

협동 워크플로우 모델링 시스템

한수영[○] 성태혁 김광훈 백수기
경기대학교 전자계산학과 시스템소프트웨어 연구실
[sooyounghan[○]@kyonggi.ac.kr](mailto:sooyounghan@kyonggi.ac.kr)

A Cooperative Workflow Modelling System

Soo-Young Han[○] , Tae-Hyuk Sung , Min-Jae Park , Kwang-Hoon Kim
Dept. of Computer science, Kyonggi University

요 약

기업 업무의 절차와 규모가 복잡해지고 증가함에 따라 한 사람이 기업의 모든 업무 프로세스를 모델링하고 관리하는 것이 어려워지고 있다. 기존의 워크플로우 모델링 방식은 한 명(또는 소수)의 디자이너가 맡아 왔다. 하지만 본 논문에서 제시하는 역할(Role) 기반의 프로세스 모델링은 각 역할에 해당되는 디자이너 그룹이 프로세스를 정의하기 때문에 모델링 작업을 단순화 시킬 수 있었으며 기존의 워크플로우 시스템들이 지원하지 못 했던 워크플로우 모델링 작업을 실시간으로 지원할 수 있게 되었다. 또한 본 워크플로우 모델링 시스템을 통한 모델링 결과는 국제 표준화 워크플로우 정의 언어(Process Definition Language)를 제공함으로써, 기존의 워크플로우 시스템들과의 상호호환성을 제공 하고자 하였다.

1. 서 론

최근의 전자적인 작업환경(Electronic Workplace)과 단위업무의 변화에 보다 효율적으로 조직을 운영하고 대처하기 위한 기술이 워크플로우라 할 수 있다. 워크플로우는 일반적으로 전체 혹은 부분적으로 비즈니스 프로세스를 컴퓨터화하여 편리하게 하거나 자동화 한 것으로 정의되며 이러한 워크플로우에 관련된 기술들은 컴퓨터 및 통신 분야뿐만 아니라 사회학 분야나 언어학 분야, 경영학 분야 등의 다각적인 협력 관계를 통해서만 성공적으로 완성될 수 있는 매우 다중적인 분야라 할 수 있다.[1]

현재 국내외적으로 전자상거래를 비롯한 고객관리기술(CRM), 공급망 및 가치사슬망 관리기술(SCM), 데이터 및 응용 프로그램 통합기술(EAI), 기업자원계획기술(ERP) 등과 같은 최첨단 정보기술의 핵심 기반 기술로서 워크플로우 기술이 매우 중요하게 인식되고 있고, 워크플로우 응용 기술의 적용사례 측면에서도 그 수가 급속하게 증가하고 있기 때문에 워크플로우 기술을 적용하는데 있어서 반드시 워크플로우 모델링 및 워크플로우 기반 소프트웨어 개발방법론에 대한 연구가 매우 필요한 시점이다. 그리고 기업간 워크플로우(Cross-Organizational Workflow)의 핵심인 기업 간 워크플로우 모델과 효율적인 운영환경을 제공하는 것이 필요하다.

이러한 기업 간 프로세스를 정의하게 되면 여러 기업을 연결하는 글로벌 프로세스를 누가 어떻게 정의하고 운영하며 보관하는지가 최대 이슈가 된다. 운영상에 있어서 이기종의 워크플로우 엔진들간의 상호운영성 표준 인터페이스를 사용하며 모델 정의시에 글로벌 워크플로우 모

델링 시스템을 이용하려면 모델링 방법상의 상호운영성 지원이 필요하다. 하지만 기존의 워크플로우 모델링 도구를 사용하여 기업 간의 워크플로우 모델을 정의할 때 각각의 워크플로우 모델링 도구를 사용할 경우에는 각 기업의 로컬한 워크플로우 프로세스들을 통합하여 하나의 글로벌 워크플로우를 생성하는데 어려움이 있다. 본 논문에서는 조직간의 독립성을 보장하면서 기업 간 워크플로우 프로세스 정의를 지원할 수 있는 모델링 방법으로서 워크플로우 모델을 제안하며 협동 워크플로우 모델링 시스템을 설계 및 구현 하였다.

