

이동 에이전트를 이용한 계층적 조정 모델 기반 협력 작업 응용 개발 환경

김영민*, 이상호**

Collaborative Work Applications Development Environment based on Hierarchical Coordination Model using Mobile Agent

Young-Min Kim*, Sang-Ho Lee**

요약

분산 환경에서 다중 사용자간의 협력 작업을 효과적으로 지원하는 협력 작업에 대한 요구가 급격하게 증가하고 있다. 이러한 협력 작업은 그룹 통신 기술, 분산 처리 기술 등의 다양한 기술적 사항들이 해결되어야 하며, 협력 작업 응용이 갖는 공통적인 요구 사항들이 많이 존재하므로, 응용을 개발할 때마다 이러한 요구 사항들을 중복하여 개발하는 것은 많은 개발 비용이 소요되며 중복 투자되는 부분이 많게 된다. 따라서, 협력 작업 응용이 공통적인 요구 사항들과 응용을 보다 경제적으로 개발할 수 있는 체계적인 개발 환경이 필요하다. 이 연구에서는 기존의 이동 에이전트의 조정 모델을 협력 작업응용에 적합하도록 개선한 계층형 역할 기반 조정 모델을 제안하고, 이를 이용해서 이동 에이전트 기반 협력 작업 응용을 위한 개발 환경을 설계한다. 계층형 역할 기반 조정 모델은 이동 에이전트들의 다층형 그룹 개념을 제공하고, 각각의 그룹별로 효과적인 조정 정책 구현을 가능하게 하며, 그룹 단위 별로 역할을 부여해서 효과적인 협력 작업 응용 개발을 지원할 수 있다.

Abstract

The requirements of Computer Supported Cooperative Work supporting efficient cooperative or collaborative works between multi-users have been increasing in distributed environments. The various technical sections such as group communication technology and distributed processing technology should be provided in Cooperative Work. The replicated development of Cooperative Work applications of a number of common requirements increases development costs enormously and duplicated investment parts. Therefore, systematical development environments are required to develop these common requirements and applications efficiently in Cooperative Work applications development. In this study, we propose the hierarchical role-based coordination model that improves the coordination model of legacy mobile agent to be appropriate in Cooperative Work applications, and design the development environment for Cooperative Work applications based on mobile agent. The proposed hierarchical role-based coordination model provides multi-layered group concepts of mobile agent, and enables implementation of efficient coordination policy per group. Additionally, it supports efficient Cooperative Work application development by role assignment per group unit.

▶ Keyword : 계층적 조정 모델(Hierarchical Coordination Model), 협력 작업(Collaborative Work)

• 제1저자 : 김영민

• 접수일 : 2006.05.09, 심사완료일 : 2006.05.22

* 부산경상대학 의료정보과 조교수, ** 충북대학교 전기전자컴퓨터공학부 교수

I. 서론

최근 들어 컴퓨터 기술과 통신 기술의 급속한 발전으로 인해 자신의 작업 공간에 있는 적은 양의 정보만을 이용하여 의사 결정을 하던 과거와는 달리 근래에는 개방된 환경 하에서 지역적으로 분산되어 있는 사람들이 서로의 정보를 공유하여 의사 결정을 하는 형태로 작업 환경이 바뀌고 있다. 이러한 변화로 인해 분산 환경에서 다중 사용자간의 협력 작업을 효과적으로 지원하는 협력 작업(Computer Supported Cooperative Work)에 대한 요구가 급격하게 증가하고 있다[1].

협력 작업은 적용되는 조직의 형태나 사용하는 사용자들의 요구에 따라서 다르게 구성되며, 이에 따라서 응용을 다시 개발해야 하는 문제가 있다. 현재와 같이 기업 내의 조직 등이 급격하게 변경되는 경우에는 이러한 문제가 더욱 심각하다. 특히 대규모 조직에서의 시스템들은 대규모 정보 처리에 대한 작업이 많으며, 전체 작업의 형태가 상위 계층의 형태는 미리 결정되어 지는 경우가 있으나, 각각의 단계들의 하위 계층은 미리 정해지지 않고, 상황에 따라서 동적으로 변경되는 형태를 갖는다. 또한, 작업의 형태가 동적인 방식과 비동기적인 방식을 혼합되어 있는 형태를 갖는다. 이러한, 환경의 변화는 정적인 형태가 아닌 동적이고 확장성이 매우 뛰어난 형태의 응용의 형태를 요구한다[2].

기존의 연구들에서는 응용의 구조나 형태는 미리 정적인 형태로 정의되고, 응용 내에서의 상호 작용의 형태를 동적으로 결정하는 부분에 초점을 맞추고 있다[3]. 그러나, 이러한 방식은 협력 작업에 참여하는 사용자들과 적용되는 환경의 구조가 일정한 형태를 갖는다는 것을 전제로 하고 있으며, 단순히 조정 정책만을 초점으로 맞추고 있기 때문에 다양한 환경에서 유연하고 동적인 적응성을 갖는 응용의 개발이 불가능하다.

복잡한 조직에서는 상호 작용의 형태에 따라서 하나의 그룹 안에 여러 소그룹이 형성될 수 있으며, 각각의 소그룹이 다시 여러 소그룹으로 구성될 수 있는 다단계의 구조를 갖는 그룹 활동이 이루어지며, 각각의 그룹 별로 그룹 정책과 역할을 동적으로 부여할 수 있고, 하나의 그룹이 다른 그룹의 소그룹으로 동적으로 이동 할 수 있어야 하나, 기존의 연구에서는 이러한 다단계 그룹의 정책과 역할 부여나 그룹의 동적인 관리가 어렵다[4, 5]. 이를 해결하기 위해서는 상태를 가지고 네트워크 상을 자율적으로 움직일 수 있

으며, 개체들간의 동적인 상호 작용을 지원하는 이동 에이전트를 이용함으로써 유연하고 동적인 하부 구조를 갖는 협력 작업 응용 개발이 가능하다.

