

개방형 분산처리 환경에서의 다자간 대화형 응용 서비스 및 다중 세션 관리 서비스 설계

이 증 화†

요 약

본 논문의 목적은 여러 다른 유형의 정보를 사용하고 다자간에 공동작업을 지원하는 응용의 요구사항을 분석하고, 이러한 특성을 갖는 응용이 필요로 하는 유용한 공통 서비스를 제공함으로써 개발 환경을 지원하는 데 있다. 응용 계층의 수평 서비스로서 다자간 대화형 응용 서비스와 기반 구조 계층의 지원 서비스로서 다중 세션 관리 서비스가 정의되었으며, 개방형 분산 처리 시스템의 개념과 규칙에 따라 설계되었다. 따라서, 응용 개발자들은 이 두 서비스를 이용하여 본 논문에서 고려한 특성을 갖는 응용을 구축할 수 있을 것이다.

Design of Cooperative Interactive Application Service and Session Management Service in ODP Systems

Jong-Hwa Yi†

ABSTRACT

The objective of this paper is to analyze requirements of distributed applications which use the different type of information and support the cooperative work environment between a group of persons in interactive mode, and to provide useful common services for the development of this kind of applications. For this, *Interactive Cooperative Application Service* as a horizontal service on the application layer and *Session Management Service* as a support service on the infrastructure layer were defined and designed using the concepts and rules of the Open Distributed Processing system. Application designers may use these services for building those applications which demand the requirements we have characterized in this paper.

1. 서 론

정보통신의 빠른 발전은 다양한 응용개발을 가능하게 하였는데, 그 중 사용자 관점에서 볼 때 파급효과가 큰 몇가지 예를 들자면 여러 사용자간의 그룹 활동을 가능하게 하는 공동 작업 환경 조성, 사용자간에 여러 다양한 유형의 정보 교환 가능, 여러 형태로의 상호동

작과 통신 서비스 지원등을 꼽을 수 있다. 이렇듯 정보통신 기술에 힘입어 가능한 신 기술을 이용하려는 다양한 응용들의 개발도 지속적인 발전을 해왔다.

각 응용을 구성하는 요소들 즉, 사용자를 포함하여 정보, 구조 (OS, 입출력 디바이스, 통신 채널등) 혹은 이용하고자 하는 다른 응용등은 일반적으로 분산되어 있다. 이러한 환경에서의 응용개발은 자원의 분산화로 인하여 발생하는 복잡한 문제점을 해결해야만 한다. 또한, 개발자들은 여러 다른 업체가 개발한 플랫폼을 비롯

†정 화 원 : 한국전자통신연구원 멀티미디어표준연구실
논문접수 : 1997년 10월 16일, 심사완료 : 1998년 2월 23일

하여 네트워크, 통신 프로토콜, 자원 액세스 방법, 정보의 처리방식, 정보 교환방식, 프로그램 언어등 매우 다른 여러 분야의 기술 요소와 개발 도구를 접하면서 이를 사용하기 위하여 많은 시간과 노력을 소비하고 있다.

분산환경이 안고 있는 많은 기술적 문제점과 개발의 복잡성을 해결해 줄 수 있는 공통의 기반 구조 (common infrastructure) 정의에 대한 연구가 산업체와 대학을 중심으로 활발히 진행되었다. 공통의 기반 구조는 먼저 이종의 시스템간의 상호 운용성과 응용의 이식성을 보장하고, 분산 투명성을 제공하는 기능을 지원하며, 시스템이나 응용 개발에 필요한 다양한 서비스를 제공하는 데 목적이 있다. 이러한 기반 구조를 정의하는 데 있어서 각 기관은 자신들의 목적, 기술 및 이권에 따라 연구를 진행시킴으로써 각 기관이 정의한 구조간에 상이함이 존재하나, 각 기관의 최종 목표는 시스템 및 응용 개발을 위한 분산 처리 구조를 제공하는 데 있다고 하겠다.

통합된 분산 구조 정립을 위하여 국제 표준화 기구인 ISO/IEC JTC1/SC21/WG7와 ITU-T SG VII Question 19를 중심으로 표준화 활동이 전개되어, 현재 개방형 분산 처리 기본 참조 모델 (RM-ODP: Reference Model for Open Distributed Processing)이 정의되어 있다[1][2]. 4 개의 문서로 구성되어 있는 기본 참조 모델은 앞서 설명한 하나의 공통 기반구조를 정의하고 있다. 이 기본 참조 모델에서 정의하고 있는 지원 기능중에 Trader와 Type Repository에 대한 표준화가 부분적으로 진행되는 상태이나 사실상 이 분야의 표준화 작업은 마무리 되었다고 볼 수 있다.

국제 표준화 활동과 더불어 산업체에서도 많은 연구 프로젝트를 수행하면서 산업체 표준안 (*de facto standards*)이 정의되었다. 우선적으로 APM (Architecture Projects Management) 기관에 의해 발표된 ANSA (Advanced Networked Systems Architecture)를 들 수 있는데[4], 이 구조가 정의되기까지의 연구 결과물이 RM-ODP에 반영되었으며, 따라서 두 구조에서 사용하는 기본 개념 및 규칙은 동일하다고 볼 수 있다. ANSA구조의 기본 기능을 개발한 플랫폼인 ANSAware는 대학교나 연구기관에 제공되어 이 분야에서 시스템 및 응용 개발에 사용되었다[3]. RM-ODP는 분산 구조의 기본 골격을 정의하고 있고, 어떤 특정분야의 시스템이나 응용을 위한 기본 참조 모

델은 RM-ODP를 기본으로 하여 세분화 할 필요가 있다. 이러한 맥락에서 차세대 정보 통신망 구조를 정립하여 새로운 통신 서비스를 제공하기 위하여 TINA (Telecommunication Information Networking Architecture)가 발표되었다[8][12]. TINA는 통신 서비스, 망 자원 관리의 실체로부터 개발 및 구축에 관련되는 기본 구조와 원칙을 제시함으로써, 효율적인 망 관리와 신규 서비스를 제공하고자 TINA 컨소시엄에서 발표한 분산 처리 구조이다.