2. 협동 워크플로우 모델

WfMC(The Workflow Management Coalition)에 의해 채택되어진 Interworkflow는 각 조직 간의 워크플로우를 지원하는 표준모델이지만 글로벌 프로세스를 먼저 정의한 뒤에 각 조직에 해당하는 액티비티 또는 서브프로세스를 각 조직에서 정의하도록 한다. 이것은 전체를 먼저 고려하고 부분 부분을 정의하는 Top-Down 방식이며 조직 간의 비즈니스 프로세스를 구성하는데 있어서 원활한 상호작용을 위해 한 곳이나 한 사람이 프로세스를 상세하게 정의하도록 하고 있다.[2] 하지만 본 논문에서 정의하는 협동 워크플로우 모델은 실제 조직에서 액티비티를 수행하는 각 참여자가 자신이 수행하게 되는 액티비티를 스스로 정의하고 이렇게 정의된 액티비티를 이용하여 전체 글로벌 프로세스를 완성하게 되는 Fragment-driven의 워크플로우 프로세스 모델링 방법을 사용한다. 이런 방법의 모델링은 소수의 디자이너들에 의해서 글로벌 비즈니스 프로세스를 각 조직의 비즈니스

프로세스를 정의하는 것이 아니라 프로세스 정의에 참여하는 어떠한 그룹에 의해 액티비티를 먼저 정의하고 전체의 프로세스로 완성시켜나가는 Bottom-Up 방식이다. 따라서 각 롤(Role)의 그룹이 직접 프로세스 정의에 참여하는 것이 가능하게 된다. 이러한 방식을 취함으로써 프로세스 정의에 참여하는 그룹은 글로벌 프로세스 전체를 고려하지 않고 자신의 롤에 해당하는 액티비티만을 고려하면 되기 때문에 프로세스 정의가 쉬어질 수 있고 한 사람 또는 소수의 디자이너에 의한 글로벌 프로세스 정의가 아닌 실제 액티비티를 수행하게 되는 각 롤의 그룹이 참여하는 프로세스 정의 방법을 제시했다는 점에서 의미가 있다.

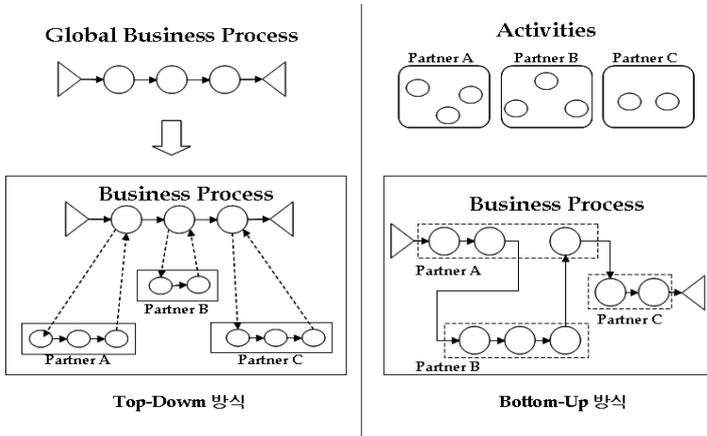


그림1. Top-Down 방식과 Bottom-Up 방식

이와 같은 액티비티로부터 시작되는 글로벌 프로세스의 정의를 각 롤에 기반하여 모델링하기 위하여 본 시스템에서는 스윘레인(Swimlane) 다이어그램을 이용한다.

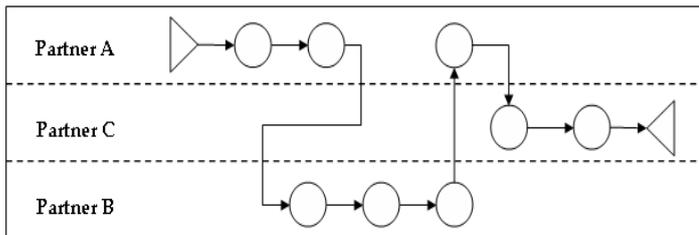


그림2. 스윘레인 다이어그램

그림 2를 보면 마치 수영 경기를 위한 수영장처럼 각 레인이 있고 각자 자신의 수영 레인에서 수영을 하듯이 프로세스에서 자신의 롤에 따라 레인을 가지게 된다. 프로세스의 단계나 작업은 각 레인에 표시하고 각 롤의 책임하에 있다. 화살표로 연결된 액티비티들이 프로세스의 흐름을 나타내고 있다.[3]