협력 작업은 이동 에이전트를 도입함으로써 다음과 같은 작업이 가능하다. 사용자 인터페이스 객체를 이동 에이전트로 구현하는 경우, 사용자 인터페이스안에 그룹 조정 프로토콜과 상호협력에서 멤버의 역할을 기반으로 한 적절한 권한을 포함시켜서 구현하고, 사용자의 위치로 이동해서 미리 정의된 프로토콜을 이용해서 그룹 활동을 지원할 수 있다. 네트워크가 지속적으로 연결되어 있어야 하는 기존의 협력 작업 환경을 확장 시킬 수 있으며, 워크플로우에서 같이 참여자 사이를 이동해 다니면서 각각의 참여자의 환경에 따라서 다양한 처리 단계를 지원할 수 있다[6, 7].

기존의 이동 에이전트 시스템에서 지원하는 조정 모델(Coordination Model)은 협력 작업에서 가장 중요한 상호 작용의 형태인 그룹 메커니즘과 역할 기반 상호 작용을 고려하고 있지 않다. 따라서, 협력 작업 응용 개발을 위해서는 다층형 그룹 지원과 그에 따른 역할 기반 관리 방식이 도입된 이동 에이전트의 조정 모델이 필요하며, 이를 효과적으로 지원하는 응용 개발 환경이 필요하다.

이 연구에서는 여러 에이전트들을 다층의 그룹으로 관리할 수 있고, 각각의 그룹을 역할 기반으로 관리할 수 있는 계층형 역할 기반 조정 모델을 제안한다. 제안된 모델은 하나의 그룹을 여러 이동 에이전트들의 조합 또는 하부 그룹들의 조합으로 이루어질 수 있는 계층형 구조를 정의하고, 이를 위한 기능들을 정의한다. 또한, 여러 그룹들로 구성된 환경에서 그룹 단위의 역할을 부여하는 역할 모델을 제안한다. 이 연구에서 제안하는 계층형 역할 기반 조정 모델은 이동 에이전트들의 다층형 그룹 개념을 제공하고, 각각의 그룹별로 효과적인 조정 정책 구현을 가능하게 하며, 그룹 단위 별로 역할을 부여해서 효과적인 협력 작업 응용 개발을 지원하도록 한다.

II. 관련 연구

2.1 협력 작업에서의 접근 제어

협력 작업은 물리적으로 분산된 다수의 사용자들이 공동된 작업을 수행하는데 있어서 상호 협력하고 이를 위해서

객체들로 구성된 환경을 공유하는 것이며, 접근 제어란 어떤 주체(subject)가 어떤 객체(object)에 대하여 어떤 목적을 갖고, 어떤 조건 하에서 접근할 수 있는지를 다루는 것이다. 여기서 주체는 능동적인 실체의 집합을 말하며, 객체는 수동적 자원의 집합을 말한다[8].

분산 환경 하에서 일어나는 대부분의 협력 작업은 정보의 교환을 통해 발생한다. 협력 작업은 정보(아이디, 파일, 프로세스)에 의존하며, 정보 공유에 대한 메커니즘은 사용자들 사이에서 협동이 끊어질 없이 수행되도록 해주어야 한다. 접근 제어는 이러한 정보 공유 시스템에서 효율적인 상호작용을 지원하기 위해서, 권한 없는 사용자의 접근으로부터 공유 정보를 보호하여야 할 뿐 만 아니라 권한 있는 사용자들 사이의 공유 정보에 대한 동시 접근 등에 대한 결정을 지원할 수 있어야 한다. 또한 이러한 결정의 조건은 상호 협력 과정에서 동적으로 생성될 수 있다.

전통적인 접근 제어인 강제적 접근 제어, 자율적 접근 제어는 규칙 수준에서 접근 제어 서비스를 제공하며, 각 사용자에게 권한을 직접 할당함으로써, 대규모화되고 복잡화 되는 접근 제어 요구를 만족시키기 어렵다.

역할 기반 접근 제어(Role-Based Access Control)는 전통적인 제어 기법과 달리 정보에 대한 사용자의 권한 부여 여부를 각 사용자의 식별자나 이미 정의된 규칙에 의해 판단하지 않고, 사용자가 소속된 조직 내에서의 역할에 의해 결정하는 특징을 가지고 있다. 따라서, 역할과 객체간의 관계로 접근 권한을 관리함으로써, 사용자와 객체의 수가 대단히 많은 분산·기업 환경에 적합한 특성을 제공한다[9].

2.2 이동 에이전트의 접근 제어

이동 에이전트는 분산 응용을 개발하기 위한 새로운 패러다임을 제공한다. 이동 에이전트는 네트워크상의 노드와 노드 사이를 클래스의 형태가 아니라 자신의 실행 상태를 가지고 네트워크를 자율적으로 이동할 수 있는 구조를 갖는다. 이러한 특징으로 인하여, 실행되는 구조나 환경에 따라서 자신이 판단해서 자신을 이주 시키거나 특정 코드를 실행시킬 수 있으며, 다른 이동 에이전트들과 상호 작용을 할 수 있다. 따라서, 가장 유연하고 동적인 분산 시스템 개발 환경으로 각광 받고 있다[10].

협력 작업 응용들은 그룹 활동에 참여하는 사용자들의 상호 작용을 지원한다. 즉, 각각의 사용자들의 다양한 상호 작용에 관한 요구를 반영해야 한다. 따라서, 사용자 간의 상호 작용은 궁극적으로 사용자를 대리하는 에이전트 간의 상호 작용으로 매핑되므로, 에이전트의 능동적인 행동과 에이

전트 그룹을 고려한 접근 제어가 필요하다.