OMG(Object Management Group)그룹은 객체 지향 기법을 도입하여 CORBA (Common Object Request Broker Architecture)라는 분산 처리 구조를 정의하였다[5][6][10]. CORBA는 위의 구조들이 사용하고 있는 기본 개념 및 규칙과는 상이함이 존재하나, 분산 환경에서의 상호 운용성과 응용의 이식성을 보장하며, 응용 개발에 필요한 기능 및 도구를 제공하고 있다. 위에서 열거한 기관들은 개념 및 모델 정립을 우선 순위로 두고 이 분야의 표준안을 진행시켰으나, OMG그룹은 현존하는 기술들을 분석, 연구 및 플랫폼 개발을 통해 상용적으로 널리 활용될 수 있도록 기술과 정보 보급에 목적을 두고 표준 작업을 실행시켰다. 또한 표준화 활동 및 상용화에 많은 산업체들이 동참할 수 있는 기반조성을 함으로써 현재 이 분야에 영향력 있는 기관으로 자리잡고 있다[13].

이밖에, OSF그룹이 정의한 DCE (Distributed Computing Environment)가 존재한다[7][11]. 이 분산 구조는 기존에 정의되어 사용되고 있는 표준들중에 분산 환경을 구성하기 위하여 필요한 표준들을 선택하여 정의하였다. DCE는 분산 시간 서비스, 디렉토리 서비스, 보안 서비스, RPC, 쓰레드, 시스템 운용체제와 통신 서비스를 제공한다.

앞서 언급한 바와 같이 공통의 기반 구조의 중요한 기능중에 하나는 시스템이나 응용 개발 환경을 제공하는 것이다. 개발 가능한 응용의 종류를 언급하기는 사실상 불가능한데, 이것은 각 사용자 및 개발자의 특별한 용도에 따라 응용의 특성이 매우 달라지며 다양하다는 것을 의미한다. 따라서 응용의 관점에서는 자신들의 다양한 필요를 이해하고 이러한 요구사항을 만족시킬 수 있는 기술적 기능들을 지원해 줄 수 있는 기반 구조가 요구된다. 즉, 각 응용은 자신의 목적과 용도에 따라 기반구조가 제공하는 기능들을 사용함으로써 효율적이면서 또한 개발자의 노력을 최대한으로 줄이는 방법으로

구축되어야 한다.

유용한 개발 환경을 지원하는 한가지 방법은 여러 다양한 응용들이 사용할 수 있는 공통의 응용 서비스와 이러한 응용 서비스 수행을 가능하게 하는 지원 서비스를 정의하여 제공하는 것이다. 이를 위해서 가장 먼저 수행되어야 할 과제는 다양한 응용들이 나타내는 특성과 논리적인 기능을 정확히 파악하고 그것을 바탕으로 응용들이 무엇을 요구하는지 또한 이를 만족시킬 수 있는 유용한 기능과 서비스가 무엇인지를 분석해야 한다.

다양한 여러 분야의 응용중에 본 논문에서는 여러 사용자들간에 멀티미디어 정보를 이용하여 대화형식으로 상호동작을 지원하는 응용들을 고려한다. 대상 응용은 이미 잘 정의되어 있는 MMCF (Multimedia Communication Forum)에서 발표한 MDC (Multimedia Desktop Collaboration) 와 ITU-T T.124 표준인 GCC (Generic Conference Control) 를 반영하였다[15][16]. 이 분야에 속하는 응용을 대상으로 그들의 요구사항을 분석하고 이들이 공통적으로 필요로하는 응용 서비스를 개방형 분산처리 구조상에서 설계 및 개발한다. 또한, 공통의 응용 서비스가 구축될 때 이들이 기반 구조 계층에게 어떠한 요구사항을 나타내는지 분석하고, 이에 대응하는 지원 서비스로서 다중 세션 관리 서비스를 정의한다. 기본적으로 이 서비스는 다자간 다중 연결 제어 및 관리 기능, 여러 형태로 멀티미디어 정보 전송등을 지원한다.

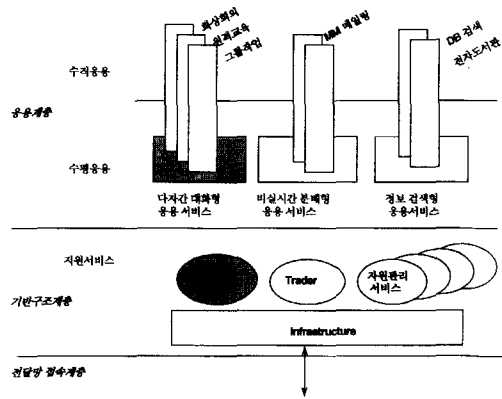
2. 분산 개발 환경

앞서 언급한 분산 환경을 일컫는 의미에서 분산 처리 구조, 시스템과 응용 개발에 필요한 기능을 제공하는 차원에서 지원 구조 (support architecture)나 지원 환경 (support environment) 혹은 상위의 응용 계층과 하위의 통신 계층사이에서 개발 환경에 필요한 기능을 지원한다는 의미에서 미들웨어라는 용어가 사용되고 있다. [1][17]에서는 하나의 분산 처리 구조가 지원해야 할 기능들을 언급하고 있다. 예를 들면, 유용성, 신뢰성, 고장 허용성, 분산 투명성, 시스템간의 연합 (federation), 시스템 보안, 시스템 관리, 개발 도구등이다. 물론, 이 두 문서는 이 분야의 정식 표준안은 아니지만, 여러 분산 구조의 적합성을 검증할 수 있는 가이드라인을 제시한다.

