을 투명하게 제공하는 응용프로그램 (Collaboration Transparency Application) 방식을 지원하며 그 중에서 복제구조를 택하여 기존에 이미 익숙하게 사용하던 프로그램 등을 매개소프트웨어를 통하여 연결함으로써 사용자의 응용 프로그램 사용에 있어서 각종 부담을 줄일 수 있게 하고 화면 전체를 복사하는 구조가 아니라 작업을 통하여 발생하는 변화(Event)만을 공유시킴으로써 작업 속도의 증가 및 네트워크의 부하를 줄일 수 있도록 한다.

3. 아키텍처

실제로 그룹작업을 지원하는 도구를 개발하기 위해서는 두 가지의 구조가 있다. 그것은 서버 중심형 구조와 복제 구조이다.

〈그림 11〉의 서버 중심형 구조는 그룹작업을 지원하는 모든 기능을 서버 측에서 담당하기 때문에 병행처리의 어려움이 있다.

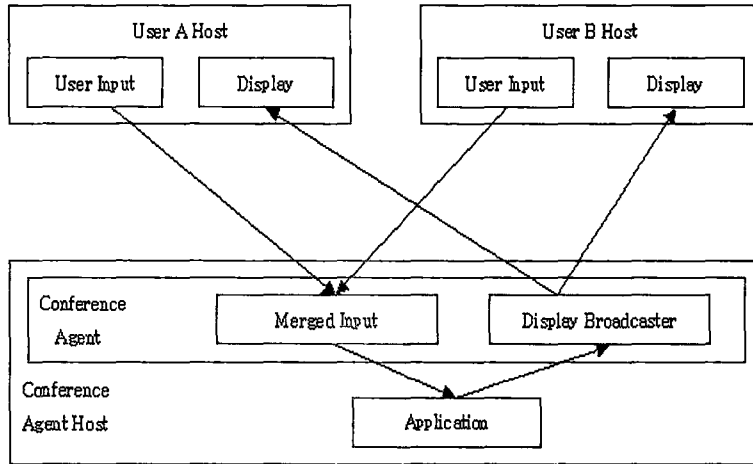


fig 11. 서버 중심형 구조

그리고 응용 프로그램을 공유하는데 있어서 공유된 응용 프로그램의 인스턴스의 화면 모습만을 보여주는 Display Broadcasting방식을 취하기 때문에 화면의 작은 부분에 대한 수정이 발생했을 때에도 화면상의 모든 내용을 전달해야 하는 이유 때문에 네트워크의 부하를 크게 요구한다.

〈그림 12〉는 복제 구조로 서버 중심형 구조와 반대의 특징을 지니고 있다. 참여자들은 각 응용 프로그램의 복사본을 가지고 작업함으로써 병렬 처리에 의한 동시 작업을 할 수 있다. 화면 처리 방식도 이벤트(Event) 처리 방식을 이용하기 때문에 그 때 그 때 발생한 이벤트만 처리하기 때문에 화면모습만 보여주는 서버 중심형 구조보다 네트워크의 부하를 크게 요구하지 않는다.

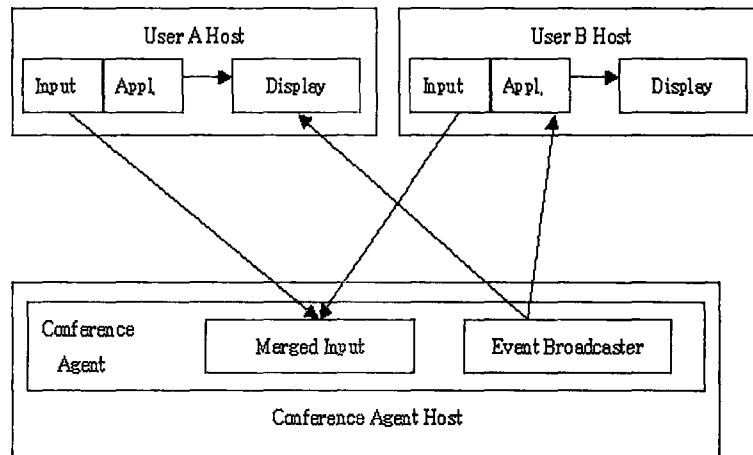


fig 12. 복제 구조

4. Session & Floor 제어

개발 틀을 서버 중심형으로 설계함으로써 한 사람이 로컬 환경에서 작업 할 수 있으면, 처음 Session을 연 사용자가 Server & Moderator의 역할을 하도록 규정한다.

워크플로우 모델링 도구의 설계에서 Process Modeling, analyzing, simulation은 ICN모델 기법을 사용한다. 그룹웨어 특성인 정보 및 공간의 공유, 그룹의 상호활동 알리기, 구성원들의 공평성, 이종의 언어지원은 그룹웨어 저작도구를 이용한다.

ICN모델의 각 구성 요소인 시작 노드, 종료 노드, 액티비트로 구성 되어 있으며, 중앙에는 사무업무 프로세스를 정의할 수 있는 작업 창과 하단부는 각 Client들을 위한 채팅 창을 만든다..