기존의 이동 에이전트 시스템에서의 접근 제어 방식은 약의적인 이동 에이전트로부터의 시스템 보호와 시스템으로부터의 에이전트 보호 등의 보안 측면에 초점을 맞추고 있으며, 이러한 보호 측면으로써 사용된 제어기법으로는 자율적 접근제어 기법인 접근 제어 리스트 (ACL : Access Control List)와 역할 기반 접근 제어 방식 등이 사용되고 있다[11, 12].

2.3 이동 에이전트 기반 협력 작업 응용을 위한 접근 제어

이동 에이전트의 보안 위주 접근 제어는 단지 개별 호스트의 보호와 이주되는 에이전트의 보호에 초점을 맞추고 있다. 그러나, 실제적으로 이동 에이전트를 이용한 응용들을 지원할 수 있는 응용 레벨의 접근 제어는 고려하지 못하고 있다. 또한, 기존의 협력작업 응용에서 널리 사용되는 역할 기반 접근 제어를 이동 에이전트를 하부구조로 사용하는 협력 작업응용에 적용시킬 경우 문제가 발생한다.

이동 에이전트 기반의 협력 작업응용을 지원하기 위한 접근 제어는 보안 측면에서의 접근 제어 뿐 아니라, 이동 에이전트의 자율적이고 이주 가능한 특징을 고려하여 응용 측면에서의 작업에서 발생할 수 있는 접근 제어 요구에 유연하게 대처할 수 있어야 한다.

III 이동 에이전트를 이용한 계층적 조정 모델

이 장에서는 협력 작업 응용 개발을 위해 필요한 요소들을 이동 에이전트에서 지원하기 위해서 이동 에이전트간의 계층형 역할 기반 조정 모델을 제안한다.

3.1 계층적 조정 모델

계층적 조정 모델은 이동 에이전트를 계층적으로 관리하며, 협력 작업의 개발을 용이하게 하기 위해 이동 에이전트를 기반으로 한 동적이고 확장성 있는 이동 에이전트의 다층형 그룹 개념을 제공하고, 그룹 단위별로 역할을 부여해서 효과적인 협력 작업 응용 개발을 지원하도록 한다. 계층적 조정 모델은 계층 그룹 관리, 공유 객체 관리, 역할기반 접근 제어, 동적 조정 정책 등의 4개의 구성요소를 포함한다.

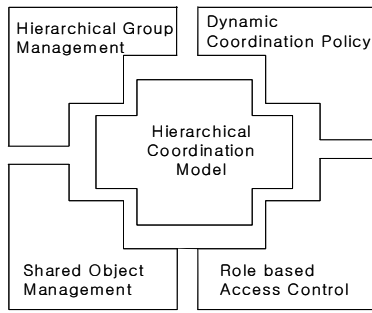


그림 1. 계층적 조정 모델
Fig 1. Hierarchical coordination model

3.2 계층적 그룹관리

3.2.1 이동 에이전트의 계층적 형태의 그룹핑

그룹 활동의 변화에 따라 그룹의 능력을 변화시키기 위해서는 먼저 그룹의 구성원이 재구성되어야 하고 구성원들의 관계가 기능적으로 분리 가능해야 한다. 또, 그룹을 운영하는 체제가 변화를 수용할 수 있는 제어 방식을 취해야 한다. 이를 위해 그룹의 운영 방식을 추상화된 일반적인 단계에서 세부적인 단계로 이전하기 위해 그룹의 구성원들이 이동 에이전트들에 대해서 계층적인 관리가 가능하도록 한다. 하나의 그룹은 여러 구성 요소들로 구성될 있으며 구성 요소들은 단일 이동 에이전트일 수도 있으며, 또 다른 그룹이 될 수 있다.

[정의 3-1] 그룹(group)

$G = \{ \langle x_1, x_2, \dots, x_n \rangle, I_c \}$
 $I_c = \{ GN, GT, PL, R, Max \}$
 $x \in \{ G, agent \}$
 where,
 G : Group
 x_i : members of the group $n \geq 0$
 I_c : information of group context
 GN : name of Group
 GT : topic of Group
 PL : list of Participants
 R : role of Group
 Max : max number of Participants' count

3.2.1 계층적 구조를 갖기 위한 에이전트 구조

그룹에는 그룹 구성원들의 상호 작용을 지원하고 그룹 관리를 책임지는 그룹 관리자가 필요하다. 그룹 관리자는 그룹을 관리하기 위한 연산들과 그룹 컨텍스트 정보를 관리한다. 이동 에이전트가 그룹에 참여하고자 하는 경우에는 그룹 관리자로부터 그룹에 관한 정보를 전달받고, 그룹 관리자에서 제공하는 그룹 참여 연산을 통해서 그룹에 참여한다. 그룹에 참여하는 에이전트는 같은 그룹내의 에이전트들과 상호작용할 수 있으며, 그룹 관리자를 이용해서 다른 그룹과 통신할 수 있다.

[정의 3-2] 에이전트(agent)

$Group = \{ Parent, ChildList \}$
 $agent \in \{ Parent, Child \}$
 $Parent = \{ Name, MgmtOp, ChildRefList, Role, Rule, GroupState \}$
 $Child = \{ Name, ExecutionModule, ParentRef, Role \}$
 where,
 $Parent$ is an entity which can have sub-agents and acts as Group Manager
 $Child$ is an entity which have no sub-agents and acts as Group Member
 $Name$ is Agent Name
 $MgmtOp$ is the Operation for Management of Child Agent
 $ChildRefList$ is References of Child Agents
 $ParentRef$ is Reference of Parent Agent
 $Rule$ is used for management of Group Activity performed by Child Agent

3.2.3 계층 그룹 관리 방식

이동 에이전트는 생성, 소멸, 복제, 이동 등의 기능을 통해서 관리된다. 이때, 생성, 소멸, 복제, 이동의 주체가 하위 노드를 갖는 부모 노드인 경우에는 부모 노드에만 영향을 미치는 것이 아니라 그룹 전체에 대한 연산으로 간주된다.

