

워크플로우 실시간 협동 액티비티 지원 프레임워크

이기원[○] 안형진 박민재 김광훈 백수기
경기대학교 전자계산학과 워크플로우 기술 연구실
{kwlee[○], hjahn, mean222, kwang, skpaik}@kyonggi.ac.kr

Realtime Cooperative Activity Supports in Workflow

Kiwon Lee[○] Hyungjin Ahn, Meanjea Park, Kwanghoon Kim, Suki Paik
Dept. of Computer Science, Kyonggi University

요 약

현재의 워크플로우 관리 시스템에서 프로세스를 이루는 각 액티비티의 작업은 단일의 수행자에 의해서 처리되어 진다. 하지만 실제 기업의 프로세스나 과학 분야의 프로세스에서는 단일의 수행자가 아닌 여러 사람들의 협동을 통해 해결해야 하는 작업이나 실시간으로 데이터를 주고받으며 업무를 처리해야 하는 경우가 많다. 이런 작업들을 필요로 하는 곳에 워크플로우 관리 시스템을 도입하기 위해서는 실시간 협동 액티비티의 지원이 필요하다. 하지만 현재의 워크플로우의 표준 스펙에 협동 액티비티의 개념이 확실하지 않고 이에 대한 내용이 정의되어 있지 않기 때문에 실시간 협동 액티비티의 처리가 불분명한 실정이다. 본 연구에서는 표준의 스펙을 확장하여 실시간 협동 액티비티를 지원하도록 기존의 워크플로우 관리 시스템을 확장 설계한다. 워크플로우 관리 시스템을 세부적으로 나누어 워크플로우 모델관점, 워크플로우 엔진 관점, 런타임 클라이언트의 관점으로 크게 세 부분으로 나누고 각 부분에서 실시간 협동 액티비티를 지원하기 위한 요구사항들과 그것의 처리 방안에 대해 제안한다.

1. 서 론

워크플로우는 조직 내, 조직 간에 작업을 처리하는 비즈니스 프로세스를 컴퓨터 기반에서 자동화하여 처리하는 것이다. 분산된 작업 환경 하에서 조직들이 작업을 처리할 수 있도록 협업과 작업 모니터링, 그리고 다양한 작업들의 수행을 지원하기 위하여 개발된 시스템이 워크플로우 관리 시스템이다. 이런 워크플로우 관리 시스템을 사용함으로써 정부나 기업과 같은 조직은 작업 효율의 증대와 비용의 감소라는 이점을 얻을 수 있다. 하지만 기업의 프로세스는 매우 복잡하고 다양하기 때문에 기업에 적합한 프로세스를 찾아서 정의하는 것은 매우 어렵다. 또한 실제의 비즈니스 프로세스를 비즈니스 프로세스 언어나 비즈니스 프로세스 모델링 표기법으로 표현하는 것에도 한계가 존재하게 된다. 이런 한계를 극복하기 위해서 여러 단체에서 비즈니스 프로세스 언어와 비즈니스 프로세스 모델링 표기법의 표준을 제정하고 그에 대한 연구를 계속적으로 진행해 나가고 있다. 그 중 워크플로우의 표준 단체인 WFMC(Workflow Management Coalition)에서는 비즈니스 프로세스 정의 언어인 XPDL(XML Process Definition Language)을 통해 더 효과적인 비즈니스 프로세스 표현 방법을 연구하고 있다.

현재 워크플로우 프로세스를 구성하는 액티비티 중 시스템(SYSTEM)에 의해 처리되지 않고, 사람에게 의해 처리되는 액티비티는 어떤 역할(ROLE)에 해당하는 한 수행자나 특정한 사람(HUMAN)에 의해서 처리되어 진다. 이것은 그룹이나 협동 처리를 위한 지원이 이루어 지지 않고 있다는 것이다. 하지만 여러 프로세스에서는 협동 작업의 수행을 필요로 하고 있다. 예를 들어 실제 기업의 비즈니스 프로세스에서 회의나 토론을 통한 실시간