(1) Session 제어

Session 제어는 그룹의 구성원들이 그룹웨어 시스템을 통해 그룹 활동에 참여를 제어 받는 것을 의미한다. Session 제어는 협동작업 시 작업자의 참여 여부, 권한 부여 등의 기능을 제공 한다. 즉, 참여자를 합류 시키는데 있어서 제한인원이 얼마인가, 참여자들이 합류하고 헤어지는 것을 어느 시간이나 가능해야 하고, 회의에 방해가 되는 간섭이나 중단 등을 막을 수단, 서로를 알아볼 수 있도록 하는 기능들을 제공한다.

Session 제어 정보로 Session을 연 Server & Moderator 등의 정보가 있다. 호스트 이름은 Server & Moderator 의 IP 주소, 호스트 포트는 Server & Moderator에서 연 포트번호이다. Session name은 Session 방의 이름이다.

(2) Floor 제어

Floor제어는 Session 제어를 통해 그룹 활동에의 접근을 허용 받은 뒤에 그룹 활동에서의 주체를 결정하는 제어 기능이다. 참여자들이 공유해야 하는 자료에 대하여 다양한 종류의 권한을 제공해야 한다. 모든 사람들이 모든 것들에 대해 동시에 권한을 가질 수 있도록 하는 것이 가장 바람직한 방법이다. 이러한 방법은 협력관계의 일을 하지 않는 소수 개인에게 피해가 될 수 있다. 따라서 어느 정도의 중재 된 권한을 제공하는 것이 실수를 막고, 허가되지 않는 침입을 차단하면, 사람들이 다투는 상황을 막기 위한 수단이 제공해야 한다. Floor 제어 정보로 맨 위의 창은 Collaboration Control에 참여한 사용자 식별하기

위해 색으로 구별하며, 두 번째 창은 입력이나 제어권을 요청하는 창이고, 세 번째는 현재 자기 자신의 정보를 나타내며, 마지막 창은 현재 참여한 참여자들의 정보를 나타낸다.

Ⅶ. 결 론

응용분야의 대형화와 업무의 복잡성에 따라 동시 다수의 요구를 네트워크 부하를 줄이면서 처리 할 수 있는 능력이 워크플로우에 요구된다. 본 논문에서는 DB에이전트를 이용하여 인스턴스풀링과 스와핑의 방법을 사용하고 ICN(Information Control Net)모델을 바탕으로 제어 흐름 의존성을 분석하기 위한 CDN(Control Dependent Nets)을 정의하였으며, CDN을 생성하는 알고리즘도 제시 하였다. 알고리즘 분석을 통해 액티비티들 간에 존재하는 실행 의존관계를 형성한다. 이 방법은 기존의 빌드타임 모듈에서 정의된 워크플로우에 대한 구문(Syntax) 에러 테스트 기능과 의미적(Semantic) 에러 테스트 기능을 제공한다. 런타임 측면에서는 워크플로우의 적용에 가장 저해가 되는 요소인 동적 변경 지원 기능의 완결성 부족 문제에 대한 해결책 및 사무 업무 프로세스에서 단위 업무 간에 경로 및 경로수의 정보를 제공한다. 또한 복제구조를 택하여 사용자의 응용 프로그램 사용에 각종 부담을 줄일 수 있게 하고 작업 속도의 증가 및 네트워크의 부하를 줄일 수 있도록 한다.

향후 조직과 역할에 기반 한 의존성을 분석하여 멀티캐스트(Multi-Cast) 작업환경의 구현이 필요하다.

참고문헌

- [1] Edward A. Stohr, J. Leon Zhao, Workflow Auto-mation: Overview and Research Issues, Information Systems Frontiers, Volume 3, Issue 3, pp. 281- 287, 2001.
- [2] Clarence A. Ellis and Gary J. Nutt, "Office Information System and Computer Science", Computing Surveys, Vol.12, No. 1, March 1980
- [3] Kwang-Hoon Kim and Su-Ki Paik, "Actor-Oriented Workflow Model", The Second Cooperative Database Systems for Advanced Application, Wollongong Australia, March 1999
- [4] David Hollingsworth, "Workflow management Coalition The Workflow Reference Model", pp. 6-9, Workflow Management Coalition, 1995.
- [5] "Workflow management Coalition Terminology & Glossary", p. 10, Workflow Management Coalition, 1996.
- [6] Workflow Management Coalition Specification Document," The Workflow Reference Model", Document Number : TC-1003 Version 1.1, Jan 1995
- [7] "Interface 1 : Process Definition Interchange Process Model", Workflow Management Coalition, 1999.
- [8] "Enterprise JavaBeans Programming", pp. 1/1-3/29, Sun Microsystems, 2000.
- [9] "The JAVATM 2 Enterprise Edition Developer's Guide", Sun Microsystems, 2000.

저자 소개



오 종 태

1989 경기대학교 전산학 학사
 1998 경희대학교 대학원 전산학 석사
 2002 경기대학교 대학원 전산학 박사
 2003 ~ 현재
 청운대학교 컴퓨터학과 교수
 <관심분야> 워크플로우, 분산데이터베이스, 전자상거래