[정의 3-3] 에이전트 관리 연산

$Op = \{ Create, Destroy, Migration, Clone \}$
 $Migration \{ ToInnerGroup, ToNewGroup \}$
 $Op \odot X \equiv \{ Op \odot X + \Phi \}$
 $\Phi \equiv nul, \quad \text{if } X \text{ is Child Node}$

$$\Phi = \{ \forall x (Op \odot x, x \in X), \\ \text{if } X \text{ is Parent Node} \}$$

3.2.4 하부 에이전트의 이주 방식

개별적인 이동 에이전트의 이주 뿐만 아니라, 하위 이동 에이전트를 포함하는 상위 에이전트의 이동을 지원하며, 상위 에이전트의 이동에 따라서 하위 에이전트가 함께 이동한다는 것을 보장한다.

에이전트 이주 기능 중에서 Embedded는 실제 상위 에이전트가 이주한 곳으로 하위 에이전트들이 이주하는 것을 말하며, Linking은 실제로 이주는 하지 않으며, 상위 에이전트가 가지고 있는 하위 에이전트의 정보를 이용해서 하위 에이전트에게 메시지를 전달하는 구조이다.

3.3 접근 제어 및 공유 객체

그룹에 참여하는 이동 에이전트에 능동적이고 정형화된 역할 부여를 가능하게 하고 부여된 역할을 갖는 이동 에이전트간의 상호작용에 대한 역할 관계를 표현할 수 있는 역할 관계 템플릿을 정의하고 이를 적용하여 기존의 역할 기반 제어 기법을 확장한 협력적 역할 기반 접근 제어 모델(collaborative role-based access control model) 및 이에 따르는 공유 객체 관리 모델을 제안한다.

3.3.1 협력적 역할 기반 제어 모델

역할 관계 템플릿을 사용한 협력적 역할 기반 접근 제어 모델은 역할간의 사용 관계와 제한을 기술한다. 이는 계층적 에이전트 그룹에서의 복잡한 역할 계층에 대한 추상화를 제공함으로써, 새로운 역할이 필요한 그룹이 생성될 때, 상속과 제약이 추가로 생성되는 새로운 역할의 생성을 최소화하고 빈번하고 동적인 그룹의 생성·소멸 및 그룹간의 접근 제어에 대한 상호작용을 지원한다. 또한, 협력적 역할 기반 접근 제어 모델은 그룹과 그룹간의 접근 제어에 대한 규칙을 적용함으로써, 역할간의 관계를 명시적으로 표현하여 동적인 그룹변화에 유연하게 대처하게 한다.

3.3.2 공유 객체 관리 모델

공유 객체 관리 모델은 협력 작업에서 사용되는 에이전트 그룹내의 역할 관련 템플릿 단위의 객체 그룹을 생성함으로써 유연한 객체 관리가 가능하도록 한다. 또한 그룹에 적용된 역할 관계 템플릿에 따른 역할을 지원하기 위하여 객체 그룹에서의 객체 관리 역할 관계 템플릿에 따라 구별되어 관리되어야 한다.

공유 정보의 동시 접근에 대한 제어는 다수의 사용자가 동시적으로 정보에 접근해야 하는 협력 작업을 지원할 때 더욱 중요하다. 제안된 모델은 각각의 잠금의 단위를 읽기/쓰기/공유/이동이라는 4개의 단위로 구분하여 보다 유연한 동시성 제어를 가능하도록 한다.

3.3.3 동시성 제어

공유 객체에 대한 사용자들간의 동시성을 제공하기 위해서는 객체를 사용하는 사용자가 객체를 사용할 때 객체에 대한 잠금을 요청하고 삭제하는 방식을 취함으로써 이루어질 수 있다. 그러나, 공유 객체를 사용하는 그룹간의 절차적, 결과적 연관성이 없는 경우 기존의 잠금 방식에 유연성을 부여함으로써, 그룹 내에서의 효율적인 상호작용을 지원할 수 있다.

이 연구에서의 잠금 방식은 각각의 객체에 대한 잠금을 READ_LOCK, WRITE_LOCK, SHARED_LOCK의 3단계로 구분하여 수행된다. 객체를 사용하는 사용자는 객체를 사용하기 전에 객체에 대한 잠금을 요청하고 요청한 결과가 참 값을 가질 때 해당 객체에 대한 요구를 수행할 수 있도록 한다.

3.4 동적 조정 정책

그룹간의 상호 작용의 형태는 그룹내의 조정 정책에 의해서 이루어진다. 조정 정책은 그룹의 형태, 그룹 내에서의 역할, 사용할 객체의 리스트, 참여자의 수, 그룹의 개방성 여부와 메시지 처리 방식 등으로 정의된다. 그룹 간의 상호 작용 또는 그룹 내에서의 상호 작용은 그룹간 또는 그룹 멤버들간에 메시지 전달을 통해서 이루어진다. 따라서, 조정 정책은 이러한 메시지를 어떻게 처리할 것인가가 가장 중요한 고려 사항이다. 이러한 메시지 처리는 그룹 내에서 발생하는 메시지와 그룹 밖에서 그룹 내로 전달되는 메시지로 구분할 수 있다.