업무처리, 여러 사람이 동시에 하나의 모델링 서버에 접속하여 역할에 따른 모델링을 통해 전체의 모델을 이끌어 내는 스웬레인 모델링 등이 있다. 또한 자연 과학(scientific)분야에 워크플로우를 접목시켜 워크플로우에서의 하나의 이슈가 되어 가고 있는 과학적 워크플로우(Scientific Workflow)에서도 협동 액티비티의 처리가 요구되고 있다. 실험을 통해 발생하는 데이터를 관리하고 처리하는 일이 주된 업무인 과학적 워크플로우에서는 데이터의 수집과 처리와 검증이 동시에 일어나야 하고, 동적인 환경에서 여러 수행자들이 데이터가 공유하는 환경을 필요로 하기 때문에 더욱 실시간의 협동 액티비티의 처리를 필요로 하고 있다.[1]

실시간으로 여러 사람이 처리하는 액티비티를 지원하기 위해 현재의 워크플로우 관리 시스템은 효율적이지 않다. 이에 많은 논문에서 이와 같은 방향이 연구되어지고 있다. 그룹웨어 시스템의 부분, CSCW(Computer Support Cooperative Work) 부분, 협동 워크플로우 관리 시스템 부분에서 이와 같은 사항들의 연구가 지속되어 가고 있다.

본 논문에서는 기존의 전통적인 워크플로우 관리 시스템을 확장하여 실시간 협동 액티비티를 지원하도록 설계한다. 워크플로우 관리 시스템을 이루는 각 부분을 프로세스 모델, 워크플로우 엔진, 런타임 클라이언트로 세분화하고 각 부분에서 실시간 협동 액티비티를 지원하는 방법을 제안한다.

워크플로우 모델관정에서는 협동 액티비티의 그래픽 모델을 정의하고, 액티비티에 할당되는 다수의 수행자를 위해 수행자들의 역할(ROLE)을 확장하여 Work Group을 생성한다. 워크플로우 엔진 관정에서는 워크아이템의 할당 메커니즘을 확장하고, 액티비티 인스턴스의 생성과

소멸시점을 재정의 한다. 그리고 실시간 작업을 위한 협동 서버를 두어 세션을 관리하고 이를 통해 실시간 작업을 수행한다. 런타임 클라이언트 관점에서는 워크플로우 엔진으로의 액티비티 완료 상태에 대한 처리 부분을 확장한다.

2장에서는 기존의 협동 액티비티에 대한 연구에 대한 관련연구를 기술하고, 3장에서는 각 부분의 확장 방안에 대해 제안한다.

2. 관련 연구

현재의 워크플로우 관리 시스템에서 각 액티비티의 수행은 단일의 수행자에 의해 처리되어지고 있다. [2]에서는 협동 작업을 지원하는 워크플로우 모델을 제안하였다. 전통적인 조직 모델에 그룹의 개념을 확장하여 워크플로우 메타 모델을 정의하고 협업 가능한 워크플로우 시스템을 구성하였다. [2]에서는 워크플로우 모델에 그룹(Group), 협업 오브젝트(Cooperative Object), 협동 도구들(Cooperative Tools)의 세 개의 새로운 요소들을 추가함으로써 협동을 지원하였다.

기존의 워크플로우 시스템은 워크아이템이 여러 사람 자원에 제공 되더라도 워크아이템은 단일의 사람 자원에 의해 실행되어진다는 단점을 들고 그 단점을 해결하기 위해서 그룹의 개념을 정의하였다. 그룹은 사람 자원들의 모임을 말한다. 그룹은 여러 사람 자원을 가질 수 있고 하나의 사람 자원은 여러 그룹의 멤버로 속하는 것이 가능하다. 이런 그룹을 통해 액티비티를 처리함으로써 협동 작업의 지원이 가능해 지게 된다.

협업 오브젝트(Collaborative Object)는 그룹에 속한 사람들 사이에서 전송되어지는 데이터를 의미한다. 협동 도구는 워크플로우 시스템 밖에서의 협업 애플리케이션 도구들을 의미한다. 이것들은 워크플로우 엔진에 의해서 호출되고 그룹에 속한 사람들 간에 강력한 통신을 지원하게 된다. 워크플로우의 실행에 있어서 협업 오브젝트는 데이터베이스에 저장되어야 하고 그룹에 속한 사용자들은 협동 도구들을 사용함으로써 협동 액티비티의 실행이 가능해진다.

