

실시간 협업 지원 협동 스웜레인 워크플로우 모델링 시스템 아키텍처

이정훈^o 오동근 한승혁 김형욱 원재강 김민홍 김광훈
경기대학교 전자계산학과 워크플로우 연구실
leejunghoon@kyonggi.ac.kr

A Cooperative Swimlane Workflow Modeling System Architecture

Jung-Hoon Lee^o Dong-Keun Oh Seung-Hyuk Han
Hyeong-Mok Kim Jae-Kang Won Min-Hong Kim 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 Workflow 모델링 방식은 프로세스 정의를 한 명(또는 소수)의 디자이너가 맡아 왔다. 그러나 스웜레인 같은 롤 기반의 프로세스 모델링은 각 롤에 해당하는 디자이너 그룹이 프로세스를 정의하기 때문에 모델링 작업을 단순화시킬 수 있다. 본 논문에서는 워크플로우 모델링 방식과 다른 역할 기반의 스웜레인 모델링을 바탕으로 그룹웨어기술을 접목시켜 기업의 업무 프로세스를 보다 효율적으로 정의할 수 있는 실시간 협업 지원 협동 스웜레인 워크플로우 모델링 시스템 아키텍처를 제시한다. 이벤트 방식의 화면 처리는 네트워크 트래픽 및 오버 헤드를 감소시킬 수 있다. 또한 분리된 어플리케이션이 비동기적으로 신뢰성 있게 통신할 수 있도록 메시징 시스템을 도입하고, 협업간 투명성(Transparency)을 보장하는 가상 협업 서버 구조를 살펴 볼 것이다. 이러한 것들은 향후 분산환경으로의 발전을 이룰 수 있다.

1. 서론

최근의 전자적인 작업환경(Electronic Workplace)과 단위업무의 변화에 보다 효율적으로 조직을 운영하고 대처하기 위한 기술이 바로 워크플로우라 할 수 있다. 워크플로우는 일반적으로 전체적 혹은 부분적으로 비즈니스 프로세스를 컴퓨터화하여 편리하게 하거나 자동화 한 것으로 정의되며 이러한 워크플로우에 관련된 기술들은 컴퓨터 및 통신 분야뿐만 아니라 사회학 분야나 언어학분야, 경영학 분야 등의 다각적인 협력 관계를 통해서만 성공적으로 완성될 수 있는 매우 다중적인 분야라고 할 수 있다. [1] 워크플로우 기술은 워크플로우 모델 분야와 워크플로우의 각 업무들을 실행하고 그들 간의 업무 흐름을 제어하는 워크플로우 관리 시스템 분야로 나누어질 수 있다. 특히 워크플로우 관리 시스템이 새로운 기술이나 조직내의 작업환경에 효과적으로 적용되기 위해서 사전에 워크플로우 모델이 잘 정의되어 있어야 한다. 기존의 Workflow 모델링은 전체 프로세스를 디자인 한 후 그에게 속하는 서브 프로세스를 디자인 하는 액티비티 기반의 Top-Down 접근 방식으로 볼 수 있으며 일반적으로 한 사람이 디자인 하는 형태였다. 반면에 역할 기반의 워크플로우 모델인 스웜레인은 디자이너마다 고유의 롤을 가지고서 롤을 먼저 정의하고(Fragment-Driven) 관리자격의 디자이너가 프로세스를 연결하는 Bottom-Up 접근 방식을 취한다.

오늘날과 같이 빠르게 발전하는 기업의 구조와 복잡하고 다양하게 변하는 기업의 규모에서 모든 업무 프로세스들을 한 사람이 모델링하고 관리하기가 점점 어려워지고 있다. 따라서 이러한 프로세스 모델링 작업을 여러 사람이 실시간으로 함께 진행할 수 있다면 효과적이고 안정적인 결과를 얻을 수 있을 것이다. 본 논문에서는 조직의 업무 프로세스를 보다 효율적으로 설계할 수 있는 실시간 협업 지원 협동 스웜레인 워크플로우 모델링 시스템 아키텍처를 제안한다. 2 장에서는 실시간 협업 지원 시 나타날 수 있는 문제점들을 고려해 보고 3 장에서는 시스템 아키텍처를 제시하며 4장에서는 협업 시나리오를 5장에서는 결론 및 향후 과제를 기술한다.