그룹 멤버들간의 조정은 그룹 조정자(group coordinator)에 의해서 이루어진다. 그룹 조정자는 외부로부터 전달되는 이벤트나 그룹 내부에서 발생하는 이벤트를 적절하게 처리하는 역할을 수행한다. 전달되는 이벤트를 어떻게 처리할 것인가는 규칙들의 집합으로 구성한다. 규칙은 그룹 멤버들이나 외부로부터 전달되는 입력 이벤트에 의해서 유발(trigger)되는 표현으로 정의되며, 조건과 실행 부분으로 구성된다. 조건 부분은 이벤트 타입에 의해서 구성되는 논리적인 표현들이며, 그룹 조정자 안에 포함되어 있는 상태 정보들로 구성된다. 실행 부분은 출력 이벤트 전송, 내부 상태의 변화 입력되는 이벤트의 처리 중지 등의 명령어들로 구성된다.

그룹 정책의 형태는 그룹의 제목, 사용할 역할, 역할 관계 템플릿, 참여자 수, 그리고 그룹에서의 규칙들로 구성된다. 그룹에서의 역할은 리스트로 구성되며, 역할 관계 템플릿은 역할들간의 사용 관계를 나타낸다. 그룹은 객체를 공유할 수 있으며, 역할에 따라서 공유하는 객체가 다를 수 있고, 같은 객체를 공유하더라도 다른 권한으로 객체를 사용할 수 있다. 그룹에서 발생하는 사용자들간의 상호 작용을 적절하고 각각의 참여자들이 전달되는 이벤트에 대해서 어떻게 처리할 것인가를 규칙을 통해서 정의한다.

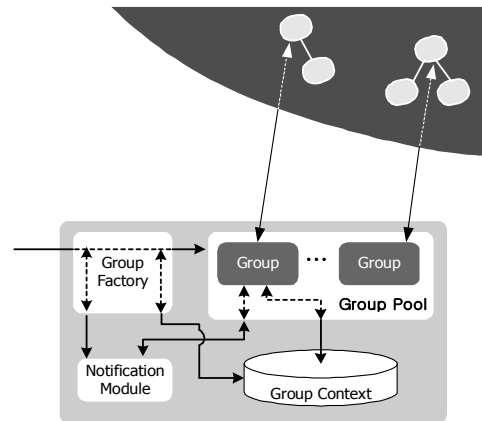


그림 2. 계층 그룹 관리자
Fig 2. Hierarchical group management

IV. 협력 작업 응용 개발 환경 설계

이 장에서는 제안한 계층적 역할 기반 조정 모델을 이용해서 협력 작업 응용을 위한 이동 에이전트 기반 개발 환경을 설계한다. 계층형 역할 기반 조정 모델을 이용한 협력 작업 응용 개발 환경은 자바 가상 머신(Java Virtual Machine) 위에서 동작하는 이동 에이전트 시스템인 IBM사의Aglet2.0.2를 이용해서 그룹 관리자 부분과 접근 관리자, 공유 객체 관리자와 그룹 조정자를 관리하는 계층 관리자 구성된다.

계층 그룹 관리자는 에이전트의 그룹에 대한 참여, 탈퇴 기능 등을 쉽게 해주는 도구이다. 따라서 여러 에이전트에 대한 정보를 참여자가 그룹에 참여하는 동안 계속 유지할 수 있어야 한다. 그룹 관리자는 그룹에 대한 정보를 개별적으로 유지한다. 그룹 관리자는 이러한 정보들을 일관성 있게 관리하여야 한다. 그룹 관리자는 외부 모듈과의 인터페이스로 사용된다. 새로운 그룹의 생성이나 그룹의 소멸 등의 요구에 대해 이동 에이전트들의 최상위 그룹 에이전트를 관리한다. 이를 위해서 그룹의 생명 주기를 담당하는 그룹 팩토리(Group Factory), 각각의 계층적으로 형성된 에이전트를 관리하는 그룹 (Group), 각각의 그룹에 대한 문맥을 관리하는 그룹 문맥(group context)으로 구성된다. 동시 모듈은 이벤트의 정의 기능과 정의된 이벤트를 그룹에 등록하는 기능을 수행한다. (그림 2)는 계층 그룹 관리자의 구성도이다.

공유 객체 및 접근 관리자는 공유와 잠금을 처리하는 공유 객체 관리자 인터페이스와 접근 제어 관리자 인터페이스로 구성된다.

공유 객체 관리자는 에이전트가 참여한 작업 그룹에 대한 객체 그룹의 생성 및 삭제 기능을 지원하는 도구이다. 따라서 객체 그룹에 등록되는 객체에 대한 정보를 에이전트에게 제공할 수 있어야 한다. 그리고 공유 객체 관리자는 객체 그룹내의 객체의 대한 인터페이스 정보를 역할 관계 템플릿에 따른 역할 단위로 관리하며 그 역할에 따라 서로 다른 인터페이스 정보를 제공하여야 한다. 공유 객체 관리자는 객체 그룹 관리자(Object Group Manager),객체 그룹 팩토리(Object Group Factory), 동시성 제어기(Concurrent Controller) 및 통지 모듈(Notification Module)로 구성되며 (그림 3)은 공유 객체 관리자의 구성도이다.

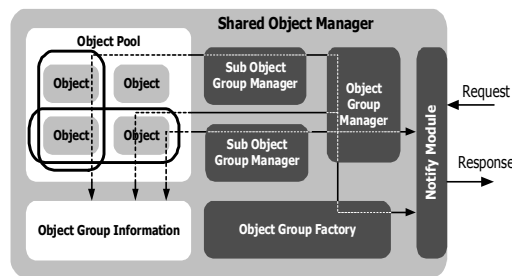


그림 3. 공유 객체 관리자
Fig 3. Shared object management

객체 그룹 관리자는 접근 제어 관리자와의 통신을 위한 인터페이스와 객체를 그룹화하여 객체의 생명주기를 담당하며, 객체 그룹 팩토리는 각각의 객체 그룹에 대한 정보를

관리한다. 동시성 제어기는 각각의 객체 그룹에 의한 객체의 공유에 따르는 동시성 문제를 제어하고, 통지 모듈은 공유 객체 관리자 내의 변화에 대한 통지 기능을 수행한다.