Fragment-driven의 워크플로우 모델링에서 스윘레인 다이어그램을 사용하는 이유는 첫째, 각각의 롤을 다이어그램 자체에 표현할 수 있기 때문이다. 둘째, 롤은 자신의 레인만 고려하여 액티비티를 정의하면 되기 때문에 롤에 기반하는 Fragment-driven 워크플로우 모델링에 적합하다. 셋째, 사전에 특별한 지식이 없는 사람도 한번에 이해할 수 있을 만큼 표현력이 뛰어나고 각각의 레

인(롤)과 다른 레인(롤)과의 상호작용을 잘 나타내고 있기 때문이다.

협동 워크플로우 모델에서 각 롤이 프로세스를 정의해 나갈 때에 고려해야 할 몇 가지 전제 사항이 있다. 우선 프로세스를 정의하기 전에 롤은 자신이 수행해야 될 액티비티를 알고 있어야 한다. 둘째, 각각의 롤은 자신의 레인 안에 있는 액티비티와 어떤 다른 액티비티와의 전달되는 관련데이터(Relevant-Data)를 알고 있어야 한다. 셋째, 각각의 롤은 레인 안에 있는 액티비티의 제어 흐름(Control-Flow)을 알고 있어야 한다. 넷째, 같은 프로세스 협동 모델링 도구를 사용한다. 다섯째, 모든 워크플로우 프로세스의 정의는 상호운영성을 위해서 상호교환이 가능한 XPD(XML Process Definition Language)로 저장되어야 한다.[1] 이러한 전제하에서 협동 워크플로우 모델은 다음과 같은 정책을 정의한다.

- 각 롤은 하나의 레인만을 가진다.
- 각 레인은 프로세스에서 각 롤이 수행하게 될 액티비티를 정의할 수 있다.
- 각 레인에는 해당 롤이 액티비티를 정의하는 것을 기본으로 한다. 필요할 경우에는 프로세스의 전체 관리자가 해당 레인의 액티비티를 정의하도록 지원한다.
- 각 레인의 액티비티의 속성들은 해당 롤만이 정의할 수 있으며(private), 각 액티비티 자체는 기본적으로 모든 액티비티에 보여진다(public).
- 각 액티비티의 Flow는 서로 관련되어 있는 액티비티 중에서 시작 액티비티가 자기 롤에 있을 때 연결이 가능하며, 필요한 경우 전체 비즈니스 프로세스 관리자가 정의할 수 있도록 한다.

위에서 제시한 정책에서 고려한 것과 같이 조직간의 비즈니스 프로세스를 정의를 할 때 각 조직과 프로세스 정의의 참여자들의 액티비티는 같은 프로세스 안에 있는 조직(또는 롤)이라면 볼 수는 있다. 하지만 조직의 독립성을 높이기 위해서 각 액티비티와 참여자, 애플리케이션, 관련 데이터등의 속성은 기본적으로 같은 조직(또는 롤)에서만 볼 수 있다. 이 정책은 협업을 위한 작업에서 조직 정보의 보안을 높이기 위한 방법이다. 이러한 정책과 개념들을 실제로 구현하기 위한 시스템의 구성요소는 그림 3과 같다.

협동 워크플로우 프로세스 정의 도구는 각 롤이 사용하는 정의 도구이다. 워크플로우 프로세스 정의 서비스(Workflow Process Definition Service)를 통하여 단일 조직에서의 프로세스 정의 및 다른 롤과 협업을 통한 조직간의 비즈니스 프로세스 정의를 지원한다. 협동 워크플로우 프로세스 정의 서비스는 협업 서버와 서비스에 접속하기 위한 최소한의 조직 스키마로 구성되며, 조직 스키마는 프로세스에 참여하는 조직에 대한 전체 정보를 가지는 것이 아니라, 서비스에 참여하기 위한 최소한의 아이디 및 패스워드, 해당 아이디가 속해 있는 롤에 대한 정보를 가진다. 협업 서버는 실시간 협업과 세션 관리를 지원하는 엔진과 세션 정보를 저장하는 데이터베이스로 구성된다.