[3]에서는 COO(Cooperative Transaction Protocol)와 현재의 워크플로우 모델을 결합하여 협동 액티비티의 실시간 데이터 교환을 용이하게 하여 협동 액티비티를 더욱 효과적으로 수행할 수 있도록 연구 했다. 이처럼 기존의 협동 액티비티에 대한 연구는 협동 액티비티가 할당되는 그룹 조직 모델 확장에 대한 연구와 협동 액티비티를 구성하는 그룹 내의 수행자들 간의 업무의 처리를 위한 실시간 데이터 교환에 대한 연구가 주를 이루었다. 본 논문에서는 이 두 가지 관점을 모두 고려하여 시스템의 확장을 제안한다.

3. 실시간 협동 액티비티 지원 프레임워크

워크플로우 관리 시스템은 다섯 부분으로 구성되어 있다. 먼저 비즈니스 프로세스를 프로세스 정의 언어로 표현하는 프로세스 정의 도구, 비즈니스 프로세스의 실질적인 처리를 담당하는 워크플로우 엔진, 작업 수행자가 업무를 확인하고 실행하는 도구인 런타임 클라이언트, 작업을 수행하는 도구인 애플리케이션, 관리자의 업무를

수행하고 프로세스 진행 상태를 모니터링 하는 관리자/모니터링 도구로 구성되어 있다.

본 장에서는 워크플로우 관리 시스템이 실시간 협동 액티비티를 지원하기 위해서 필요한 기능들을 기존의 워크플로우 관리 시스템에 확장함으로써 실시간 협동 액티비티의 지원이 가능하도록 설계 한다.

3.1 모델 (Model) 확장

모델 확장은 프로세스를 언어로 기술하는 프로세스 정의 언어에 대한 확장을 의미한다. 프로세스 정의에 대한 표준은 Workflow Process Definition Interface 에 정의 되어 있고 XPDL을 기반으로 확장하였다.[4] 모델의 확장에서는 그래픽적인 협동 액티비티의 지원과 ROLE의 확장을 제안 한다.

3.1.1 협동 액티비티

일반적으로 프로세스를 모델링 하는 사용자는 프로세스를 모델링 할 때 프로세스 정의 언어를 사용하여 직접 정의하는 것이 아닌 그래픽적인 모델링 도구를 사용해서 모델링 작업을 수행한다. 이 모델링 도구는 그래픽적인 모델을 프로세스 정의 표준에 따라 프로세스 정의 언어(XPDL)로 해석하여 준다. 이를 위해서는 그래픽적인 모델링을 위한 협동 액티비티의 표현이 필요로 하게 된다. 다음의 표 1은 확장하여 정의한 협동 액티비티와 기존의 액티비티들의 그래픽 적인 표현을 나타내고 있다.

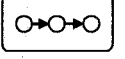
Generic-Activity	○	
Route-Activity	○	XOR
	●	AND
Block-Activity		
Subflow-Activity	Ⓢ	
Cooperative-Activity	Ⓒ	

표 1 액티비티의 그래픽 적 표현

협동 액티비티는 크게 두 가지로 구분 지을 수 있다. 하나는 작업의 수행을 실시간으로 처리해야 하는 액티비티이다. 한 업무를 수행함에 있어서 그 업무를 처리하는 모든 작업 인원들이 동시에 접속해서 함께 토론하고 데이터를 공유하면서 실시간으로 일을 처리하는 것이다. 이런 액티비티의 처리는 같은 시간에 모든 작업 인원들이 접속해서 함께 일을 처리해야 한다. 예를 들어 회의나 토론을 통해 수행되는 작업의 처리, 실시간 협업 모델링 등이 있다. 그리고 다른 하나는 작업의 수행이 미리 분담되어 실시간으로 동시에 처리하지 않아도 되는 액티비티이다. 한 업무를 수행함에 있어서 업무를 처리

하는 그룹 인원들의 업무가 각각 독립적으로 나누어져 있어 자신의 업무만 수행하면 되는 것이다. 본 논문에서는 실시간으로 수행되어지는 협동 액티비티에 대해 연구한다.

