

# 실시간 그룹웨어 응용을 위한 그룹 공유 영역 및 공유 객체 관리 모델

김준성 이재영 임헌규 임영환  
한국전자통신연구소 시각언어 연구실

## A Model for Group Shared Area and Controls of Shared Objects for Real Time Groupware Applications

JuneSung Kim JaeYoung Lee HunGyoo Lim YoungHwan Lim  
ETRI Visual Language Section

### 요약

본 논문에서는 그룹웨어(groupware) 또는 CSCW(Computer Supported Cooperative Work) 사용자들이 공유할 데이터들을 그룹 공유 영역에 위치시킨 후 그룹 공유 영역 내의 공유 객체에 대한 접근을 그룹웨어 사용자들에게 제공하는 구조를 제시하고, 그룹 공유 영역의 구현을 위해 실시간 그룹웨어 응용에 의존하지 않는 구조를 지닌 공유 영역 서버인 SAS(Shared Area Server)에 대해서 기술한다. 그룹웨어 응용에게 공유 영역에 대한 접근을 제공하는 SACI(Shared Area Client Interface)와 함께 SAS는 하이브리드 구조를 구성하고 SAS와 SACI 그리고 그룹웨어 응용 사이에는 비동기 메시지 전달 방식을 제공한다. 본 논문에서는 실시간 그룹웨어에서 사용하는 공유 객체에 대한 동시성 제어를 다루면서 기존 연구와의 차이점을 기술한다.

### 1. 서론

그룹웨어 또는 CSCW는 공동의 목표를 위해 공동 작업에 참여하는 그룹 사용자들을 보조해주는 컴퓨터 기반 시스템을 말하며, 그룹 사용자들에게 공유 환경에 대한 인터페이스를 제공한다. 그룹웨어는 사용자 상호 작용의 동시성 존재 여부에 따라 실시간(real-time) 그룹웨어와 비 실시간 그룹웨어로 분류할 수 있다[1].

실시간 그룹웨어는 지역적으로 떨어진 둘 이상의 사용자들이 자신의 컴퓨터 환경을 이용하여 동일 시간에 동시에 공동 작업을 수행할 수 있도록 지원해 주는 각종 응용 도구 및 통신 기술을 포함하는 제반 환경을 말한다[2]. 실시간 그룹웨어의 예로 그룹 공동 편집기, 그룹 의사 결정 시스템, 전자 회의, 그리고 MUD(multi-user dungeon)과 같은 게임 등을 들 수 있는데[1], 특히 그룹 공동 편집기는 그룹 사용자 사이의 상호 작용이 빈번하게 이루어진다는 점에서 많은 연구가 진행되어 왔다[1,5-6,9]. 그룹 공동 편집기는 편집 대상에

따라 문자 편집기, 그래픽 객체 지향 편집기, 그리고 픽셀 지향(pixel-oriented) 편집기로 구분될 수 있는데, 그룹 공동 편집기의 설계는 편집기의 제어 구조, 공유 객체의 동시성 제어 및 사용자 인터페이스와 밀접한 관계를 지닌다.

그룹 공동 편집기와 같은 전형적인 실시간 그룹웨어 응용의 제어 구조를 설계할 때는 사용자의 작업 행동이 실시간으로 처리되어야 한다는 제약성을 고려해야 하고 이에 따라 공유 객체의 처리 방법도 달라진다. 일반적으로 실시간 그룹웨어 응용에서 처리하는 공유 객체에 대한 동시성 제어는 분산 컴퓨팅 환경에서 사용하는 전통적인 동시성 제어 방법을 그대로 적용할 수 없는데 그 이유는 그룹웨어 응용이 컴퓨터와 컴퓨터 사이의 상호 작용 뿐만이 아니라 사용자와 사용자 사이의 상호 작용에 대한 제어 또한 고려해야 하기 때문이다.

실시간 그룹웨어 응용의 개발을 어렵게 하는 또다른 요인은 단일 사용자 응용(single

user application)의 사용자 인터페이스를 그룹 사용자 응용에 적용할 수 없다는 점인데, 동시성 제어 문제와는 별도로 사용자 인터페이스 문제는 모든 그룹 사용자에게 공동 작업의 진행을 동등하게 보여 주고 공동 작업 인식을 사용자 인터페이스에 포함시켜야 하기 때문이다.

지금까지의 연구는 특정 실시간 그룹웨어 응용을 위해 응용 차원에 종속적인 방향으로 개발되어 왔었는데, 이와 같은 개발 방향은 특정 응용 분야에 국한되는 것이어서 다른 응용에 적용하기가 어렵고 유지 보수도 힘들었다. 최근에 와서 응용 프로그램 개발자가 그룹 공동 편집기와 같은 실시간 그룹웨어 응용을 개발할 수 있는 라이브러리(library) 수준의 도구들에 연구가 활발히 수행되고 있는데[2,4-6], 이들 연구는 그룹 공동 편집기와 같은 실시간 그룹웨어 응용에서 도출되는 동시성 제어, 동기화, 통신 등의 문제점에 대한 해결을 일반화시키는 방향으로 진행되고 있다.