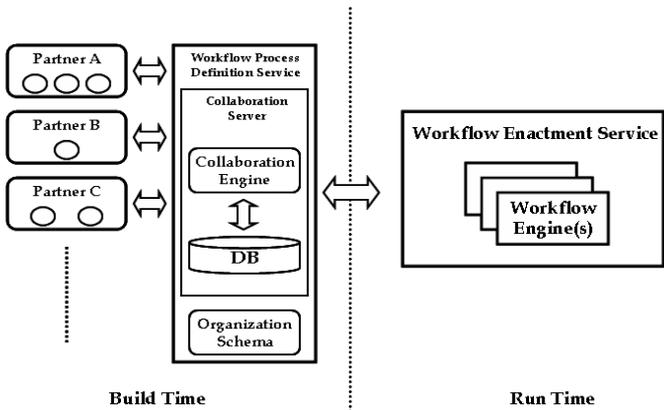


그림3. 협동 워크플로우 모델링 시스템의 구성요소

3. 협업 워크플로우 모델링 시스템

워크플로우 모델링 시스템은 크게 2부분으로 구성이 된다. 실제 모델링 작업을 하는 클라이언트와 클라이언트 간의 작업을 연계시켜주는 협업 서버이다.

3.1 모델링 클라이언트

모델링 클라이언트는 프로세스 정의를 위해 각각의 사용자에게 GUI(Graphic User Interface) 기반의 환경을 제공하는 워크플로우 프로세스 정의 모델링 툴이다. 워크플로우 프로세스를 정의하기 위한 방법으로는 여러 가지 표기법(Notation)이 있지만 그 중에서 사용자가 직관적이고 쉽게 프로세스를 정의하고 이해하기 위해서 ICN(Information Control Net) 기반의 표기법을 따른다. 또한 이기종의 시스템과 상호 운영성을 위해 WfMC의 XPD를 Import/Export 기능을 지원함으로써 프로세스 정의 상호 교환을 가능하게 한다. 또, 클라이언트는 협업 작업 시 네트워크의 트래픽(traffic) 및 오버헤드(Overhead)를 줄이고 효과적인 화면 처리를 위하여 이벤트 기반의 화면 처리 방식을 사용한다.

3.2 협업 서버

협업 서버의 주요 목적은 모델링 클라이언트로부터 받은 이벤트를 해당되는 클라이언트가 속해 있는 그룹에 이벤트를 전달함으로써 같은 그룹에 있는 클라이언트들과 협업을 할 수 있도록 도와준다. 협업 서버는 이벤트를 같은 그룹에 속해 있는 클라이언트들에게 서로 충돌이 나지 않게 순서대로 잘 전달될 수 있도록 이벤트의 트랜잭션을 보장해 주기 때문에 이벤트 순서화 서버라고도 불린다. 많은 수의 클라이언트들이 프로세스 정의 작업에 참여함에 따라 서버의 부하가 증가되고 이벤트의 전달 시간이 길어지게 되면 협업 모델링 작업이 원만히 진행되기 어렵다. 그러므로 실시간 협업이 이루어지기 위해서 서버는 최소한 이벤트의 전달 시간을 보장해줘야 한다. 이를 보장하기 위해서 협업 서버는 멀티서버, 분산 환경으로 이루어진다.

4. 협업 워크플로우 모델링 시스템 설계

4.1 이벤트(Event)

이벤트는 협업을 하기 위한 기본 통신 단위이며 실제 프로세스 디자인 작업과 밀접한 연관이 있기 때문에 중요한 의미를 가진다. 이벤트는 크게 모델링 화면에 보이는 그래픽컬한 부분을 담당하는 디자인 이벤트와 XPD와 관련된 부분을 담당하는 XPD 이벤트로 나뉜다. 기본적으로 두 이벤트 모두 사용자의 룰을 판별하기 위한 사용자 ID와 작업 세션(Session)을 나타내는 세션 ID를 가진다. 각각의 이벤트 설계는 다음과 같다.