2. 실시간 협업 지원 시 고려사항

2.1 Session Control

Session 제어란 그룹의 구성원들이 그룹웨어 시스템을 통해 그룹 활동에의 참여를 제어 받는 것을 의미한다. Session 제어는 협동작업 시 작업자의 참여여부, 권한 부여 등의 몇 가지 기능을 제공해야 한다. 즉, 참여자들이 합류하고 헤어지는 것을 어느 시간이나 가능해야 하고, 회의에 방해가 되는 간섭이나 중단 등을 막는 수단, 서로를 알아볼 수 있도록 하는 기능들이 제공되어야 한다.

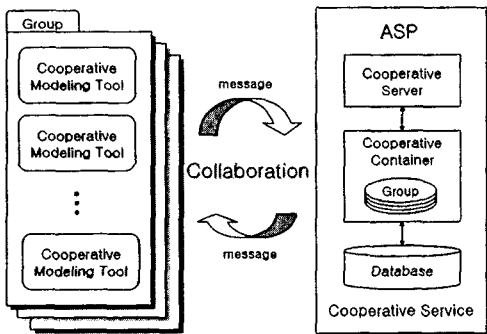
2.2 Floor Control

Floor 제어란 Session 제어물 통해 그룹 활동의 접근을 허용 받는 뒤에 그룹 활동에서의 주제를 결정하는 제어 기능이다. 참여자들이 공유해야 하는 자료에 대하여 다양한 종류의 권한을 제공하여야 한다. 모든 사람들이 모든 것들에 대하여 동시에 권한을 가질 수 있도록 하는 것이 가장 바람직한 방법이다. 하지만 이러한 방법은 협력관계의 일을 하지 않는 소수 개인에게 피해가 될 수 있다. 그래서 어느 정도의 중재된 권한을 제공하는 것이 전체 회의에 있어서 실수를 막고, 허가되지 않는 침입을 차단하며, 사람들이 다투는 상황을 막기 위한 수단이 제공해야 한다.

2.3 Synchronization

실시간 협업 지원 시스템에서 또 한 가지 짚고 넘어가야 할 문제점으로 동기화 문제가 있다. 일반적으로 동기화를 시스템 차원에서 제공해 주지 못하면 작업간의 충돌에 의해 이상현상이 발생하여 협업이 갖는 효용성을 저해할 뿐만 아니라 최악의 경우에는 작업을 더 이상 진행할 수 없는 상황에 이를 수 있다. 하지만 스웸레이온은 풀 기반의 워크플로우 모델링 시스템이기 때문에 유저들의 작업을 자신의 롤로 한정지움으로서 충돌을 피할 수 있다. 이런 시스템 특성으로 인해 비 동기적 방식의 통신이 가능하다.

3. 아키텍처



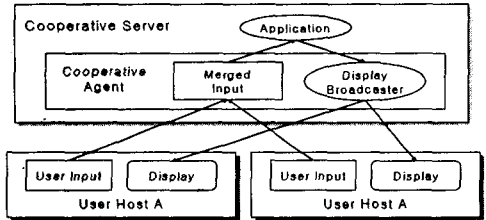
<그림 1> 실시간 협업 시스템 아키텍처

협업 시스템에서 중요한 것은 유저들간의 작업을 어떻게 인터페이스화 하느냐 이다. 세 가지 경우를 생각해 볼 수 있는데 첫 번째 경우는 협업 하고자 하는 클라이언트들이 서로 모여 자신들만의 협업 인터페이스를 만들고 그에 따라 협업을 할 수 있으며, 두 번째는 서비스 형태로 각 클라이언트들을 연결 시켜주는 ASP(Application Service Provider)가 있고, 세 번째는 표준화된 협업 인터페이스를 바탕으로 협업을 할 수 있다. 첫 번째 경우는 조직 내에서의 협업은 용이하나 조직간 협업을 위해서 너무 많은 인터페이스가 필요하다. 두 번째 경우는 ASP가 제공하는 모델링 툴과 서비스를 이용해 조직간은 물론 조직 내에서도 협업을 할 수 있다는 장점이 있으나, ASP에 의존한다는 단점과 약간의 추가적인 비용을 지불해야 한다