본 논문은 실시간 그룹웨어 응용 개발을 지원하기 위한 구조로서 개발자용 라이브러리를 포함하는 그룹 공유 영역(group shared area)라는 개념을 도입하였다. 그룹 공유 영역은 그룹웨어 응용이 처리할 모든 데이터들을 객체 단위로 저장하는 가상의 영역으로 볼 수 있는데, 그룹 사용자 또는 그룹웨어 응용은 그룹 공유 영역에 공유하기를 원하는 공유 객체를 그룹 공유 영역 안에 저장할 수 있으며 그룹 공유 영역에 저장된 모든 공유 객체에 대해서 접근할 수 있다. 그룹 공유 영역의 특징은 공유 영역에 놓이는 공유 객체가 특정 응용에만 의존하지 않는다는 점이다. 공유 객체는 특정 응용이 처리하는 자료가 될 수도 있지만, 다른 그룹웨어 응용이 공통적으로 사용할 수 있는 자료가 될 수도 있다. 또한 공유 객체는 구체적인 형상을 가지지 않는 개념적인 객체가 될 수 있는데, 서로 다른 응용이라도 공유 영역에 저장된 공유 객체를 해석할 수 있는 인터페이스를 가지고 있다면 추상화되어 있는 공유 객체에 접근할 수 있다.

그룹 공유 영역의 설계 목표는 그룹웨어 응용 개발자에게 공유 객체의 처리와 응용에 종속적인 사용자 인터페이스 부분을 분리하도록 하여 응용 개발자의 자유도를 높이는 데에 있다. 공유 객체의 처리와 사용자 인터페이스 부

분을 분리하기 위해서는 각 그룹웨어 응용마다 공유 영역 접근을 위한 통일된 인터페이스를 제공한다. 공유 객체의 처리는 하나의 중앙 집중 서버에서 관리하고, 그룹웨어 응용은 서버에 대한 인터페이스를 통해 공유 객체에 대한 접근을 한다. 공유 영역 서버와 공유 영역 인터페이스는 하나의 가상 공유 영역을 구성한다. 본 논문에서는 실시간 그룹웨어 응용 개발을 위한 그룹 공유 영역 개념을 소개하고 공유 영역 서버인 SAS(Shared Area Server)와 공유 영역 접근을 위한 인터페이스 SACI(Shared Area Client Interface)에 대해 논하겠다.

2장에서는 실시간 그룹웨어 응용에 관련된 기존의 연구에 대해서 소개하고, 3장에서는 본 논문에서 제시하는 그룹 공유 영역에 대한 정의를, 4장에서는 그룹 공유 영역의 설계와 기능에 대해서 논하겠다. 5장에서는 실제 적용 예로 이미 구현되었거나 구현 중인 실시간 그룹웨어 응용에 대해서 소개하고 마지막 6장에서 결론을 맺는다.

## 2. 기존 연구

실시간 그룹웨어의 특징은 사용자에게 다른 사용자가 함께 작업하고 있다는 공동 작업 인식(collaboration-awareness)를 제공해야 한다는 점이다. 즉 사용자는 다른 사용자와 공동 작업을 수행하고 있다는 사실을 알 수 있도록 사용자의 공동 작업 행위를 다른 사용자들에게 보여 주어야 한다. 분산 컴퓨팅 시스템이 사용자로 하여금 분산 환경의 자원을 마치 혼자 사용하는 것과 같은 관점을 제공하는 반면, 실시간 그룹웨어는 사용자의 공동 작업을 상호 볼 수 있도록 해야 한다.

그룹 공동 편집기와 같은 실시간 그룹웨어 응용 개발에 대해 많은 연구 결과가 발표되었는데[2,5,8,9], 실시간 그룹웨어 응용에 대한 연구는 크게 사용자 인터페이스 관련 연구와 공유 객체의 처리에 대한 연구가 활발히 이루어져 왔다. Bentley, et al.[4]은 User Display Agent라는 구성 요소를 두어서 사용자와 시스템 설계자가 사용자 인터페이스를 가공할 수 있도록 하였고, Greenberg[9]는 실시간 그룹웨어 응용에 적용되는 사용자 인터페이스의 요구 조건에 대해서 기술하였다. 그에 따르면 일반화된 그룹웨어 관련 사용자 인터페이스

는 응용 분야에 따라 서로 다르게 선택되어야 한다고 기술하였다. Sohlenkamp[8]은 Virtual Office 메타포어(metaphor)를 제공하는 DIVA 시스템을 통해 사용자의 동시 작업을 직관적으로 보여 주었다. 동시성 제어와 관련한 공유 객체의 처리 문제는 Ellis[1]에 의해 기술되었고, Newman-Wolfe[5]는 Display Manager를 통해 그룹 공동 편집기를 구현하였다.

