JACE : 인터넷 환경을 지원하는 신뢰성 있는 그룹통신 시스템

문 남 두† · 안 건 태† · 유 양 우† · 이 명 준††

요 약

인터넷의 성장과 함께 네트워크 응용서비스가 빠른 속도로 증대되고 있다. 이러한 응용서비스는 일시적으로 네트워크가 단절되거나, 또는 응용서비스가 실행되고 있는 호스트가 실패하더라도 투명하고도 안정적이며, 지속적으로 제공되는 것이 바람직하다. 이러한 요구사항을 만족시키기 위하여 다수의 *그룹통신 시스템*이 개발되어 왔는데 이러한 기존의 그룹통신 시스템은 단일 LAN 환경에 국한되거나 직접적으로 상호 연결된 LAN 환경으로 그룹통신의 범위를 제한하고 있다. 인터넷 환경에서 그룹통신을 지원하기에는 인터넷을 통한 통신이 현재까지는 너무 느리고 신뢰할 수 없었지만, 가까운 장래에 인터넷을 통한 통신이 그룹통신을 지원하기에 충분한 속도와 신뢰성을 제공할 수 있을 것으로 전망되고 있다. 본 논문에서는 인터넷 환경에서 *Extended Virtual Synchrony* 모델을 지원하는 그룹통신 시스템인 *JACE*(Java Advanced Communication Environments) 시스템의 설계와 구현에 대하여 기술한다. JACE 시스템은 기반통신 계층인 GC(Group Communication), 응용프로세스 그룹을 관리하는 RPGS(Reliable Process Group Service) 계층, 그리고 자바응용서비스와 클라이언트 개발을 위한 JACE API로 구성되어 있다. 개발된 JACE 시스템을 이용하여 *중복객체공간*(replicated object space)이 실험적으로 구현되었다.

JACE : A Reliable Group Communication System over the Internet

Nam-Doo Moon† · Geon-Tae Ahn† · Yang-Woo Yu† · Myung-Joon Lee††

ABSTRACT

Important network application services have been rapidly increased along with the growth of the Internet. So, it is desirable for such applications to serve transparently, continuously and safely even if the network is temporally disconnected or certain hosts running those services are crashed down. To satisfy such requirements, many *group communication systems* have been developed; but, those systems restrict its range within a single LAN or directly interconnected LAN environments since the communication through the Internet is too slow and too unreliable to support group communication. As of now, it is expected the Internet is going to be sufficiently fast and reliable in the near future to support group communication. In this paper, we present the design and implementation of a group communication system, named *JACE*(Java Advanced Communication Environment), supporting *Extended Virtual Synchrony* model over the Internet environment. The JACE system consists of three modules: GC(Group Communication) which is a basic communication layer, RPGS(Reliable Process Group Service) which manages application process groups, and JACE API for developing Java application services and clients programs. In addition, an experimental *replicated object space* is developed as an application of the JACE system.

※ 본 연구는 '99년도 정보통신연구진흥원 대학기초연구지원 사업 과제의 지원으로 수행되었음.

† 준 회 원 : 울산대학교 대학원 컴퓨터정보통신공학부
†† 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수

논문접수 : 1999년 10월 15일, 심사완료 : 1999년 11월 12일

1. 서 론

인터넷의 급속한 성장과 정보통신 관련 기술의 발전으로 오늘날 다양한 분야에서 네트워크 응용서비스 개발이 활발하게 이루어지고 있다. 이러한 응용서비스는 시간과 장소에 제한되지 않으면서도 안정적이고 지속적인 서비스를 제공하는 것이 바람직하다. 이러한 요구사항은 이기종 분산환경에서 동일한 서비스를 제공하는 응용서비스를 하나의 프로세스 그룹으로 동작될 수 있게 지원하고 일시적으로 호스트의 실패나 네트워크의 분할(partition)이 발생되더라도 그룹 구성원간에 일관성을 보장하는 *그룹통신 시스템*(group communication system)을 이용함으로써 해결될 수 있다.