접근 제어 관리자는 역할에 따른 요구에 따라 객체 접근에 대한 허용 여부를 에이전트에게 제공하며, 같은 역할을 갖는 에이전트 일지라도 역할 관계 템플릿에 따라 서로 다른 접근 제어 기능을 제공함으로써, 이동 에이전트 기반 협력 작업응용의 요구에 따라 유연한 접근 제어를 수행한다.

접근 제어 관리자는 접근 결정기(Access Decision), 역할 제어기(Role Controller), 요구 처리기(Request Handler) 및 관리자 인터페이스(Administrator Interface) 등의 처리 모듈과 역할 풀(Role Pool), 역할 관계 템플릿 풀(Role Relation Template Pool) 등의 정보, 그리고 각 사이트에 위치하는 지역 접근 결정기(local Access Decision)로 구성된다. (그림 4)는 접근 제어 관리자의 구성도이다.

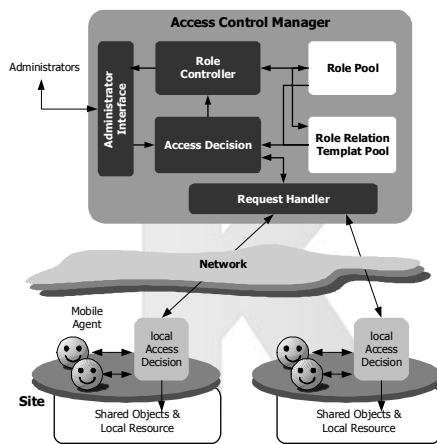


그림 4. 접근 제어 관리자
Fig 4. Access control management

접근 결정기는 요구 처리기로부터의 질의에 따라 접근 제어 여부를 결정하고 그 결과를 반환하며, 요구 처리기는 지역 접근 결정기로부터의 질의와 접근 결정기의 질의 결과에 대한 전달 기능을 수행한다. 역할 제어기는 역할 및 역할 관계 템플릿 정보를 관리하며, 관리자를 위한 기능을 수행한다.

통신 관리자는 지역 통신기(Local Communicator)와 이벤트 분배기(Event Distributor)로 구성된다. 지역 통신기는 에이전트로부터 이벤트들의 등록과 삭제를 담당하고, 이벤트 분배기에게 이벤트를 보내고 받는 역할을 하는 이벤

트 통지 모듈(Event Notification Module)과 비동기 통신을 위한 메시지 보드(Message Board)로 구성된다. 이를 위해 요구 처리기(Request Handler)를 두어서 에이전트들은 이를 통해 지역 통신기와 통신을 하도록 한다. 이벤트 분배기는 지역 통신기들의 통신을 담당하며, 지역 통신기들의 정보를 관리하며, 지역 통신기의 등록과 삭제를 처리한다. 이벤트 분배기의 전송기(Transmitter)는 지역 통신기에서 받는 이벤트들을 브로드캐스팅 해주는 역할을 한다.

V. 실험 및 평가

이 장에서는 계층형 조정 모델을 기반으로 설계한 이동 에이전트 기반 협력 작업 응용 개발 환경의 타당성을 입증하기 위해서 제안한 개발 환경을 이용해서 협력 작업 시스템을 개발한다. 이때, 실험에서 구현한 응용의 수행 과정을 통해 설계한 시스템의 각 모듈의 기능을 평가한다.

5.1 공동 저작 시스템

전체적인 교육 시스템은 저작자와 편집자를 위해서 각각 다른 기능을 제공한다. 저작자에게 제공하는 전체 저작물의 구조를 보여주는 뷰어와 저작을 위해 사용되는 화이트보드와 편집기로 구성된다. 편집자는 저작자가 작업한 내용을 수집해서 보여 주는 기능으로 구성된다. 그리고 저작자와 편집자 모두에게 조정 정책을 입력할 수 있는 편집기 기능이 추가된다. 공동 저작은 1명의 주저작자, 2명의 보조저작자, 그리고 편집자 1명으로 이루어진다. 공동 저작을 위한 응용은 저작물 편집을 위한 에디터, 채팅기능을 수행하는 채팅기, 그리고 화이트보드로 구성된다. 또한, 저작 책임자에게는 공동 저작에서 사용되는 동적 역할 및 역할 관계 템플릿을 관리할 수 있는 역할 관리 에이전트가 제공된다. 각각의 참여자들은 자신의 정책을 기술하고 이를 이용해서 계층적 이동 에이전트를 구성한다.

실험은 다음과 같은 저작 과정의 시나리오에 의해 수행되었다.

- 저작 작업을 수행한다. 저작 작업 수행 시, 접근 권한에 따른 접근 제어가 올바르게 수행되는 지를 검증한다.
- 주 저작자의 공동 저작 작업의 탈퇴/복귀에 따른 보조 저작자와의 역할 위임이 발생한다.

- 저작이 완료된 후, 편집 작업으로의 작업 형태의 변화가 발생한다. 역할 관계 템플릿을 사용하여 작업 형태의 변화에 따라 유연한 접근 제어의 수행 여부를 검증한다.
- 공동 저작 작업이 완료된다.

응용에서의 이동 에이전트들은 응용 개발자에 의해 정적 역할 클래스의 인스턴스를 갖게 되며, 작업 시작 시 주저작자의 공유객체 등록에 의해 (그림 5)와 같은 접근 권한이 형성되며, 이는 접근 제어관리자에 의해 관리된다.