실시간 그룹웨어 응용을 개발하는 데에는 크게 두 가지 방법이 있다[6]. 한 방법은 기존 단일 사용자 용 응용 프로그램을 그룹웨어 응용에 적용시키는 방법으로 이러한 방식을 공동 작업 투명성 방식(Collaboration Transparent Approach)라고 한다. 반면 처음부터 그룹 공동 작업의 요구에 맞도록 그룹웨어 응용을 설계하는 방법을 공동 작업 인식 방식(Collaboration Awareness Approach)라고 한다. 공동 작업 투명성 방식은 주로 사용자의 화면 출력을 공유하는 방식을 이용하여 기존 응용을 사용할 수 있다는 장점이 있으나, 실시간 그룹웨어의 주요 요구 조건인 다른 사용자의 상호 작용이 제한된다는 단점이 있다. 공동 작업 인식 방식의 경우는 설계 목표가 사용자 사이의 공동 작업을 제공하기 위함이기 때문에 그룹 공동 작업에 적절한 방법이지만 각 응용 분야에 따라 구현이 종속적이고 구현이 쉽지 않다는 단점이 있다.

그룹 공유 영역은 공동 작업 투명성 방식과 공동 작업 인식 방식을 모두 제공할 수 있는 실시간 그룹웨어 응용 개발을 위한 구조로, 공동 작업 투명성 방식에서 필수적인 그룹 간 통신 기능 제공과 공동 작업 인식 방식에서 필요한 공유 객체에 대한 제어를 포함하고 있다. 이는 그룹 공유 영역의 구조가 공유 객체의 처리와 사용자 인터페이스 사이의 독립성을 유지하기 때문에 가능하다. 먼저 그룹 공유 영역의 정의에 대해서 알아본다.

### 3. 그룹 공유 영역의 정의

그룹 공유 영역은 다음과 같은 5-튜플(tuple)로 정의된다.

$$\text{그룹 공유 영역 GSA} = (U, O^A, O^N, F, C)$$

그룹 공유 영역 GSA는 그룹 멤버 집합 U, 유

명 공유 객체(named shared object) 집합  $O^A$ , 무명 공유 객체(anonymous shared object) 집합  $O^N$ , 그리고 공유 객체 연산 집합 F와 그룹 멤버 사이의 통신 C로 구성된다. 그룹 멤버 집합은 다음과 같이 표현된다.

#### 3.1 그룹 멤버 집합

$$U = \{ u_j \mid u_j = (id_j, attr_j) \}$$

$u_j$ 는 그룹 멤버의 식별자로 GSA에 연결된 모든 그룹 멤버는 서로 다른 식별자를 가진다. 그룹 멤버는 GSA에 참여하는 사용자를 말하는데, 여기서 사용자는 최종 사용자가 아니라 최종 사용자가 사용하는 하나의 그룹 응용을 지칭한다. 즉 최종 사용자가 여러 개의 응용을 실행시킬 때, 최종 사용자는 여러 개의 그룹 멤버로써 공유 영역에 참여하게 된다.  $attr_j$ 는 그룹 멤버에 대한 정보를 나타내는데 실제의 내용은 그룹웨어 응용에 종속적이다.

#### 3.2 유명 공유 객체 집합

$$O^A = \{ o_i^A \mid o_i^A = (id_i, name_i, u_i, \Delta_i, st_i) \}$$

GSA에는 이름을 가지는 유명 공유 객체와 이름을 가지지 않는 무명 공유 객체가 있는데, 이름의 유무에 관계없이 모든 공유 객체는 식별자  $id_i$ 로 구분된다. 유명 공유 객체는 그룹웨어 응용이 식별할 수 있는 이름을 가지도록 함으로써 흐름 제어(flow control)이나 순번 소유(turn-taking)와 같은 제어 객체로 사용된다.  $name_i$ 는 공유 객체의 이름을 나타내고,  $u_i$ 는 공유 객체를 생성한 그룹 멤버를 뜻한다. GSA내의 공유 객체는 여러 그룹 멤버로부터 동시에 접근되기 때문에 접근을 조정하기 위해 동시성 제어가 필요하다.  $\Delta_i$ 는 공유 객체에 대해 현재 접근 권한을 가지고 있는 그룹 멤버를 나타낸다.  $st_i$ 는 공유 객체에 대한 상태 정보를 나타내는데  $st_i$ 는 그룹 멤버의 생성 연산에 의해 초기에 설정되고, 수정 연산에 의해 상태가 변화하게 된다.