3.1.2 역할(ROLE) 확장

현재의 워크플로우 프로세스 모델에서 액티비티에 할당되는 자원으로는 ROLE, HUMAN, 그리고 SYSTEM이 있다. ROLE은 공통의 목적을 지닌 HUMAN들의 집합으로서 액티비티에 ROLE이 할당되면 그 ROLE을 가진 그룹에서 한 HUMAN이 일을 처리하게 된다. HUMAN은 어떤 직접적인 한 개인을 의미한다. 그리고 SYSTEM은 사람에게 의해서 처리되는 것이 아닌 시스템, 컴퓨터에 의해 처리되는 것을 의미한다.[4] 현재의 워크플로우 시스템에서 액티비티의 처리는 한 개인에 의해서만 처리가 가능하다. 그렇기 때문에 협동 액티비티를 지원하기 위해서는 수행자에 대한 확장, 특히 ROLE에 대한 확장이 필요하다.

역할(ROLE)에 대한 확장은 ROLE Agent를 통해 가능하다. ROLE Agent는 사용자에게 할당되어 있는 기존의 역할을, 조직의 구조와 연동하여 직책 또는 역할에 따라 사용자에게 새로운 등급 또는 역할을 부여한다. 이를 통해서 협동 액티비티를 수행하는 수행자들은 등급에 따른 새로운 역할을 가지게 된다.

그림 2 에서 Actor a 는 A부서의 A-1 이라는 역할을 가지고 있다. 그리고 Actor c 는 A부서의 A-2 라는 역할을 가진다. 이때 ROLE Agent 는 이런 기존의 역할에 추가적인 역할을 추가시킨다. 그리하여 미리 정의된 WorkGroup A에 Actor a를 배치하고 WorkGroup C에 Actor c를 배치한다. 이때 기존의 역할은 가지고 있으면서 동시에 WorkGroup에도 포함하는 것이다. 이런 Role Agent를 사용함으로써 동적으로 WorkGroup에 수행자 할당이 가능해 지고 WorkGroup 배치의 정책을 사용자가 직접 정해서 자유로운 그룹의 생성이 가능하게 된다. 이를 통해 생성된 WorkGroup이 협동 액티비티에 할당되어 다수의 수행자에 의해 액티비티의 수행이 이루어지게 된다.

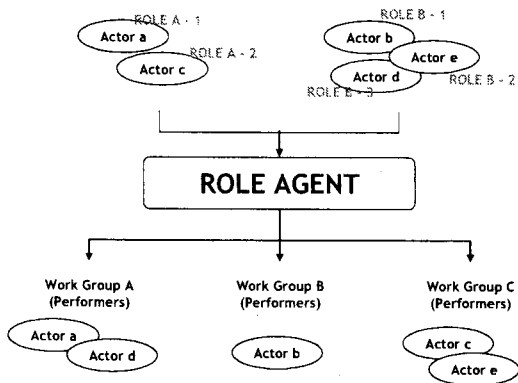
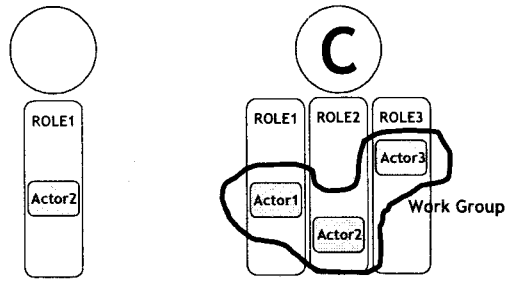


그림 2 Role Agent를 통한 ROLE 확장



A) General Activity Role B) Cooperative Activity Work Group

그림 3 액티비티의 수행자 할당

기존의 수행자 할당은 미리 정의되어 있는 역할을 모델링 시점에 액티비티에 할당하는 방식이다. 그렇기 때문에 런타임 시점에서 수행자의 변경이 불가능하다. 하지만 협업 액티비티에서는 모델링 시점에서 액티비티에 WorkGroup을 할당하고 그 그룹의 멤버는 Role Agent를 통해 런타임시에도 변경이 가능하기 때문에 더 유연한 수행자 할당이 가능하다.