분산 환경이 제공해야 할 기능은 기반 구조를 비롯

하여 정보 저장, 분산 처리, 통신 프로토콜과 망, 개발 도구등 다양한 차원에서 지원되어야 한다. 시스템이나 응용의 관점에서 볼 때 또 다른 중요한 기능중에 하나는 시스템이나 응용 개발에 필요한 개발 도구와 유용한 공통의 응용 서비스를 지원하는 것이다. 현재 정의되어 있는 분산 구조들도 이러한 목적을 달성하기 위해 필요한 기능과 서비스를 제공하고 있으나, 여러 분야에서의 다양한 응용들의 요구사항을 충족시키기에는 미흡한 실정이다. 이것은 기반 구조상에서 다양한 응용 개발이 활발히 이루어져 이들의 요구사항을 새로이 도출시켜야 하며, 분산 구조는 이들의 요구사항을 이해하고 이를 만족시킬 수 있는 효과적인 서비스를 정의 및 추가시켜야 한다는 것을 의미한다.

본 절에서는 그림 2.1에서 나타나는 바와 같이 응용 개발에 필요한 분산 개발 환경을 정의한다. 크게 응용 계층, 기반 구조 계층과 전달망 계층으로 구성될 수 있는데, 각 계층에 대하여 간략하게 기술하면 다음과 같다.



(그림 2.1) 분산 구조 및 구성요소
(Fig. 2.1) Distributed Architecture and its Components

● 응용계층

기반구조가 제공하는 서비스를 이용하는 응용들이 위치하는 계층으로 수직 응용 (Vertical Applications) 과 수평 응용 (Horizontal Applications)으로 나눌 수 있다. 수직 응용이란 사용자가 개발하고자 하는 각각의 논리적인 응용을 의미하며, 이러한 수직 응용이 공통적으로 필요로 하는 서비스를 도출하여 정의함을 수평 응용이라 일컫는다. 따라서, 유용한 수평 응용을 정의함으로써 동일한 특성을 갖는 수직 응용 개발을 용

이하세 할 수 있다.

다양한 응용을 구분하는 방법에는 여러가지가 있지만, 여러 사용자간에 이루어지는 상호동작 방식이 대화형인지 분배형인지 아니면 검색형인지에 따라 개괄적으로 다음과 같이 3수평 응용 서비스로 분류할 수 있다[15].

○ **다자간 대화형 응용** : 두 명 이상의 사용자에게 실시간으로 멀티미디어 정보를 교환할 수 있는 공동의 작업 공간을 제공한다. 다자간에 공동 작업 공간을 이용하여 서로 상호 동작을 지원하는 응용들이 이에 해당하며, 구체적으로 개발 가능한 수직 응용의 종류로는 화상 회의, 원격 교육, 대화형 원격 사평, 원격 공동 진료 (collaborative diagnosis), 공동 디자인 (collaborative design) 등이다.

○ **비 실시간 분배형 응용** : 사용자들이 멀티미디어 화일이나 메시지를 비 실시간으로 분배할 수 있도록 해준다. Store-and-forward 방식으로 한 사용자가 멀티미디어 정보를 전송하거나 수송할 수 있다. 멀티미디어 메일링, 일정한 장소에 저장되어 있는 멀티미디어 정보 (예: presentation 혹은 OHP files) 분배등을 이용하는 수직 응용들이 이 종류에 해당한다.

○ **정보 검색형 응용** : 사용자들이 다양한 정보 제공자들에게 원하는 정보를 요청하고 그에 대한 응답으로 정보를 제공받을 수 있도록 해준다. Request-and-Reply 방식을 이용하여 비 실시간 멀티미디어 정보 검색을 목적으로 하는 수직 응용(예: 전자 데이터베이스 검색, 전자 카탈로그, 전자 도서관 등)들이 이에 해당된다.

● **기반 구조 계층**

이형 시스템간의 상호 운용성, 응용의 이식성, 분산 투명성을 통합적으로 지원할 수 있는 공통의 기반 구조에 해당하며, 이밖에 개발 환경을 지원하기 위하여 요구되는 서비스나 개발 도구 (예: IDL 컴파일러, 서비스 인터페이스) 가 이 계층에서 제공된다. 또한, 기반 구조를 이용하여 개발되어 상호 동작하는 응용이 필요한 기능을 효과적으로 지원하기 위하여 여러 다양한 응용들의 요구사항을 분석하여 이를 충족시킬 수 있는 기능을 도출하여 제공하는 것도 이 계층의 중요한 목표라고 할 수 있다.

분산 환경에서 해결해야 하는 중요한 사항중에 하나

는 자원의 분산성에 따른 여러 문제점을 응용 개발자들이 고려하지 않도록 내부적인 기술 사항들을 차단해 주고 응용에 따라 필요한 투명성을 지원할 수 있어야 한다. 개방형 분산처리 기본 모델에서는 접근, 위치, 중복, 동시성, 이전, 실패, 자원, 연합, 그룹 투명성을 정의하고 있으며, 각 응용은 기반 구조에게 자신이 필요한 투명성 기능을 선택적으로 요구하여 지원 받게 된다.

공통의 기반 구조 모델을 정립함에 있어 재사용성, 확장성, 새로운 서비스 수용성등의 기술적 이점을 얻기 위하여 객체 지향 기법을 도입하는 추세를 볼 수 있다. 객체 지향 기반 구조상에서 사용자 응용뿐만 아니라 기반 구조가 제공하는 모든 서비스 응용들이 객체 지향 설계 및 프로그램 기법으로 구축되어 수행되도록 통합된 객체 지향 분산 환경을 제공하게 된다.