#### 3.3 무명 공유 객체 집합

$$O^N = \{ o_i^N \mid o_i^N = (id_i, u_i, \Delta_i, st_i) \}$$

무명 공유 객체는 유명 공유 객체와는 달리 그룹웨어 응용에서 식별할 수 있는 이름이 존재하지 않는데, 무명 공유 객체는 주로 문자나 그래픽 데이터와 같이 제어가 아닌 데이터로서의 의미만을 가지는 객체에 해당된다.

### 3.4. 공유 객체 연산 집합

F={ Create, Delete, Update, Lock, Unlock)  
GSA내의 공유 객체에 대한 연산은 생성, 삭제, 수정, Lock, 그리고 Unlock 연산이 있다. 그룹 멤버는 공유 객체의 생성을 임의로 수행할 수 있으나, 생성된 공유 객체에 대한 삭제나 수정 연산을 위해서는 공유 객체 접근을 위한 Lock 연산을 먼저 수행해야 한다. Lock 연산 후에는 반드시 Unlock 연산이 뒤따른다.

### 3.5 그룹 멤버 통신

공유 객체 연산과는 상관없이 그룹 멤버 사이의 정보 교환을 위한 통신 기능이 제공이 필요하다. 그룹 멤버 통신 C에는 GSA에 참여하는 모든 그룹 멤버들에게 메시지를 보내는 방송(broadcast) 형태와 전체 그룹 멤버 집합의 부분 집합에 속하는 그룹 멤버들에게 메시지를 보내는 형태가 있다.

## 4. 그룹 공유 영역의 구조

공유 영역의 구현은 공유 객체의 처리 형태에 따라 세 가지 구조로 분류할 수 있다 [1,5-6]. 중앙 집중형 구조(centralized structure)는 그룹웨어 사용자들 사이에 오직 하나의 응용만이 존재하는데, 모든 그룹웨어 사용자들의 입력은 하나의 응용에 집중된 후 출력은 각자의 화면에 분산되는 방식이다. 복제형 구조(replicated structure)는 공유 자원들이 각 응용 프로그램에 복제되는 구조이고, 하이브리드 구조(hybrid structure)는 중앙 집중형 구조와 복제형 구조의 절충 구조이다.

중앙 집중형 구조는 일반적으로 서버-클라이언트 형태를 가지는데 공유 객체에 대한 조작의 동기화와 접근 제어를 위한 프로토콜이 비교적 쉬운 반면, 네트워크 상의 병목 현상과 고장 시의 문제점이 단점이다. 복제형 구조는 고장 시 복구 능력이 상대적으로 뛰어나지만, 공유 객체에 대한 동기화와 동시성 제어가 어렵고 네트워크 환경 하에서 주요 성능 비교가 되는 호스트(host) 사이의 연결 수가 많아진다 는 점이 단점이다.

그룹 공유 영역은 서버 형태를 가지는 SAS(Shared Area Server)와 클라이언트 인터페이스인 SACI(Shared Area Client Interface)로

구성되는 하이브리드 구조 형태를 가진다. 하이브리드 구조는 중앙 집중형 구조의 장점인 제어의 단순화와 오류 허용이 가능한 복제 구조의 장점을 가지고 있으므로 실시간 그룹웨어 응용의 요구를 충족시키는 구조이다. SAS는 공유 객체에 대한 전역 복제(global copy)를 가지는데, 클라이언트의 공유 객체 연산 요구를 수행하고 동시성 제어 및 동기화를 담당한다. 또한 SAS는 그룹 멤버의 관리와 그룹 멤버 간 통신을 관리한다. SACI는 SAS에 저장된 공유 객체에 대한 지역 복제(local copy)를 가지고 있고[그림1], 그룹 공유 영역 접근을 위한 공통의 인터페이스를 포함한다. SACI는 그룹웨어 응용으로부터 공유 객체에 대한 연산 요청을 SAS에 전달하고 SAS로부터의 연산에 대한 응답을 응용에 전달하는 역할을 가진다.

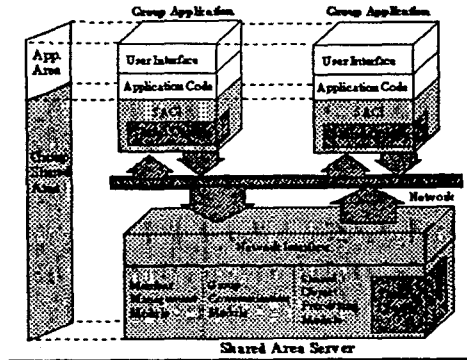


그림 1. 그룹 공유 영역

연산 요청과 응답은 SAS와 SACI 사이에 비동기 메시지 전달(asynchronous message passing) 방식으로 수행되는데, 메시지의 구조는 다음과 같다.