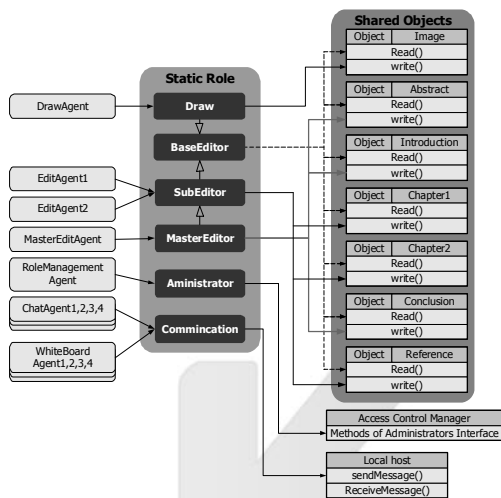


그림 5. 정적 역할에 대한 권한 부여
Fig 5. Permission assignment on static role

또한, 각각의 협력 작업자들은 이동에이전트의 정적 역할에 따라 (그림 6)과 같은 동적 역할을 할당 받는다.

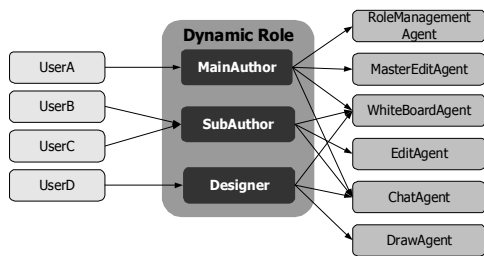


그림 6. 작업자에게 할당된 동적 역할
Fig 6. Dynamic role assigned to workers

따라서, 주 저작자 UserA는 역할 관리 에이전트를 통해 공동 저작에 필요한 동적 역할 및 역할 관계 템플릿을 관리하고, 전체 저작물에 대한 쓰기 연산을 수행할 수 있는 권한을 갖는다. 보조 저작자인 UserB, UserC는 각각 저작물의 각 장에 대하여 쓰기연산과, 다른 부분에 대한 읽기 연산을 수행할 수 있으며, 디자이너 역할의 UserD는 저작물에 대한 읽기 연산과 이미지 객체에 대한 쓰기 연산을 수행할 수 있다.

주 저작자인 UserA는 자신의 동적 역할인 MainAuthor를 보조 저작자인 UserB에게 위임하여, 협력 작업을 진행시킨다. 주 저작자의 작업 탈퇴에 따라 주 저작자의 역할이 보조 저작자 UserA에게 위임될 수 있으며, 이때, 주 저작자의 RoleManagementAgent와 MasterEditAgent의 소유권이 UserB에게 전달된다.

주 저작자가 공동 저작에 복귀하면 위임된 역할은 작업 초기 시로 복귀된다. 그 후, 보조 저작자들의 부분 저작이 완료되면, 편집 작업을 수행한다. 이것은 저작에서 편집이라는 상호작용의 형태의 변화를 발생시킨다. 이때, 기존의 편집자의 역할은 단지 모든 공동 저작물에 대해 읽기 연산만이 수행 가능하였다. 그러나, 편집을 위해 저작물에 대한 쓰기 연산이 필요하게 되며, 이를 위해 새로운 역할을 생성하는 대신, 역할 템플릿을 사용하여 편집자의 에이전트인 DrawAgent는 MasterEditor의 write() 메소드를 수행할 수 있도록 한다.

(그림 7)은 템플릿 적용 후 역할 사용에 따른 DrawAgent의 접근 권한 확장을 나타낸다.

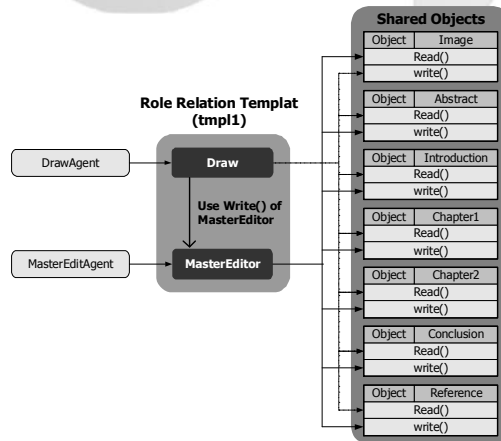


그림 7. 역할 관계 템플릿 사용에 따른 접근 권한의 변화
Fig 7. Change of access permission according to the use of role relation template

5.2 평가

실험에서 동적 역할과 정적 역할로의 역할 구분을 통하여 이동 에이전트의 특징을 보장하면서 동적인 역할 부여가 가능하다는 것과 역할간의 사용관계를 표시할 수 있는 역할 관계 템플릿을 사용하여 협력 작업에서의 작업 형태 변화에 따라 유연한 접근 제어를 수행함을 보였다.

제한된 역할 관계 템플릿의 사용은 다음과 같이 정형화되어 표현될 수 있다. 즉, 제약 조건을 고려하지 않은 $n(n \geq 2)$ 개의 역할이 있을 때, 각 역할 간의 관계 수는 다음과 같다.

$$\sum_{a=2}^n \left\{ \frac{1}{2}(n-(a-1)(n-(a-2))) \right\} \quad (n \geq 2)$$

또한, 이러한 역할 간의 관계를 상속에 의해서 표현하는 경우, 같은 수의 역할이 필요하다. 즉, 역할 상속에 의해 모든 역할 관계를 표현하기 위해 생성하여야 할 역할의 수는 다음과 같다.

$$\sum_{a=2}^n \left\{ \frac{1}{2}(n-(a-1)(n-(a-2))) \right\} \quad (n \geq 2)$$

반면, 역할 관계 템플릿에 따른 사용에 의한 관계 표현은 역할 사용의 전의에 의하여

$$2(n-1) \quad (n \geq 2)$$

로 표현 가능하다. 또한, 이러한 관계를 표현하기 위한 템플릿의 수는

$$n(n-1) \quad (n \geq 2)$$

와 같다.