● **전달망 접속 계층**

시스템이나 응용의 분산 수행에 필요한 수송 계층과 전달망 계층의 통신 서비스를 제공한다. 기본적으로 상위 계층인 기반 구조는 이러한 통신구조와는 무관하게 위치하나, 점점 다양해지는 응용이 요구하는 서비스를 제공하기 위하여 전달망 접속 계층의 효율적인 통신 서비스와 프로토콜을 필요로 한다. 따라서, 기존의 통신 서비스이외에 다중 연결 관리, 신속한 멀티미디어 정보 전송, 서비스 품질 보장등 새로운 서비스를 제공할 수 있는 수송 계층 서비스와 차세대 정보 통신망이 요구된다.

3. 다자간 대화형 응용 서비스

3.1 전체 기술

두 명 이상의 사용자에게 서로의 정보를 대화형 방식으로 교환할 수 있는 공동 작업 환경을 제공하고자 하는 응용들을 고려한다. 이러한 특성을 갖는 응용들의 요구사항을 도출하고 필요한 서비스를 정의하기 위하여 기본적으로 MMCF에서 발표한 MDC와 ITU-T T.124 표준안인 GCC를 참고하였다.

MMCF는 통신사업자, 서비스 제공자, 멀티미디어 응용 및 장비 개발자, 사용자등이 모여 통신망상의 멀티미디어 기술 지원과 업체간의 상호 호환성 있는 제품 개발을 촉진하기 위해 설립된 비영리 조직이다. 다양한 멀티미디어 통신 및 응용 서비스를 지원하기 위하여 기반 구조, 응용 공통 서비스, 하위 수송 계층, 서비스 품질 제공을 위한 QoS 요구사항, 보안등의 기술적 사항을 중점적으로 다루고 있으며, 이를 지원하는 기반 구

조를 제시하고자 MMCF 참조 모델을 정의하였다. 이 참조 모델은 많은 부분에서 수정 및 보완되어야 할 기술 사항을 남겨 놓고 있으나, 현재 이 기관의 활동은 추진되지 않고 있는 상황이다. 기본적으로 MMCF가 고려하고 있는 수평 응용 분야는 데스크탑 회의 서비스, 멀티미디어 메시징과 메일 그리고 정보 검색 서비스이다. 이 중 본 논문에서는 멀티미디어 정보를 이용하여 공동 작업을 지원하는 데스크탑 회의 서비스를 대상으로 한다[14][15].

ITU-T T.124 표준안은 일반 회의의 제어에 필요한 서비스를 정의한다. 회의 시작과 종료, 회의 진행 절차, 참가자들의 회의 가담 제어, 회의에 관련되는 정보 관리를 위한 서비스를 비롯하여 회의 시간 연장, 잔여 회의 시간 광고등의 부가적인 서비스를 제공한다[16].

위의 두 표준안에서 정의하는 기능과 서비스를 기초로하여 본 논문에서는 다자간에 멀티미디어 정보를 이용하여 공동작업을 수행할 수 있는 가상 작업 공간(Virtual Working Space)을 지원하는 응용의 요구사항을 분석하고, 하나의 수평 서비스로써 *다자간 대화형 응용 서비스*를 정의하고자 한다.

3.2 요구사항 도출

객체 지향 분석 기법 OMT (Object-Oriented Modelling and Design) 가 정의하는 3가지 모델인 객체, 동적 그리고 기능 모델을 사용하여 다자간 대화

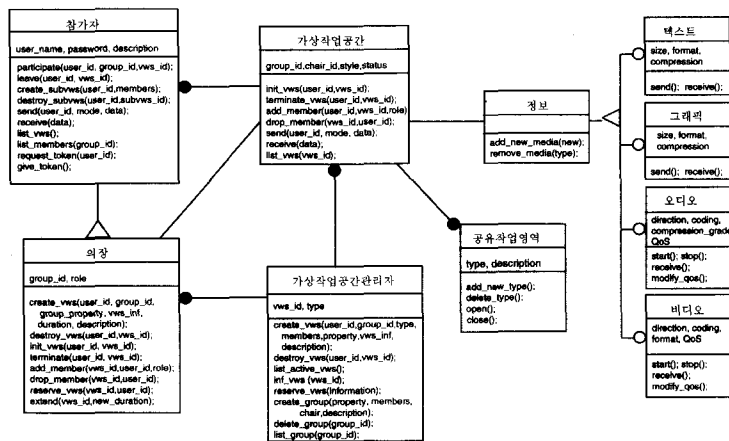
형 응용의 요구사항을 도출하였다. 이 중 본 논문에서는 객체 모델 (그림 3.1)을 기술한다.

1. 참가자

- 가상작업공간에 참여하는 일반 참여자를 의미하며, 현재 진행중인 혹은 예약된 가상작업공간들의 정보(예 : 가상작업공간 성격, 현 상태, 참가자 명단, 의장 명단, 참가하는 그룹 정보, 가상작업공간 목적등)를 포괄적으로 검색할 수 있어야 한다.
- 진행중인 가상작업공간에 참가하고, 원하는 때에 이탈 할 수 있어야 한다. 참가자가 이미 등록되어 인증된 경우와 의장의 입시 허락에 의해 어느 특정 가상작업공간에 참여하는 청강자 경우가 있다. 후자인 경우에는 가상작업공간 진행중에 동적으로 참여(예 : 발원권, 정보 전송 요청등)는 불가능하나, 진행 상황등을 관찰할 수 있다.
- 일반 참가자는 발언권을 신청하여 할당받은 경우에 발언권을 갖게 되며, 이때 정보는 모든 참여자들에게 전달되어야 한다.
- 필요에 따라 보조 가상작업공간 개시를 요청하고, 참가 및 이탈할 수 있다.

2. 의장

- 참가자 객체 기능외에 가상작업공간 생성, 종료, 예약, 연장등의 관리를 한다.
- 생성된 가상작업공간 시작 및 끝남을 요청한다.