### 4.1 메시지 구조

msgID	ownID	msgSize	wParam	lParam	msg
-------	-------	---------	--------	--------	-----

표 1 SAS 메시지 구조

- msgID : 메시지 식별자
- ownID : 메시지 소유자
- msgSize : 메시지 내용의 크기
- wParam : 메시지 인자(16bit)
- lParam : 메시지 인자(32bit)

모든 메시지는 msgID 필드(field)에 의해 구별되는데, msgID 필드는 공유 객체의 연산과 그룹 멤버 사이의 통신에 대해서 정의되어 있

다. 예로 SACI로부터 SAS로의 공유 객체 연산 요청에 관련한 msgID 필드 값은 다음의 표 2와 같다.

공유 객체 연산	msgID
공유 객체 생성 요청	SM_REGISTER_OBJECT
공유 객체 삭제 요청	SM_DELETE_OBJECT
공유 객체 수정 요청	SM_UPDATE_OBJECT
공유 객체 lock 요청	SM_LOCK_OBJECT
공유 객체 unlock 요청	SM_UNLOCK_OBJECT

표 2 공유객체 연산 관련 msgID 필드의 내용

모든 메시지는 메시지를 보내는 그룹 멤버의 식별자를 포함하는데 ownID 필드에 저장된다. msgSize 필드는 msg 필드의 크기를 나타내고, wParam 필드와 lParam 필드는 인자(parameter)로써 msgID에 따라 그 의미가 달라진다. msg 필드 또한 msgID 필드에 따라 해석이 달라지는데, 예로 msgID 필드가 공유 객체의 생성에 관한 메시지인 SM\_REGISTER\_OBJECT 라면 msg 필드는 그룹웨어 응용이 생성하고자 하는 공유 객체의 내용이 되고, 공유 객체에 대한 접근을 요청하는 SM\_LOCK\_OBJECT 메시지의 경우 msg 필드는 값을 갖지 않으면서 wParam 필드는 공유 객체에 대한 식별자 값을 가지게 된다.

#### 4.2 그룹 공유 영역 서버

그룹 공유 영역 서버 SAS의 내부는 크게 5가지의 처리 모듈이 있는데, 각각 Network Event Handler(NEH), Thread Manager(TM), Callback Manager(CM), Member Manager(MM), Shared Object Manager(SOM)이다[그림2].

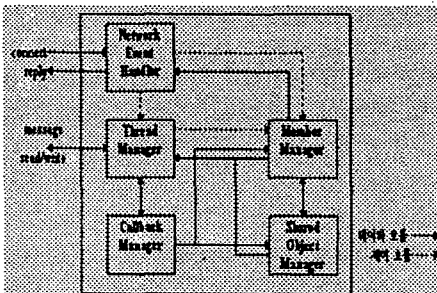


그림 2 SAS의 구조

##### 4.2.1 Network Event Handler

그룹 멤버로부터의 초기 연결 요청을 처리하는 부분이다. Network Event Handler 는 별도의 하나의 스레드(Thread)로 운용되는데,

SAS 초기 실행 시 생성되고 TCP/IP 관련 초기화 과정을 수행한 후, 외부로부터의 연결 요청을 기다린다. 일단 그룹 멤버로부터 연결 요청이 들어오면 메시지를 주고받을 수 있는 전용 포트를 연 후 Thread Manager에게 포트 정보와 함께 제어를 넘긴다. 그룹 멤버의 연결 요청에 대한 응답으로 Network Event Handler는 Member Manager로부터 멤버 식별자를 전달받은 후 이를 그룹 멤버에게 전송한다.

##### 4.2.2 Thread Manager

SAS와 연결된 그룹 멤버와 메시지를 주고받는 스레드를 생성, 관리하는 관리자로서, 그룹 멤버 하나에 대해 하나의 스레드를 생성한다. 그룹 멤버로부터 연결 요청이 이루어지면 Network Event Handler로부터 받은 포트 정보를 가지고 하나의 스레드를 생성한다. 생성된 스레드는 포트를 통해 그룹 멤버와 메시지를 주고받게 된다. 그룹 멤버로부터 받은 메시지는 처리를 위해 Callback Manager로 전달된다.

##### 4.2.3 Callback Manager

Thread Manager에서 받은 메시지는 처리를 위해 Callback Manager로 전달된다. Callback Manager는 메시지의 종류에 따라 처리할 콜백 함수들의 집합을 유지하고 있는데, Thread Manager로부터 받은 메시지는 Callback Manager에 의해 호출되는 콜백 함수에서 처리된다. 호출되는 콜백 함수는 메시지에 따라서 Member Manager, Shared Object Manager 그리고 Thread Manager와 연관된다.

##### 4.2.4 Member Manager

SAS에 연결된 그룹 멤버들에 대한 정보를 유지하는 관리자로서 멤버의 생성, 삭제 등의 연산을 수행한다. Member Manager는 멤버에 대한 정보를 유지하는 테이블을 유지하는데, 멤버와 관련한 메시지는 Callback Manager에 의해 Member Manager로 전달된다.