(그림 8)는 위의 수식에 의하여 제안한 확장된 역할 기반 접근 제어와 사용과 기존 역할 기반 접근 제어의 역할 관계 표현과 역할의 수에 대한 비교이다.

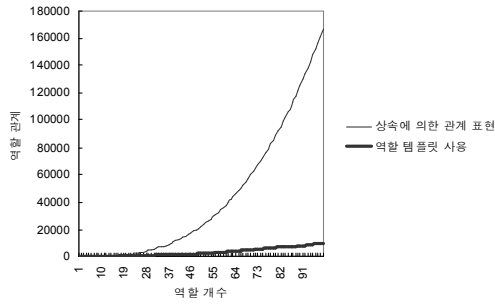


그림 8. 역할 관계의 수와 역할 수의 비교
Fig 8. Comparison between role relation and role count

따라서, 역할의 사용 관계에 의한 역할 관계의 표현은 새로운 역할 생성을 최소화하면서 역할 관계를 표현할 수 있다. 또한, 이러한 역할이 임시적으로 생성/폐기되어야 하고, 실행 시간에 일어나야 하는 이동 에이전트 기반 협력 작업 응용에 적용될 경우 보다 유연하고 효율적인 접근 제어의 수행이 가능함을 검증하였다.

V. 결론

이 연구에서는 분산 환경에서 다중 사용자간의 상호 협력적 응용 개발을 위한 계층적 조정 모델을 제시하였고, 제시한 모델을 이용해서 협력 작업 응용을 위한 개발 환경을 설계 및 구현하였다. 제시한 모델은 독립된 개체로써 존재하는 이동 에이전트가 아니라 상호 작용의 형태에 따라서 그룹 개념을 도입하고, 그룹 내에 또 다른 그룹을 포함할 수 있는 계층적 구조로 정의하였다. 또한, 기존의 역할 기반 관리 모델을 확장해서 계층적 그룹 관계에서 그룹 단위로 할당되는 역할을 정의하고 그룹의 계층적 구조를 표현하기 위해서 역할 관계 템플릿을 정의하였다. 이러한 계층적 역할 기반 조정 모델은 협력 작업 응용 개발에 필요한 요구 사항들을 만족함으로써, 이동 에이전트를 기반으로 한 유연성이고 동적인 협력 작업 개발을 가능하게 하였다.

이 논문에서 제안한 모델을 공동 저작 시스템에서만 적용시켜 실험을 하였으나 차후에 다양한 협력 작업 응용들에 적용하기 위해서는 유비쿼터스 환경에서의 협력 작업 지원에 대한 연구가 반드시 필요하다.

참고문헌

- [1] Andy Crabtree, Tom Rodden and Steve Benford, "Moving with the Times: IT Research and the Boundaries of CSCW", Computer Supported Cooperative Work (CSCW) :The Journal of Collaborative Computing, Springer, Vol. 14, No.3, June 2005, pp. 217-251.
- [2] Axel Guicking, Peter Tandler, "Agilo:A Highly Flexible Groupware Framework", In Proceedings of the CRIWG 2005, LNCS 3706, pp. 4956, 2005.
- [3] G. Cabri, L. Leonardi, F. Zambonelli, "A Web Infrastructure for People and Agent Interaction and Collaboration", Proceedings of the IEEE Ninth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), CS Press, NIST (USA), June 2000.
- [4] Goopeel Chung and Prasun Dewan, "Towards Dynamic Collaboration Architectures", In Proceedings of the CSCW04, 2004, pp. 1-10.
- [5] Saul Greenberg, "Enhancing Creativity with Groupware Toolkits," In proceedings of the CRIWG 2003, LNCS 2806, pp. 19, 2003.
- [6] Ichiro Satoh, "MobileSpaces: A Framework for Building Adaptive Distributed Applications using a Hierarchical Mobile Agent System", Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS'2000), pp. 161-168, IEEE Computer Society, April 2000.
- [7] Khaing Moe San, Hnin Aye Thant, Sint Sint Aung, Khin Mar Lar Tun, Thim The Naing, and Ni Lar Thein, "Mobile Agent Based Collaborative Learning System", Information and Telecommunication Technologies, 2005. APSITT 2005 Proceedings. 6th Asia-Pacific Symposium on pp. 83-88.
- [8] Andrew Rowley, "Dynamic Access Control for Shared Objects in Groupware Applications", In proceedings of the ECOOP'98 Workshop Reader, LNCS 1543, p. 286, 1998.
- [9] R. S. Sandhu, E. J. Coyne, "Role-Based Access Control Models", IEEE Computer, pp 38-47, 1996.
- [10] Danny B. Lange, M. Oshima, "Seven good reasons for mobile agents", Communication ACM 42, pp. 88-89, Mar. 1999.
- [11] 이호, "웹 기반 응용을 위한 직무 기반 접근 제어 시스템 모델 설계", 한국컴퓨터정보학회 9권 3호, pp 67-68, 2004.
- [12] 황유동, 박동규, "유비쿼티스 환경의 접근제어를 위한 확장된 GTRBAC 모델", 한국컴퓨터정보학회 10권 3호, pp 46-48, 2005.

저자 소개



김영민

1988년 2월 인하대학교
전자계산학과
1990년 2월 인하대학교
전자계산학과 이학석사
1998년 8월 충북대학교 대학원
전자계산학과(박사수료)
1992년~현재 부산경상대학교
의료정보과 조교수



이상호

1989년 2월 숭실대학교 대학원
전자계산공학과 공학박사
1981년 3월 ~ 현재 충북대학교
전기전자컴퓨터공학과 교수