(그림 3.1) 다자간 대화형 응용 요구사항
(Fig. 3.1) Requirements of Cooperative Interactive Applications

이 경우는 이미 생성된 가상작업공간을 종료하지 않고 어느 시간 이후 다시 진행할 경우 사용 가능하다.

- 가상작업공간에 새로운 참가자를 추가하거나 기존 참가자를 축출할 수 있다.

3. 가상작업공간

- 가상작업공간 생성시에 모든 참여자들이 공유하는 공동작업영역 (예 : 오디오 윈도우, 비디오 윈도우) 이 생성되며, 각 공동작업영역에 대한 참여자 입력을 제어하는 방식 (예 : 경쟁방식, 라운드 로빈 방식, 의장이 지정하는 방식등)을 결정하며, 이 제어 방식은 가상작업공간 진행중에도 변경가능하다.
- 하나의 가상작업공간내의 일부 참여자들만이 참여하는 보조 가상작업공간을 갖을 수 있다.
- 의장이 모든 참여자에 대해 설정하는 다양한 기능에 대한 접근 권한에 따라 각 참여자들의 참여 방식을 제어한다.
- 참여자나 의장은 자율적으로 가상작업공간에 가담 및 이탈할 수 있고, 이에따른 변경사항이 참여자들에게 통보되어야 하며, 참여자나 다른 사용자가 요구할 때 제공되어야 한다.
- 가상작업공간에 대한 정보 (예 : 이름, 설명, 의장, 참가자 명단, 가상작업공간 스타일, 현 상태 등)를 동적으로 관리한다.

4. 가상작업공간 관리자

- 의장의 요청에 따라 가상작업공간을 혹은 의장이나 참여자의 요청에 의해 보조 가상작업공간을 생성 및 종료한다. 보조 가상작업공간은 기본적으로 일반 가상작업공간의 기능을 갖게 되며, 해당 가상작업공간이 종료될때 혹은 요청이 있을 경우 종료된다.
- 가상작업공간들에 대한 정보를 총괄적으로 저장, 관리 및 제공한다.
- 예약된 가상작업공간 정보를 관리하고, 임의의 사용자, 참가자 및 의장에게 제공할 수 있어야 한다.
- 새로운 그룹을 생성, 제거 및 이미 설정된 그룹 특성이나 규칙을 변경할 수 있도록 한다. 그룹 생성시 인증된 사용자 명단, 의장, 그룹 멤버에게

적용될 규칙등이 정해진다. 이미 생성된 그룹 식별자를 이용하여 원하는 시간에 가상작업공간을 예약할 수 있다.

5. 공유작업영역

- 가상작업공간이 생성될때 사용할 공유작업영역이 생성되며, 동일한 가상작업공간에 참여하는 모든 참가자들이 공유하는 작업영역 (윈도우)를 의미하며, 공유작업영역에 입력되는 내용은 모든 참여자들의 화면에 동기방식으로 전송되어야 한다.
- 가능한 공유작업영역 유형은 비디오, 오디오, 화이트 보드, 편집, 원격응용 및 공유응용이 있다.
- 각 공유작업영역에 연결된 참여자들의 상태를 관리해야 한다.
- 참가자들은 공유작업지원 기능을 사용하여 공유작업영역을 통하여 상호 동작하게 되며, 각각의 공유작업영역 특성에 따라 지원 기능들을 선택하여 이용할 수 있다.
- 각 공유작업지원 기능은 이것을 사용하고자 하는 공유작업영역에서 설정한 스타일에 따라 제어된다.
- 가능한 공유작업지원 기능으로는 뷰, 포인터, 마커, 키 입력, 화일 업로드, 윈도우 이동, 윈도우 크기 조절, 스냅샷, 기록, 재생 등이다.

6. 정보

- 임의의 가상작업공간에 가담하는 참여자들이 오디오, 비디오, 텍스트, 그래픽등 여러 다른 미디어 정보를 사용하여 상호동작한다.
- 각 미디어 사용에 필요한 기능 (예 : 읽기, 저장, 송신, 수신등)을 제공해야 한다.

3.3 서비스 인터페이스 설계

위에서 분석된 다자간 대화형 응용 서비스 요구사항을 만족시킬 수 있는 공통의 응용 서비스를 개방형 분산처리 시스템 기본 참조 모델에서 정의하고 있는 여러 관점을 이용하여 설계하였다. 그 중 본 논문에서는 각 수행 객체가 제공하는 인터페이스 및 연산을 정의하는 수행 모델 (그림 3.2) 을 기술한다.

각 수행객체가 제공하는 서비스 인터페이스와 연산은 IDL를 이용하여 정의되었으며[9], 그 중 의장과 가상작업공간 객체에 대한 IDL 표기는 표 3.1과 같다.

<표 3.1> 의장 및 가상작업공간 객체 서비스 인페이스
<Table 3.1> Service Interface of Chair and VWS Objects

```
// 의장 객체 서비스 인터페이스
interface Chair_Interface {
typedef enum mode {unicast, multicast, broadcast};
void create_vws (in string user_id, in unsigned long group_id,
                in string group_property, in string vws inf, in
                short duration,
                in string description, out unsigned long
                vws_id);
void destroy_vws (in string user_id, in unsigned long vws_id);
void init_vws (in string user_id, in unsigned long vws_id, out
               string result);
void terminate_vws (in string user_id, in unsigned long vws_id);
void add_member (in string user_id, in unsigned long vws_id, in
                string role, out string result);
void drop_member (in string user_id, in unsigned long vws_id);
void reserve_vws (in string user_id, in unsigned long group_id,
                  in string group_property, in unsigned long
                  time, out string result);
void extend_vws (in unsigned long vws_id, in short
                 new_duration, out string result);
};
```

```
// 가상작업공간 객체 서비스 인터페이스
interface Virtual_Working_Space {
void init_vws (in string user_id, in unsigned long vws_id, out
               string result);
void terminate_vws (in string user_id, in unsigned long vws_id,
                    out string result);
void add_member(in string user_id, in unsigned long vws_id, in
                string role, out string result);
void drop_member (in string user_id, in unsigned long vws_id,
                  out string result);
void send (in string user_id, in short mode, in char* data);
void receive (in char* buffer);
void list_vws (in unsigned long vws_id, out List information);
};
```