##### 4.2.5 Shared Object Manager

SAS에 연결된 그룹 멤버들이 생성한 공유 객체를 유지하는 관리자로서 공유 객체의 생성,

삭제, 수정과 같은 연산을 수행한다. Shared Object Manager는 공유 객체에 대한 정보를 유지하는 테이블을 관리한다.

#### 4.3 그룹 공유 영역 클라이언트 인터페이스

그룹 공유 영역을 이용하는 그룹웨어 응용들은 그룹 공유 영역 클라이언트 인터페이스인 SACI를 통해 공유 객체에 접근한다. SACI는 SAS에서 제공하는 각종 공유 객체에 대한 연산 요청에 대한 API를 포함하고, SAS로부터의 응답을 포함한 모든 형태의 메시지를 그룹웨어 응용에게 전달하는 함수들을 가지고 있다.

또한 SACI는 SAS에서 제공하는 그룹 멤버 간 통신을 지원하는 함수를 가지고 있는데, 이들 SACI 함수들은 내부적으로 SAS와 비동기 메시지 교환을 통하여 그룹웨어 응용과는 콜백 함수 형태로 교환하게 된다.

SAS와 SACI사이에는 네트워크 상의 메시지 전달 형태를 지니고, SACI와 그룹웨어 응용 사이에는 그룹웨어 응용이 호출할 수 있는 형태의 함수 형태나 윈도우(Window) 기반 시스템의 윈도우 메시지 전달 형태의 2 가지 방법으로 정보 전달이 이루어진다.

#### 4.4 동시성 제어

분산 컴퓨팅 환경에서의 동시성 제어 문제는 주로 락(lock) 기법과 낙관적 접근 방법(optimistic approach)으로 해결해 왔다[10]. 락 기법은 하나의 트랜잭션(transaction)이 일정 시간 동안 공유 객체에 대한 접근 권한을 획득하는 방법으로 다른 트랜잭션의 공유 객체에 대한 동시 접근을 허용하지 않기 때문에 동시성 문제의 해결이 될 수 있다. 락 기법은 낙관적 접근 방법에 비해 상대적으로 구현하기가 쉽지만 네트워크 대역폭에 따라 실행이 느릴 수 있다는 단점이 있다.

낙관적 접근 방법은 공유 객체에 대해 접근을 얻기 위해서 락을 요청한 후 허락을 기다리는 락 기법의 단점을 보완하기 위한 방법으로, 일단 공유 객체에 대해 접근한 후 수행된 접근 결과를 허락 받기 위해 요청하는 방법이다. 이후 요청에 대한 응답이 접근을 허용하면 그대로 진행되지만, 만약 응답이 접근을 허용하지 않으면 접근 요청 이전 상태로 복귀하도록 하는 방법이다. 낙관적 접근 방법은 공유 객체에 대한 접근 요청이 빈번하지 않은 환경에서 주

로 사용되지만, 락 기법에 비해 상대적으로 구현하기가 어렵고 접근 이전 상태로 돌아가는 것이 불가능한 상황이 존재한다는 점이 단점이다.

동시성 제어를 위해 실시간 그룹웨어 응용에 낙관적 접근 방법을 적용하는 데에는 무리가 따를 수 있는데, 이는 사용자 인터페이스와 관련되어 있다. 예로 그룹 공동 편집기에서 그룹 사용자가 하나의 그래픽 객체에 대한 위치를 이동시키고자 한다면, 낙관적 접근 방법에 따라 사용자의 위치 이동 행위는 바로 실행된다. 그러나 락 요청에 대한 응답이 그래픽 객체에 대한 접근을 허용하지 않는다면, 그 그래픽 객체는 이동되기 전 위치로 다시 복귀해야 하고 이와 같은 상황은 그룹 사용자로 하여금 혼란을 일으킬 수 있다. 즉, 그룹 사용자의 관점에서 보면 원래 위치로 복귀되는 그래픽 객체가 자신의 행위에 대한 결과인지, 아니면 다른 그룹 사용자의 행위 결과인지에 대해 모호해 질 수 있다.

락 기법과 낙관적 접근 방법을 접목한 방식이 발표되어 있다[2]. Greenberg는 락 기법을 적용시키는 데에 있어서 락을 비 낙관적 락 방식(non-optimistic locking), 전 낙관적 락 방식(fully-optimistic locking), 반 낙관적 락 방식(semi-optimistic locking)으로 구분하였다. 구분의 기준은 락 요청에 대한 응답의 대기 여부와 여러 공유 객체에 대한 접근이 동시에 가능한 가 여부에 따라 결정된다. 비 낙관적 락 방식은 일반적인 락 기법과 같고, 전 낙관적 락 방식과 반 낙관적 락 방식은 공유 객체에 대한 접근을 하면서 그 공유 객체에 대한 접근 결과를 요청하는 것이 아니고, 공유 객체에 대한 락 요청을 한다는 점이 일반적인 낙관적 접근 방식과 구별된다. 이를 위해서 임시 락을 각 그룹웨어 응용마다 가지고 있다.