(1) 디자인 이벤트

- ① Insert Event
 - 워크플로우 모델에 새로운 요소를 추가하는 이벤트
- ② Remove Event
 - 워크플로우 모델에 요소를 삭제하는 이벤트
- ③ Move Event
 - 사용자에 의해 모델에서 요소를 이동 시키는 이벤트
- ④ Pull Event
 - 사용자가 모델링 작업에 참여하기 위해 해당 세션에 접속하면, 다른 클라이언트에서 현재까지 정의된 모델을 동기화 하는 과정으로 협업 서버로부터 모델을 업데이트하는 이벤트
- ⑤ Chat Event
 - : 세션 내에서 사용자간의 의사소통을 하기 위한 이벤트
- ⑥ Update Event
 - 모델에 있는 요소의 정보를 갱신하는 이벤트

(2) XPD 이벤트

- ① Insert XPD Event
 - XPD에 새로운 요소를 추가하는 이벤트
- ② Remove XPD Event
 - XPD의 요소를 삭제하는 이벤트
- ③ Pull Event
 - 다른 클라이언트와 동기화를 위해 현재까지 진행되었던 XPD 요소를 삽입하는 이벤트
- ④ Update XPD Event
 - XPD의 요소의 정보를 갱신하는 이벤트

4.2 가상 협업 서버

가상 협업 서버는 Sun Microsystems사의 Java 가상 기계(Virtual Machine)에서 착안한 개념으로 각 클라이언트는 가상 협업 서버를 실제 협업 서버로 인식한다. 그림 4는 클라이언트에 존재하는 가상 협업 서버가 실제 협업 서버를 바라보는 그림이다.

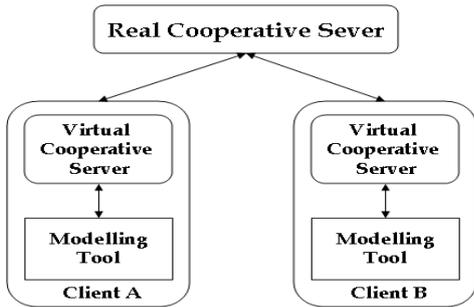


그림4. 실제 협업 서버를 바라보는 가상 협업 서버

이러한 구조는 모델링 클라이언트가 실제 협업 서버가 어디에 있는 알 필요가 없다는 것을 의미하며 협업 서버를 감춤으로써 협업 서버에 대한 투명성(Transparency)을 보장해준다. 또한 가상 협업 서버 구조는 단순한 서버/클라이언트 구조에 그치지 않고 가상 협업 서버가 어디를 향해 있는지에 따라 기업 내 조그만 팀부터 부서, 기업 또는 그 이상의 계층 구조를 형성화 시킬 수 있는 유연성을 가진다.

4.3 협업 모델링 시스템 아키텍처

가상 협업 서버와 모델링 클라이언트로 구성된 협업 모델링 시스템은 EJB 기반의 통신 인프라를 통해 협업 서버와 이벤트를 주고받으며 통신을 한다. 그림 5는 워크플로우 모델링 시스템 전체 아키텍처이다.