는 것이다. 세 번째 경우는 어떤 의존도 없으면서 조직간 협업이 가능하지만, 아직 표준화된 인터페이스가 없고, 각 조직이 표준화된 협업 시스템을 갖추고 있어야 하므로 조직의 부담이 크며 협업 중 생기는 재난에 대한 책임을 모두가 부담 해야 한다. 따라서 본 논문에서는 ASP의 단점에도 불구하고 모든 협업이 가능하면서 조직에게 큰 부담이 되지 않는 ASP 방식을 선택한다. 그림 1 에서와 같이 모델링 작업을 하는 클라이언트들이 있고 이들에게 협업 서비스를 제공하는 ASP가 있다. 클라이언트들은 모델링 작업에 따라 의미적으로 일련의 그룹을 형성하게 되며 클라이언트들은 그런 그룹을 신경 쓰지 않고 ASP의 협업 서버와 통신을 하게 된다. 협업 서버는 전달 받은 클라이언트 작업을 ASP의 협업 컨테이너에게 전달하고 협업 컨테이너는 협업에 필요한 최소한의 처리와 협업에 의한 결과물인 워크플로우 모델링 정의를 DB에 저장한다. 클라이언트들은 협업 컨테이너에서 그룹별로 관리되며 작업 그룹의 나머지 클라이언트들에게 작업 내용을 브로드캐스팅한다.

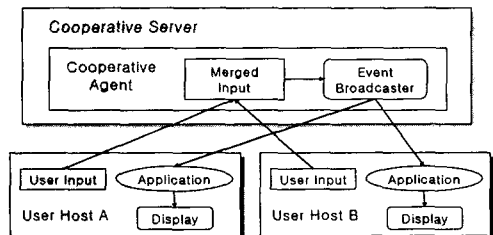
3.1 화면 처리 방식

클라이언트의 화면 처리 방식에는 디스플레이 방식과 이벤트 처리 방식이 있다.



<그림 2> 디스플레이 방식

그림 2는 디스플레이 처리 방식으로서 한 클라이언트에서 그린 액티비티가 그대로 스크린 램처 되어 다른 클라이언트들에게 전송 된다. 따라서 모든 클라이언트가 똑같은 그림을 볼 수 있다는 장점이 있지만 화면의 작은 부분에 대한 수정이 발생했을 때에도 화면상의 모든 내용을 전달해야 하는 이유 때문에 상당한 오버 헤드가 발생하며 그로 인해 네트워크의 트래픽을 가중시킨다. 그리고 각 클라이언트마다 해상도가 다르다면 화면을 보낸 클라이언트와 받은 클라이언트 사이에서 변환이 이루어져야 할 것이며, 협업을 지원하는 모든 기능을 서버 측에서 담당하기 때문에 병행처리의 어려움이 있다.



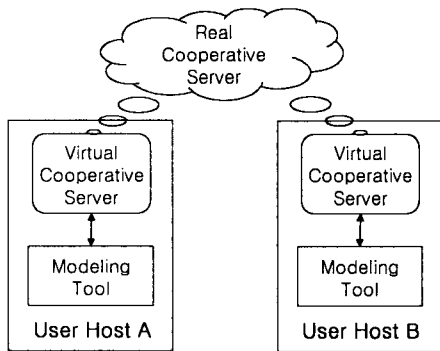
<그림 3> 이벤트 방식

그림 3은 이벤트 방식의 처리 구조를 나타내고 있다. 이벤트 처리 방식은 클라이언트가 그린 액티비티에 대한 이벤트가 전송되기 때문에 오버 헤드가 적으므로 많은 수의 클라이언트들간 협업이 이루어질 수 있다. 또한 비 동기적으로 이벤트를 처리할 함으로서 병렬 처리가 가능해지며 각 클라이언트들 사이의 해상도를 고려할 필요 없이 이벤트를 받은 클라이언트가 디스플레이 하는 시점에서 자신의 해상도에 맞게 변환 할 수 있다.

3.2 메시지 방식의 통신

분산 어플리케이션이 급격히 증가함에 따라 이전에 발생하지 않았던 동기화, 안정성, 확장성 그리고 보안 등에서 문제가 발생하기 시작하였다. 이에 대한 한가지 해결책은 메시지를 통해 각 컴포넌트 사이의 결합성(coupling)을 약화시키는 메시징 시스템이다. 또한 메시징 시스템은 분리된 어플리케이션이 비 동기적으로 신뢰성 있게 통신할 수 있도록 해 준다. 이런 메시징 시스템의 장점을 모든 협업 시스템에서 받아 들일 수 있는 것은 아니다. 일반적으로 협업 시스템은 동기화 메커니즘에 의해 운영되나 2.3 에서 살펴본 바와 같이 본 시스템 특성상 비 동기적 방식의 통신이 가능하다고 하였다. 따라서 협업을 위한 통신 수단으로 비 동기적인 메시지 방식을 채택 함으로서 동적이고, 신뢰성 있고 유연한 시스템을 구현할 수 있으며 그에 따라 시스템의 나머지 부분에 영향을 주지 않고 전체적인 구성을 변경할 수 있다.