그림 3 은 일반 액티비티와 협동 액티비티의 역할(ROLE) 수행자 할당을 비교 한다. 일반 액티비티는 액티비티 하나에 한 역할이 할당되더라도 그 역할에 속하는 한명의 수행자가 작업을 처리하게 된다. 하지만 협동 액티비티는 Role Agent를 통해 구성된 각 역할(ROLE)의 수행자들이 모여 WorkGroup을 생성하고 그것이 협동 액티비티에 수행자로 할당된다.

3.2 Workflow Engine 확장

워크플로우 엔진은 프로세스 정의 언어를 받아들여 이를 해석하고 그에 맞게 협동 액티비티를 처리하는 부분이다. 엔진에서는 워크아이템의 할당 메커니즘, 액티비티 인스턴스의 생성/소멸 시점, 협동 액티비티를 지원하는 협동 서버의 확장을 필요로 한다.

3.2.1 워크 아이템 할당 메커니즘

워크리스트 핸들러는 수행자와 워크플로우 엔진에 의해 유지되는 작업 목록 사이의 상호 작용을 관리하는 소프트웨어 컴포넌트를 말한다. 이것은 워크플로우 관리 시스템에서 워크아이템이 전달 되도록 하고 수행자와 워크플로우 관리 시스템 사이에서 완료 통지 또는 다른 작업 상태 조건을 전달해 준다.

현재 워크리스트 핸들러에서 수행하는 워크아이템 할당 메커니즘은 수행자 타입에 따라 달라진다. 액티비티의 수행자에 HUMAN이 할당되어 있으면 그 HUMAN에 워크아이템을 할당한다. 그리고 수행자에 ROLE이 할당되어 있으면 그 ROLE을 가진 수행자 모두에게 워크아이템을 할당한다. 그리고 그 중 누군가가 워크아이템을 가져가면 나머지 사람들에게 할당된 워크아이템을 삭제하는 메커니즘을 사용하고 있다.

협동 액티비티를 지원하기 위해서 ROLE에 워크아이템을 할당 할 때 워크아이템 할당 메커니즘을 확장해야 한다. 액티비티의 처리 시 협동 액티비티인지를 판단하고

만약 협동액티비티일 경우 할당된 모든 수행자들에게 워크아이템을 할당해야 한다. 이 메커니즘을 통해 기존의 일반 액티비티와 협동 액티비티 모두 지원이 가능하다.

3.2.2 워크플로우 액티비티 인스턴스의 생성과 소멸

프로세스를 구성하는 각 액티비티는 실행시점에 액티비티 인스턴스를 생성하게 된다. 협동 액티비티는 여러 수행자에 의해 수행되기 때문에 액티비티 인스턴스의 생성 시점과 소멸시점이 일반 액티비티와 다르다. 협동 액티비티 인스턴스의 생성 시점은 크게 두 가지의 방법으로 구분하여 정의할 수 있다.

첫 번째는 시간의 제약을 통해 액티비티 인스턴스의 생성 시점을 결정하는 방법이다. 모델링 시점에서 협동 액티비티에 인스턴스 생성 시간을 정의하고 그 시간이 되면 액티비티의 인스턴스가 자동으로 생성되어 진다. 정의 되어 지는 시간에는 상대 시간(이전 액티비티의 상태가 완료상태가 된 일정시간 후 또는 프로세스의 상태가 활성화 상태가 된 일정 시간 후 등) 또는 절대시간으로 정의가 가능하다.

두 번째 경우는 협동 액티비티 수행자들의 협동 서버에 대한 연결 수에 제약을 두는 방법이다. 이 방법은 협동 액티비티를 이루는 참여자들의 협동 서버에 연결 수에 제약을 두어, 실시간으로 연결되어지는 수에 따라 액티비티 인스턴스의 생성 시점이 결정되어 지는 방법이다.

협동 액티비티 인스턴스의 소멸 시점은 모든 수행자의 업무가 끝날 때, 즉 협동 액티비티를 수행하는 모든 구성원들의 협동 액티비티 상태가 완료상태가 되었을 때 액티비티 인스턴스가 소멸된다.

액티비티 인스턴스의 생성 시점과 소멸 시점은 실시간으로 처리되어야 하는 협동 액티비티의 특성 때문에 많은 예외가 존재한다. 이에 워크플로우 엔진에서는 이런 예외들에 대한 정책을 미리 세우고 그 정책에 따라 예외 처리에도 신경을 써야 한다.