그룹 공유 영역의 동시성 제어는 기본적으로 락 기법을 사용하는데, 네트워크 상의 지연에 의한 락 기법의 단점을 줄인다. 모든 클라이언트는 서버에 존재하는 공유 객체의 복제를 가지고 있으므로, 각 공유 객체의 접근 권한 여부도 모든 클라이언트에 복제되도록 한다. 이는 비동기(asynchronous) 메시지 전달에 의해 해결될 수 있다.

하나의 공유 객체가 SAS에 생성되면 모든 클라이언트들은 똑 같은 복제를 비동기 메시지

전달을 통해 가지게 된다. SAS는 자신이 관리하는 모든 공유 객체에 대해 접근 권한 정보를 가지고 있다. 이 접근 권한 정보는 공유 객체의 락 여부, 락을 요청한 클라이언트 식별자 그리고 락을 요청한 시점의 타임스탬프(timestamp)를 포함한다. 하나의 클라이언트가 SAS에 생성된 공유 객체에 대한 접근을 위한 락 요청을 먼저 SACL에 요청한다. SACL은 자신이 유지하는 지역 공유 객체(local shared object)에 락이 걸려 있는지에 따라 다음과 같은 행동을 취한다.

지역 공유 객체가 이미 다른 클라이언트에 의해 락이 걸려 있을 경우:

락을 요청한 클라이언트에게 접근할 수 없음을 즉시 알린다.

지역 공유 객체에 락이 걸려 있지 않은 경우 :

SAS에 해당 공유 객체에 대한 락을 요청한다.

SAS로부터의 모든 메시지는 비동기 방식으로 전달되기 때문에 락 요청에 대한 SAS로부터의 응답은 SACL에 비동기 메시지로 전달된다. SACL은 락 응답을 클라이언트에게 전달하게 되고 클라이언트는 일단 락 요청을 한 후 응답을 받기 전까지 다른 행동을 수행할 수 있다.

자신의 락 요청뿐만이 아니라 다른 클라이언트의 락 요청에 대한 응답도 SACL이 받기 때문에 클라이언트는 자신의 SACL에서 유지하는 공유 객체의 락 여부를 통해 전체 그룹 공유 공간상의 공유 객체에 대한 락 여부를 알 수 있다.

이미 락이 걸려 있는 공유 객체에 대해 빈번한 락 요청이 클라이언트로부터 SAS에 집중될 경우 네트워크 교통량은 많아질 수밖에 없다. 즉 클라이언트 차원의 동시성 제어는 서버 차원의 동시성 제어에 있어서, 네트워크 상의 집중화 현상을 감소시킨다.

## 5. SAS의 용례

그룹 공유 영역을 이용한 실시간 그룹웨어 응용 4 가지가 현재 개발 완료되었거나 개발 중이다. 이들 응용 중 객체 지향 그래픽 편집기 CoShed와 픽셀 지향 프리젠테이션(presentation) 응용인 전자칠판은 공동 작업 인식 방식을 따르는 응용들이고, DisplayShare는 그룹 사용자 사이의 화면을 공유하는 프로그램으로 공동 작업 투명성 방식으로 구현된 응용이다. 이 외에도 서로간의 동화상과 음성

을 교환하는 영상회의 시스템도 그룹 공유 영역을 이용한다. 이 절에서는 특히 CoShed와 관련하여 논의될 수 있는 점에서 대해 논하였다.

### 5.1 객체 지향 그래픽 편집기 : CoShed

그룹 공동 편집기인 CoShed는 객체 지향의 그래픽 편집기로써 다양한 그래픽 객체를 그룹 사용자들이 생성, 수정할 수 있도록 한 응용 프로그램이다. CoShed에서 다루는 데이터의 종류는 원, 직선 등과 같은 그래픽 객체와 이미지 객체를 포함한다. 이들 객체에 가할 수 있는 연산에는 생성, 삭제, 수정 연산이 있고 각 객체는 사용자가 움직이거나 크기를 조정할 수 있다.

CoShed에서 객체의 이동이나 크기 조정과 같이 객체의 속성을 변경시키는 연산을 수행하기 위해서는 락킹이 필요한데, 해당 객체에 대한 락을 SAS에 요청하고 SAS로부터 허가를 받으면 속성을 변경시킬 수 있다.

그래픽 편집기의 특성상 사용자는 다수의 객체를 한번에 선택할 수 있는데(객체들의 이동 등과 같은 예), 선택된 각 객체마다 락을 요청하기보다는 SAS의 다중 락 기능을 이용하여 한번의 메시지로 다수의 객체에 대한 락을 요청할 수 있다.