3.3 가상 협업 서버 (Virtual Cooperative Server)



<그림 4> 실제 협업 서버를 바라보는 가상 협업 서버

가상 협업 서버는 Sun Microsystems사의 Java 가상 기계(Virtual Machine)에서 착안한 개념으로 각 클라이언트에게는 가상의 협업 서버가 존재하며 클라이언트는 가상 협업 서버를 실제 협업 서버인양 바라본다. 이러한 구조는 클라이언트가 실제 협업 서버가 어디에 있든, 무엇이든 알 필요가 없다는 것을 의미하며 협업 서버를 감춤으로써 협업 서버에 대한 투명성(Transparency)을 보장해 준다. 또한 가상 협업 서버 구조는 단순한 서버/클라이언트 구조에 그치지 않고 가상 협업 서버가 어디를 가리키는냐에 따라 기업 내 조그마한 팀으로부터 부서, 기업 또는 그 이상의 계층 구조로 형성화 시킬 수 있는 유연성도

가진다. 가상 협업 서버와 실제 협업 서버는 내부적으로 메시지 기반의 통신을 수행 함으로서 메시징 시스템의 장점을 살려 분산환경으로의 발판을 만들 수 있고 향후 확장 의 여지를 남긴다. 그림 4는 클라이언트에 존재하는 가상 협업 서버가 실제 협업 서버를 바라보는 모습이다.

4. 시나리오.

워크플로우 정의과정을 간단히 개념적 시나리오를 통해 설명한다. 클라이언트들(프로세스 디자이너 그룹)은 프로세스 정의를 위해 세션에 참가하며, 참가 순서대로 레인을 할당 받는다. 작업도중 클라이언트(A)가 액티비티를 그림과 동시에 협업 서버에게 이벤트를 보내며 협업 서버는 클라이언트(A)로부터 받은 이벤트를 세션에 참여중인 그룹에게 브로드캐스팅한다. 협업 서버로부터 이벤트를 전달 받은 클라이언트들은 자신의 View에 반영한다.

5. 결론 및 향후 연구 과제

본 논문에서는 복잡한 기업의 프로세스를 보다 효율적으로 정의하기 위한 여러 가지 방법 중 그룹웨어 기술을 접목 시켜 협업 워크플로우 모델링 시스템 아키텍처를 제시했다. 기존의 워크플로우 모델은 현재와 같은 작업환경(협업 시스템, 협동 그룹웨어 등)에는 적당치 못 하기 때문에 역할 기반 모델인 스웬레인을 기반으로 하였다. 협업을 위한 구조로서 여러 가지 구조가 있지만 모든 협업이 가능하고 조직에게 큰 부담이 없는 ASP방식을 선택하고 통신간 네트워크의 트래픽을 줄일 수 있는 이벤트 방식의 화면 처리와 메시지 기반의 통신을 수행함으로써 비 동기적이고 신뢰성 있는 통신을 할 수 있다. 또한 가상 협업 서버는 향후 분산 환경으로의 전환과 확장을 위한 투명성을 제공할 수 있다. 향후 발전 과제로는 제시한 시스템 아키텍처를 기반으로 실제 구현 적용해 보는 것이며 향후 연구과제로는 암호화를 통한 정보보호에 관한 추가적인 연구가 요구가 요구되며, ASP간 협업으로의 확장이 필요하다.

Acknowledgement

이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2002-003-D00247)

참고 문헌

[1] 원재강 외 3명, "역할 기반 워크플로우 모델", 한국정보처리학회 추계학술발표논문집, 2000.10
 [2] Roger S. Pressman, "Software Engineering, A Practitioner's Approach", 3rd Ed, McGraw Hill, 1997
 [3] 박웅, 김상배, 오종태, 김광훈, 백수기, "협동작업을 지원하는 워크플로우 모델링 도구의 설계", 한국인터넷정보학회 춘계학술발표논문집, 2000.06
 [4] 김상배, 배성용, 김광훈, 백수기, "실시간 협업지원 그룹 워크플로우 모델링 도구", 한국정보처리학회 추계학술발표논문집, 2000.10