3.2.3 Cooperative Server

실시간 처리를 지원하기 위해서는 각 수행자 간에 데이터 전달과 애플리케이션으로의 통신 세션을 위한 협동 서버(Cooperative Server)를 필요로 한다. 협동 서버는 협동 액티비티들이 공유하는 데이터를 저장하기 위한 데이터베이스와 협동 액티비티들이 실시간으로 작업을 처리하기 위한 애플리케이션들을 포함하고 있다.

그림 4는 협동 서버를 통한 클라이언트들의 세션 할당 메커니즘을 나타내고 있다. 그림에서의 Client1, Client2, Client3 은 협동 액티비티 애플리케이션 A를 수행해하는 수행자들이다. 애플리케이션 A 는 다수의 사용자에 의해 작업의 처리가 가능한 협동 애플리케이션이다. 다수의 클라이언트들이 한 애플리케이션에 연결되기 위해서는 하나의 세션이 생성되고 그 세션에 연결해야 한다. 그러기 위해서는 한 세션의 정보를 모든 클라이언트들이 공유 할 수 있어야 한다. 워크플로우 엔진은 협동 서버의 위치를 알고 있고, 협동 서버는 엔진과의 인터페이스를 통해 엔진에게 서버 관련 정보를 서비스 한다는 전제 아래에서 다음의 메커니즘이 이루어진다. 우선 엔진은

애플리케이션 A에 대한 위치 정보와 수행자 정보를 가지고 있다. 협동 액티비티가 활성화 되면 각 클라이언트에게 워크아이템을 보내기 전에 협동 서버의 인터페이스를 통해 협동 서버의 애플리케이션 A에 세션을 생성한다. 그 후에 각 클라이언트에 워크아이템과 함께 애플리케이션 A의 세션 정보와 위치 정보 등을 함께 넘겨준다. 그러면 각 클라이언트들은 그 정보를 통해 애플리케이션에 접근하여 작업을 수행한다. 이와 같은 메커니즘을 통해 각 클라이언트들이 협동 애플리케이션의 세션을 알고 그 세션에 연결이 가능하다.

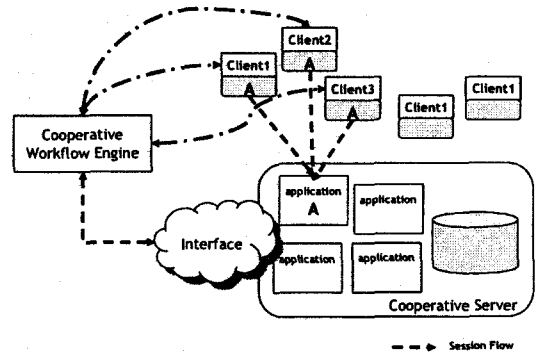


그림 4 워크플로우 엔진과 협동 서버의 세션 메커니즘

협동 서버의 구성 방법에는 엔진과 협동 서버를 분리하는 방법, 엔진과 협동 서버를 통합하여 엔진의 플러그인으로 협동 서버가 존재하는 방법, 기존의 서버를 협동 서버로 이용하는 방법이 있다.

먼저 워크플로우 엔진과 협동 서버를 따로 존재할 경우 엔진은 프로세스와 관련된 부분만을 처리하고 협동 서버는 협동 액티비티의 처리만을 담당하게 된다. 이때 세션 정보는 협동 서버를 통해 엔진이 가지고 있게 된다. 이를 위해서는 엔진과 협동 서버간의 미리 정의된 인터페이스를 필요로 한다. 이 같은 경우 네트워크 부하, 또는 서비스 처리의 지연 등의 문제가 발생할 수 있다.

이를 해결하기 위한 방법으로는 워크플로우 엔진의 플러그인으로 협동 서버가 존재하는 방법이다. 워크플로우 엔진에 협동 서버를 돌리게 되면 엔진과 서버 사이에 특별한 인터페이스가 필요 없게 되고 엔진과 서버 사이에 네트워크 트래픽을 염려 하지 않아도 된다.