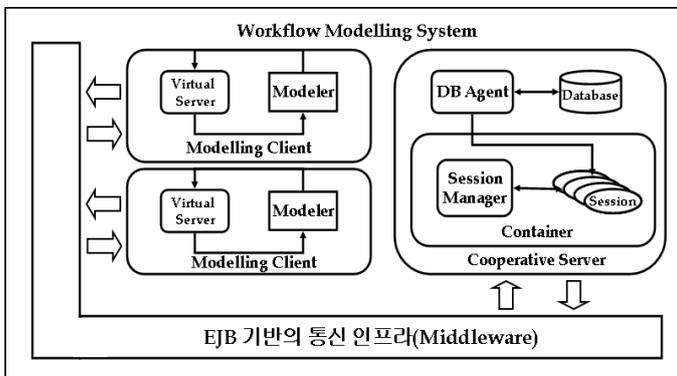


그림5. 워크플로우 모델링 시스템 아키텍처

5. 협업 워크플로우 모델링 시스템 구현

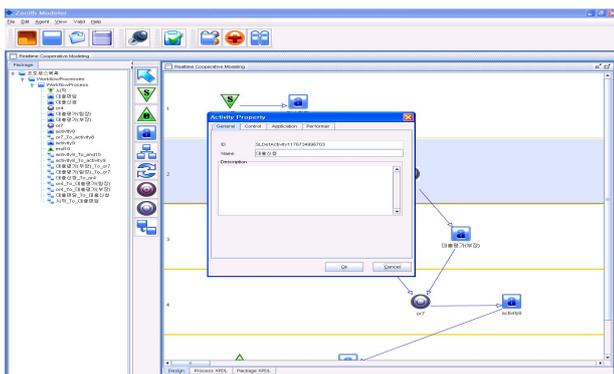


그림6. 모델링 클라이언트의 모델링 화면

그림6은 각 레인에 해당되는 사용자 5명이 동시에 모델링 클라이언트를 통해서 워크플로우 프로세스 정의를 하고 있다. 그 중에서 한 명의 사용자가 액티비티의 정보를 설정하는 모델링 클라이언트의 모습이다. 그림의 좌측에는 워크플로우 모델 정의를 계층적으로 보여주는 트리가 있으며 가운데 세로로 놓여진 톨바는 액티비티를 비롯한 모델링 디자인에 사용되는 요소들로 이 중하나를 선택하여 우측에 보이는 화면 보이는 레인 중에서 자신의 해당되는 레인에 디자인 할 수 있다. 자신의 해당되는 레인의 액티비티만 속성을 정의할 수 있으며 액티비티와 액티비티간의 트랜지션의 연결은 트랜지션의 시작 액티비티가 자신의 레인에 있을 때 연결이 가능하다.

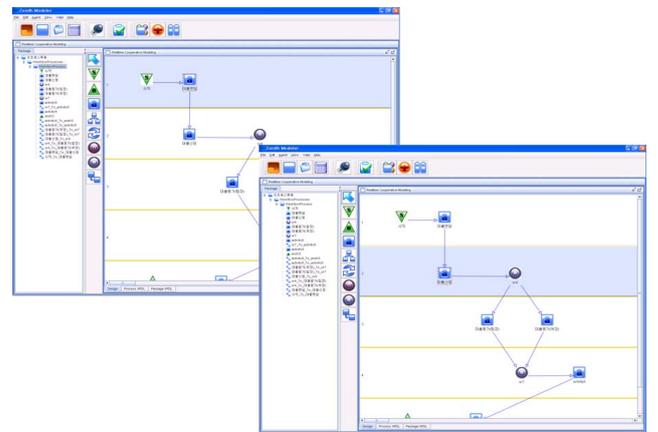


그림7. 모델링 클라이언트간 협업 화면

그림 7은 5명의 사용자 중에서 2개의 모델링 클라이언트 화면을 보여주고 있다. 위에 그림과 같이 모든 모델링 클라이언트의 화면이 같은 것을 알 수 있다. 따라서 사용자는 작업에 참여중인 그룹의 멤버들이 실제로 멀리 떨어져 있어도 바로 옆에서 작업하는 것처럼 느껴진다.

6. 결론

본 논문에서는 다양하고 복잡한 기업의 프로세스를 보다 효율적으로 정의하기 위해 워크플로우 협업 모델링 시스템의 아키텍처 및 워크플로우 모델을 제시하였으며 시스템을 설계 및 구현하였다. 기존의 워크플로우 모델과는 다르게 역할 기반의 모델을 기반으로 하였으며, 사용자에게 보다 쉬운 워크플로우 모델링 환경을 제공하기 위해 ICN 노테이션을 따르는 비주얼한 모델링 클라이언트를 구현하였으며 클라이언트 통신간의 네트워크 트래픽을 줄일 수 있는 이벤트 방식의 화면 처리와 향후 분산 환경으로 전환과 확장을 위한 투명성 보장을 제공하는 가상 협업 서버를 포함하고 있다. 그리고 협업 서버는 각 클라이언트들이 원활히 협업 할 수 있도록 실시간 이벤트 전달을 보장하며 이벤트의 트랜잭션을 관리한다. 그러므로 본 논문에서 구현한 모델링 시스템을 사용하면 복잡한 기업의 워크플로우 프로세스를 보다 효율적으로 정의할 수 있다.

7. 참 고 문 헌

- [1] 원재강 외 3명, “역할 기반 워크플로우 모델”, 한국정보처리학회 추계학술발표논문집, 2000.10
- [2] Harou Hayami, Masashi Katsumata, Ken-ichi Okada, "Interworkflow : A Challenge for Business-to-Business Electronic Commerce", Workflow Handbook 2001, 2002
- [3] Alec Sharp, Patric McDermott, "Workflow modeling tools for process improvement and application development", Artech House Boston London 2001
- [4] 오동근 외 6명, “전자물류 워크플로우 모델링 시스템”, 한국정보과학회 가을학술발표논문집(1) 29권 2호, pp154-156, 2002.10