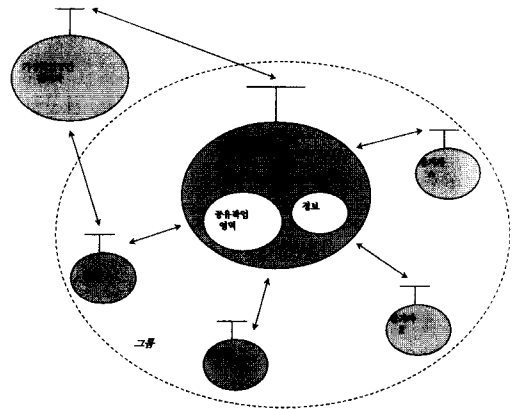
4. 다중 세션 관리 서비스

4.1 전체 기술

다자간 대화형 응용 서비스는 여러 참가자가 하나의 가상작업공간을 통하여 설정된 규칙에 따라 서로의 정보를 교환하며 공동작업 수행을 지원하는 기능을 제공한다. 이런 특성을 갖는 응용이 필요로 하는 기능/서비스를 효과적으로 제공하기 위해서 기반 구조 계층 지원 서비스의 하나로 다중 세션 관리 서비스를 정의한다. 이때 세션이란 하나의 가상작업공간을 조정하기 위하여 이에 가담하는 여러 참석자들을 하나의 공동 그룹으로 묶는 것을 의미한다. 이 서비스는 다자간 다중 연결 제어 및 관리 기능, 여러 형태의 멀티미디어 정보 전송동

을 지원한다.

다중 세션 관리 서비스는 개괄적으로 세션을 생성하고 종료하며, 세션에 관한 전반적인 제어 및 정보관리를 담당하는 *세션관리자*, 생성된 세션에 참가자를 추가 혹은 삭제하며 참가자 정보를 동적으로 관리하는 *세션* 그리고 참가자간의 연결 설정과 해제, 정보전송의 기능을 담당하는 *채널*로 이루어진다.



(그림 3.2) 다자간 대화형 응용 서비스 수행 모델
(Fig. 3.2) Computational Model of Cooperative Interactive Application Service

세션이 생성된후 허락된 참가자들이 가담하기 시작하며, 세션진행중이라도 새로이 참가요청 및 이탈할 수 있다. 진행중인 세션은 요청에 의해 혹은 참가자 전원이 이탈한 경우 종료된다.

진행중인 세션에 대하여 일시중지 및 재시작을 요청할 수 있다. 이 경우는 설정된 세션을 종료하지 않고 일정한 시간이후 다시 진행할 경우 유용하다. 이때, 가담하던 모든 참가자들의 정보를 관리해야 한다. 별도로 진행중인 두개의 세션을 하나로 통합 요청할 수 있다. 이때 두 세션의 참여자들은 새로운 세션으로 옮겨진다. 이와 반대로 하나의 세션을 두개로 분리 요청할 수도 있다. 각각의 세션으로 참가자가 나뉘어 할당된다.

참가자간의 다중 채널을 설정하여 서로 통신 가능하게 한다. 서로 교환하기 원하는 정보 유형과 원하는 서비스 품질에 따라 적절한 채널을 참가자간에 연결한다. 새로운 참가자가 추가될때마다 혹은 기존의 참가자가 이탈한 경우마다 채널에 연결을 추가 및 삭제해야 한다. 연결된 채널을 통해 여러 다양한 방법으로 멀티미

더어 정보를 주고 받는다.

4.2 요구사항 도출

그림 4.1에서 다중 세션 관리 서비스 요구사항을 나타내며, 각 객체의 기능은 다음과 같다.

1. 세션관리자

- 다자간 대화형 응용의 요청에 따라 세션을 생성 또는 종료한다.
- 생성된 세션들의 총괄적인 정보(예 : 세션특성, 다른 세션과의 관계등)를 관리한다.
- 세션관리자에 의해 생성된 세션에 대해 일시중지 및 재시작을 요청할 수 있다. 이 경우는 이미 생성된 세션을 종료하지 않고 일정한 시간 이후 다시 진행할 경우 사용가능하다.
- 두개의 세션을 하나로 통합 혹은 하나의 세션을 두개로 분리할 수 있다.
- 세션은 요청에 의해 몇몇의 참가자를 대상으로 다른 세션을 생성할 수 있으며, 이때 두 세션간의 관계가 성립된다.
- 새로 생성되는 세션은 참가자 전원이 이탈한 경