그룹웨어 인식을 제공하기 위해서 CoShed에서는 텔리포인팅(telepointing) 기능을 제공하는데, 이는 다른 사용자의 마우스 동작을 자신의 화면에 보이도록 하는 기능이다. 텔리포인팅 기능을 제공하기 위해서 SAS의 방송 기능을 사용한다. 즉 한 사용자의 마우스 동작에 의한 이벤트 발생은 (x, y) 좌표 값으로 나타낼 수 있는데, 연속적인 (x,y) 좌표 값을 SAS에게 전송하면 SAS는 다른 사용자들에게 좌표 값을 방송하게 된다. 텔리포인팅과 같은 기능은 기존 연구에서 그룹웨어를 위한 기본 기능으로 제공하고 있지만, SAS에서는 이 기능을 기본 기능으로 보기보다는 그룹웨어 응용에 종속적인 기능으로 간주하여 제공하지 않는다.

## 6. 앞으로의 연구 방향 및 결론

본 논문에서는 실시간 그룹웨어 응용을 지원 하는 구조로 SAS와 SACL를 제안하였다. 공유 객체를 중앙에서 관리하는 서버인 SAS와 그룹웨어 응용에게 공유 객체에 대한 인터페이스를

제공하는 SACI로 구성된 그룹 공유 영역 하이브리드 구조는 그룹웨어 응용과 공유 객체의 처리를 분리하였다는 관점에서 기존 중앙 집중형 구성과 분산 복제형 구성과 차이가 있다.

그룹 공유 영역 서버 SAS는 전역 공유 객체의 동시성 제어와 순차화 처리를 통해 각 그룹웨어 응용에서 일어날 수 있는 불일치 문제를 해결한다. 공유 영역 클라이언트 인터페이스 SACI는 그룹웨어 응용으로부터 공유 객체의 처리를 분리하고, 공유 객체 접근을 위한 인터페이스를 그룹웨어 응용에게 제공함으로써 응용 프로그래머의 자유도를 높였다.

또한 공유 객체의 형태를 그룹웨어 응용 프로그래머가 정의할 수 있도록 함으로써 특정 그룹웨어 응용 분야에서 요구하는 다양한 종류의 데이터형을 제공하도록 하였다.

하지만 SAS와 SACI로 구성된 하이브리드 구조가 모든 그룹웨어 응용 분야에서 나올 수 있는 모든 요구들을 다 만족할 수는 없다. 차후 우선적으로 연구해야 할 방향은 SAS 에서 관리하는 공유 객체들의 그룹화(grouping) 문제이다. 현재 SAS에서 관리하는 공유 객체들은 서로간이 종속적이지 않고 독립적이다. 공유 객체 사이의 종속성 관계에 관한 연구 분야는 그룹웨어 응용 분야에서 충분히 활용할 수 있는 분야이다. 그룹화 문제는 락 요청을 위한 단위 크기를 나타내는 공유 객체의 granularity 문제와도 연관이 된다. 공유 객체 사이의 종속성에 의해 그룹화 또는 계층화된 구조를 만들 수 있다면 락킹과 같은 동시성 제어 문제를 수월히 해결할 수 있다고 본다.

4. Bentley, R., Rodden, T., Sawyer, P., and Sommerville., I. An architecture for tailoring cooperative multi-user displays. Proceedings of CSCW'92, ACM, Toronto, 1992, pp. 187-194
5. Newman-Wolfe, R.E., Webb, M.L., and Montes, M. Implicit Locking in the Ensemble Concurrent Object-Oriented Graphics Editor. Proceedings of CSCW'92, ACM, Toronto, 1992, pp.265-272
6. Computer-Supported Cooperative Work. Computer, 1993
7. Karsenty, A., and Beaudouin, M. An Algorithm for Distributed Groupware Applications. Proceedings of the 13th International Conference on Distributed Computing Systems, Pittsburgh, 1993. pp, 195-202
8. Sohlenkamp, M., and Chwelos, G. Integrating Communication, Cooperation, and Awareness: The DIVA Virtual Office Environment. Proceedings of CSCW'94, ACM, ,1994, pp. 331-343
9. Greenberg, S., and Marwood, D. Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface. Proceedings of CSCW'94, ACM, 1994, pp. 207-217
10. Mary, Q., and College, W. Distributed systems : Concepts and Design. Addison-wesley Publishing., 1994

#### [참고 문헌]

1. Ellis, C.A., Gibb, S.J., and Rein, G.L. Groupware: Some issues and experiences. Communications of the ACM. 34,1, 1991, pp. 38-58
2. Roseman, M., and Greenberg, S. GROUPKIT A Groupware Toolkit for Building Real-Time Conferencing Applications. Proceedings of CSCW'92, ACM, Toronto, 1992, pp. 43-50
3. Shen, H., and Dewan, P. Access Control for collaborative environments. Proceedings of CSCW'92, ACM, Toronto, 1992, pp. 51-58