그리고 다른 방법은 기존에 존재하는 서버를 협동 서버로 사용하는 방식이다. 대표적인 예로 인터넷을 통해 음성이나 회화로 대화하기 위한 통신 소프트웨어인 마이크로소프트사의 넷 미팅의 방식을 이용하는 것이다. 넷 미팅은 어용자 위치 서버(uis.microsoft.com)라는 서버를 가지고 있다. 이것을 통해 대화 상대방을 탐색하여 그 대화상대와 통신을 하는 것이다. 이를 활용해 세션 정보 서버를 따로 두고 세션이 생성되면 그 서버에 세션 정보가 올라가게 한다. 그러면 다른 사용자들은 자신이 수행할 협동 액티비티의 세션 정보를 어용자 위치 서버에서 찾는 방식으로 서버를 구성 할 수도 있다.

3.3 Runtime Client 확장

3.3.1 Activity 완료 통지

수행자가 자신의 업무를 모두 마치게 되면 수행한 업무에 관련 데이터를 애플리케이션 데이터베이스나 관련 데이터베이스에 저장하고, 워크플로우 엔진에 업무를 마쳤다고 통보 하게 된다. 그 후에 액티비티의 상태는 완료 상태가 되고 프로세스는 다음 액티비티로 진행된다. 일반적인 액티비티에서는 한명의 수행자에 의해 작업이 끝나면 데이터베이스에 저장하고 워크플로우 엔진으로 완료 메시지와 관련 데이터를 전송함으로써 액티비티의 완료 처리가 된다. 하지만 협동 액티비티에서는 작업 인원 전체가 작업을 완료해야 액티비티가 끝난 것이다. 그렇기 때문에 협동 서버의 애플리케이션은 현재 수행자들이 모두 완료 상태가 될 때 까지 기다렸다가 모두 완료상태가 되면 워크플로우 엔진에 액티비티 완료 메시지를 보낸다.

4. 결론 및 향후 연구 방안

본 논문에서는 워크플로우 관리 시스템의 협동 액티비티를 지원하는 방안에 대해 연구해 보았다. 기존의 워크플로우 관리 시스템을 확장하여 비즈니스 프로세스 모델, 워크플로우 엔진, 런타임 클라이언트의 관점에서 협동 액티비티를 지원할 수 있는 프레임 워크를 구성하였다. 이런 협동 액티비티를 지원함으로써 실제 기업의 비즈니스 프로세스를 더욱 효과적으로 처리가 가능해진다. 향후 연구 과제로는 본 논문에서 제안한 확장 내용을 직접적인 워크플로우 관리 시스템에 적용함으로써 협동 액티비티의 지원을 원활하게 하는 것이다.

참 고 문 헌

- [1] Khaled Gaaloul, Francois Charoy, and Claude Godart, "Cooperative Processes for Scientific Workflows", LNCS 3993, pp. 976-979, 2006
- [2] Lizhen Cui, Haiyang Wang, "Research on Cooperative Workflow Management Systems" LNCS 3168, pp. 359-367, 2005
- [3] Daniela Grigori, Hala Skaf-Molli, Francosi Charoy, "Adding Flexibility in a Cooperative Workflow Execution Engine" LNCS 1823, pp. 227-236, 2000
- [4] The Workflow Management Coalition, "Workflow Process Definition Language - XML Process Definition Language" 『Workflow Management Coalition Specification』 Document Number WFMC-TC-1025 Document Status - 1.0 Final Draft
- [5] Yuyue Du, Changjun Jiang, "Towards a Workflow Model of Real-Time Cooperative System" LNCS 2885, pp. 452-470, 2003
- [6] The Workflow Management Coalition, "The Workflow Reference Model", 『Workflow Management Coalition Specification』 Document Number TC00-1003 Document Status - Issue 1.1, 1995
- [7] The Workflow Management Coalition,

"Terminology & Glossary", 『Workflow Management Coalition Specification』 Document Number WFMC-TC-1011 Document Status - Issue 3.0, 1999
 [8] 심성수, 김광훈, "워크케이스 기반의 초대형 워크플로우 시스템 아키텍처", 한국 데이터베이스 학회 추계권 퍼런스논문집, pp. 203-416, 2002