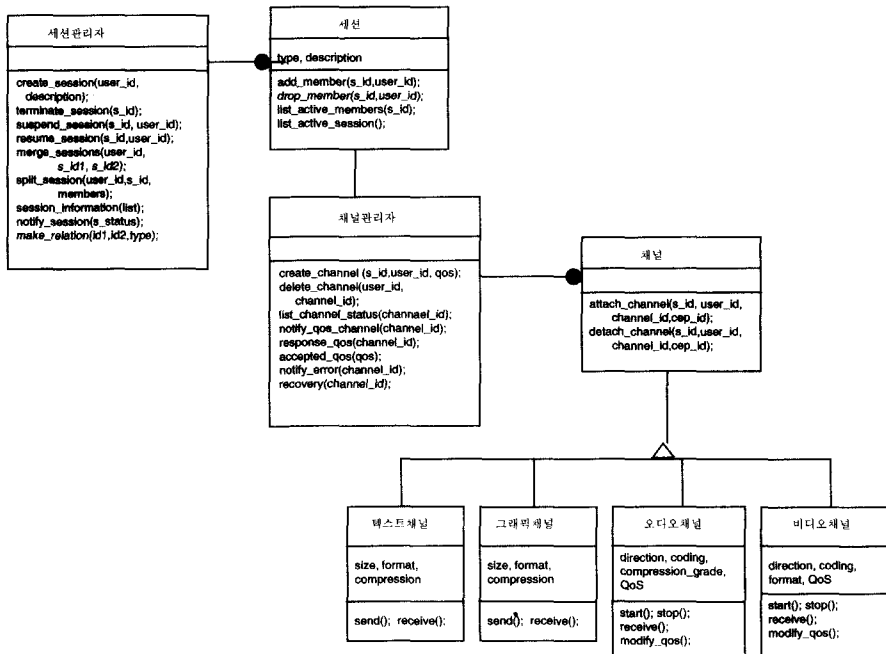
우, 요청에 의한 경우 그리고 해당 세션이 종료되는 경우 종료된다.

2. 세션

- 세션이 시작된 후 참가자들이 추가되며, 기존 참가자들이 이탈할 수 있다.
- 세션의 참가자들이 자율적으로 가담 및 이탈함에 따라 참가자 정보 (예: 참가자명단)를 동적으로 관리한다.
- 세션은 응용의 요청에 따라 채널을 설정하며, 하나의 세션은 전송하고자 하는 정보 유형에 따라 여러 채널을 갖을 수 있다.

3. 채널관리자

- 참가자간의 텍스트, 그래픽, 오디오 및 비디오 정보를 전송할 수 있는 채널을 생성 및 종료한다. 참가자간에 point-to-multipoint(unidirection) 채널을 설정한다.
- 채널 설정 과정에서 서비스 품질 값을 협상할 수 있다.



(그림 4.1) 다중 세션 관리 서비스 요구사항
(Fig. 4.1) Requirements of Session Management Service

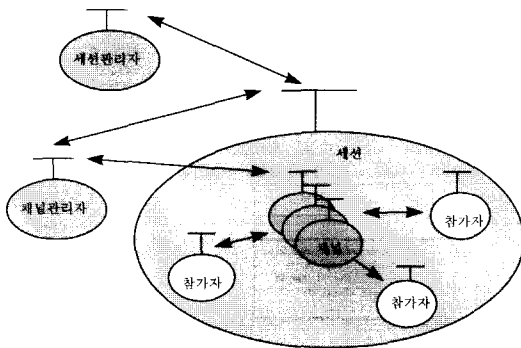
- 설정된 채널들에 대한 정보를 동적으로 관리 및 제공한다.

4. 채널

- 이미 생성된 채널에 새로운 멤버를 추가 혹은 기존의 멤버를 삭제할 수 있다.
- 오디오, 비디오등의 멀티미디어 정보를 여러 다른 전송방법으로 참가자들에게 전달할 수 있어야 한다. 가능한 전송 방법으로는 unicast(어느 특정한 참가자에게), multicast(몇몇의 참가자들에게) 또는 broadcast (모든 참가자들에게)가 있다.
- 각 채널에 대한 상태 정보를 관리한다.

4.3 서비스 인터페이스 설계

다자간 대화형 응용 서비스 설계와 동일한 방법으로 개방형 분산처리 시스템의 개념 및 규칙에 따라 다중 세션 관리 서비스도 설계되었다. 그림 4.2는 수행모형을 나타내며, 이 모델에서는 세션 관리자, 세션, 채널 관리자, 채널 객체와 각 객체가 제공하는 인터페이스 및 연산들이 IDL로 기술되었다. 이 중 표 4.1는 세션 관리자 객체가 제공하는 인터페이스를 보여준다.



(그림 4.2) 다중 세션 관리 서비스 수행 모델
(Fig. 4.2) Computational Model of Session Service Management

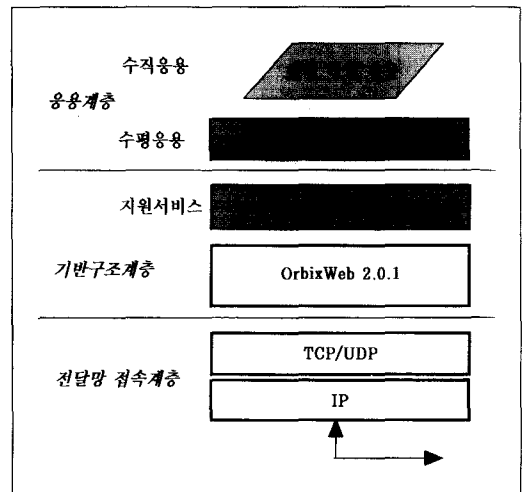
5. 구현 환경

그림 5.1은 소프트웨어 구조를 나타낸 것이다. 앞서 IDL로 기술된 다자간 대화형 응용 서비스와 다중 세션 관리 서비스는 각각 응용 계층의 수평 서비스와 기반 구조 계층의 지원 서비스로 제공된다. 이들 서비스를

<표 4.1> 세션관리 서비스 인터페이스
<Table 4.1> Interface of Session Manager

```
// 세션관리자 객체 서비스 인터페이스
interface Session_Manager{
typedef enum relation_type {independent, subsession, dependent_session,
group_session, subgroup&dependent,
subsession&group};

typedef struct {
unsigned long session_id;
enum status {opened, suspended, resumed, closed, no-members};
string description;
} Session_Inf;
typedef sequence <Session_Inf> Session_List;
void Create_Session (in string user_id, in string description, out unsigned
long session_id);
void Terminate_Session (in unsigned long session_id);
void Suspend_Session (in unsigned long session_id, in string user_id, out
string result);
void Resume_Session (in unsigned long session_id, in string user_id, out
string result);
void Merge_Sessions (in string user_id, in unsigned long session_id1, in
unsigned long session_id2, out unsigned long
new_session_id);
void Split_Session (in string user_id, in unsigned long session_id, in string
members, out unsigned long new_session_id1,
out unsigned long new_session_id2);
void Session_Information (out Session_List list);
void Notify_Session_Status (in Session_Inf session_status);
void Make_Relation (in unsigned long session_id1, in unsigned long
session_id2, out unsigned long relation_type);
};
```



(그림 5.1) 소프트웨어 구조
(Fig. 5.1) Software Architecture

이용하는 프로토타입 수직 응용으로는 원격 작업 응용으로 분산되어 위치하고 있는 3 사용자간에 상호 협동

작업을 지원한다.

현재 각 서비스에 대한 설계가 마무리 되고 구현 단계에 있다. 구현 시스템 사양은 Sparc Workstation (Solaris 2.5)과 Windows95를 탑재한 Pentium PC 2대를 사용하며, 다자간 응용 서비스는 JDK1.1.3과 IDL 컴파일러를 비롯하여 Orbixweb이 제공하는 개발 도구와 기능을 이용하고, 다중 세션 관리 서비스는 Orbix 2.0과 C++를 이용하여 현재 개발중이다.

6. 결 론

개방형 분산 처리 시스템은 객체지향 개념에 근거하여 공통의 기반 구조를 제시함으로써, 이 구조상에서 개발되는 응용 또한 객체지향 기법으로 설계되도록 조성함으로써 응용 서비스들이 모듈화되어 재사용이 용이하고, 개발되어 있는 모듈들을 이용하여 쉽고 경제적으로 다른 응용을 구축 또는 확장 할 수 있는 개발 환경을 지원한다.

앞서 언급한 응용 계층의 수평 서비스나 기반 구조 계층의 지원 서비스가 응용들에게 공통적으로 필요한 기능 및 서비스를 제공함으로써 결과적으로는 응용 서비스 개발을 촉진시킬 수 있을 것이다. 이를 위해서 실행해야 할 중요한 과제는 다양한 응용의 논리적 기능 파악을 바탕으로 그들의 요구사항을 정확히 분석하는 일이다. 이러한 맥락에서 본 논문에서는 현재 사용자들이 많은 관심을 나타내는 다자간에 대화형식으로 협동 작업을 가능하게 하는 응용들의 개발을 지원하기 위하여 다자간 대화형 응용 서비스와 다중 세션 관리 서비스를 정의 및 설계하였다. 각 서비스의 구조 및 인터페이스가 설계된 상태이며, 현재 구현 중에 있다.

이러한 방법으로 새로운 응용의 요구사항을 도출하고 유용한 지원 서비스를 제공, 관리함으로써 향후 계속적으로 출현할 각종 응용을 효과적으로 수용하고 체계적으로 지원할 수 있는 분산 구조 기반을 정립할 수 있을 것이다.

참 고 문 헌

- [1] ITU-T Rec. X.901 | ISO/IEC 10746-1, "ODP Reference Model Part1: Overview", DIS, May 1995.
- [2] ITU-T Rec. X.903 | ISO/IEC 10746-3, "ODP Reference Model Part3: Architecture", IS, February 1995.
- [3] APM Ltd., "ANSA: An Engineer's Introduction to the Architecture", Release TR.03.02., November 1989.
- [4] A.J. Herbert, "An ANSA Overview", IEEE Network, January/February 1994.
- [5] Object Management Group, "The Common Object Request Broker : Architecture and Specification", OMG Document Number 91.12.1. Revision 1.1, Draft 10. December 1991.
- [6] Object Management Group, "The Common Object Request Broker: Architecture and Specification", Revision 2.0. July 1995.
- [7] A. Beitz, P. King, K. Raymond, "Comparing two Distributed Environments: DCE and ANSAware", In: DCE - The OSF Distributed Environment: Client/Server Model and Beyond. A. Schill (ed.). Lecture Notes in Computer Science, No. 731. Springer-Verlag, 1993.
- [8] TINA-C, "Overall Concepts and Principles of TINA", Version 1.0, February 1995.
- [9] TINA-C, "TINA Object Definition Language (TINA-ODL) Manual", Version 1.3, June 1995.
- [10] Steve Vinoski, "CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments", IEEE Communications Magazine, February 1997.
- [11] The Open Systems Newsletter, "OSF's Distributed Computing Environment and the requirements", Vol.5 issue 9. September 1992.
- [12] 신영석, 오현주, 고병도, 김재근, "차세대 개방형 정보통신망 구조인 TINA 연구", 한국전자통신연구원 주간 기술동향, TIS 96 16 744, 1996. 5. 1.
- [13] 이종화, "국제 표준화 기구 현황-2 (OMG그룹)", TTA저널, 제51호, 1997.6.
- [14] MMCF, "Core services of MMCF Middleware", MMCF/95-005 Draft 4.0, June 1996.
- [15] MMCF, "Multimedia Desktop Collaboration", MMCF/95-002, Rev. 1.5, October 31, 1995.
- [16] ITU-T Recommendation T.124 (Draft), "Generic Conference Control", November 1994.

- [17] ISO/IEC JTC1 SC21 WG7. "Basic Reference Model of Open Distributed Processing. Part 1: Overview", Working draft. December 1991.



이 종 화

1987년 8월 칠레 산티아고대학
전산학과 졸업(학사)

1990년 2월 한양대학교 대학원
전자공학과 졸업(석사)

1990년~현재 멀티미디어표준연
구실 선임연구원

1996년 Technical University of Madrid, Spain
통신공학과 졸업(공학박사)

관심분야 : 개방형 분산처리 시스템, 분산응용구조, 멀
티미디어 응용, 객체지향 설계 기법