그룹통신 시스템이란 응용프로세스가 그룹으로 동작될 수 있도록 지원하는 기반 시스템으로서, *VS*(Virtual Synchrony) 모델[1, 2]을 지원하는 Cornell 대학의 ISIS [3, 4]가 그 시작이 되었다. VS 모델은 상호 통신할 수 있는 그룹 구성원 사이에 일관된 멤버십과 메시지 전달을 보장하여 구성원간에 일관성을 유지한다. 그러나 VS 모델을 지원하는 그룹통신 시스템은 응용프로세스가 실행되고 있는 호스트가 실패하거나 일시적으로 네트워크의 분할이 발생되어 그룹이 상호 통신할 수 없는 여러 *구성요소*(component)로 분리된 후 재결합될 때 이들 사이에 일관성을 유지하지 못하는 문제점을 가지고 있다. 네트워크의 분할이 발생되면 계속적인 서비스를 제공하는 단일의 *주구성요소*(primary component)와 실행을 중지하는 나머지 *부구성요소*(non-primary components)로 구별된다. 여기서 구성요소란 상호 통신할 수 있는 호스트의 집합을 의미한다. 최근에는 VS 모델을 확장하여 그룹의 구성원이 실패 후, 재참여 하거나 분할된 네트워크가 재결합되더라도 투명하게 이전과 같이 단일의 그룹으로 동작될 수 있도록 그룹 구성원간의 일관성을 보장하는 *EVS*(Extended Virtual Synchrony) 모델[5]이 제안되었다. EVS 모델을 지원하는 시스템으로는 Totem[6], Transis[7], 그리고 Horus[8] 등이 있다. 그러나 기존의 개발된 그룹통신 시스템은 그룹통신의 범위를 단일 LAN(Local Area Network) 환경으로 국한하거나 직접적으로 상호 연결된 LAN 환경으로 그 범위를 제한하고 있다.

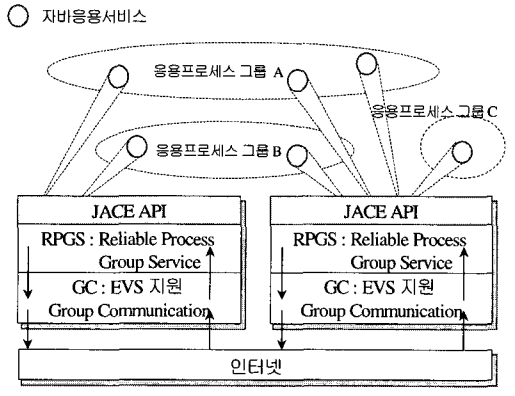
본 논문에서는 그룹통신의 범위를 제한하지 않고 인터넷 환경에서 그룹통신을 지원하는 *JACE*(Java Advanced Communication Environments) 시스템의 설계와 구현에 대하여 기술한다. 먼저 JACE 시스템의 기반 통신 계층으로서 EVS 모델을 지원하는 *GC*(Group Communication) 계층[9]의 구현에 대하여 설명한다. GC 계층에서는 그룹에 참여하는 응용프로세스가 위치한 호스트를 대상으로 *가상 링*(virtual ring)을 형성한다. 가상 링에 논리적인 *토큰*(token)을 순환시킴으로써, 하나의 가상 링에서 발생되는 메시지의 일련순서를 결정하고 동일한 순서로 메시지를 전달하여 구성원간에 일관성을 보장한다. 그룹별로 존재하는 가상 링은 지리적 위치에 제한되지 않으며 상호 통신할 수 있는 인터넷상의 호스트로 구성된다. GC 계층에서는 메시지의 순서화 기능과 가상 링에 참여하는 호스트의 멤버십 관리 기능을 제공한다. JACE 시스템의 *RPGS*(Reliable Process Group Service) 계층[10]은 각 응용프로세스 그룹 관리와 네트워크의 재결합으로 분할된 구성요소들이 하나의 구성요소로 합쳐질 때 동일한 프로세스 그룹에 참여하는 구성원들의 일관성을 유지하기 위하여 *상태전이*(state transfer) 기능과 프로세스 그룹 멤버십 관리 기능을 제공한다. JACE 시스템은 응용서비스의 가용성을 높이기 위하여 네트워크의 분할로 인하여 프로세스 그룹이 두 개 이상의 구성요소로 나뉘어지는 경우, 응용서비스의 특성에 따라 기존의 처리결과에 영향을 받아 다음 서비스를 처리해야 하는 *history-sensitive* 그룹과 기존의 처리결과에 영향을 받지 않고 다음 서비스를 처리할 수 있는 *history-free* 그룹으로 구분하여 실행될 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 JACE 시스템의 구조에 대하여 간략하게 기술하고, 3장에서는 EVS 모델을 지원하기 위한 JACE 시스템의 기반 통신 계층인 GC에 대하여 기술한다. 4장에서는 프로세스 그룹 관리를 위한 RPGS 계층에 대하여 기술한다. 다음으로 자바응용서비스 및 클라이언트 프로그램을 개발하기 위한 *JACE API*[11]에 대하여 5장에서 설명하고, 6장에서는 JACE 시스템을 이용하여 실험적으로 구현한 인터넷상의 *중복객체공간*에 대하여 설명한다. 중복객체공간은 인터넷 환경에서 자원의 효율적인 공유와 간편한 통신 수단을 제공한다. 마지막으로 7장에서 결론을 맺는다.

2. JACE 시스템

JACE 시스템은 인터넷 환경에서 실행되는 동일한

자바응용서비스를 그룹으로 동작될 수 있도록 지원하고 그룹 구성원간에 일관성을 유지하는 그룹통신 시스템이다. 2장에서는 JACE 시스템의 구조와 EVS 모델에 대하여 간략하게 살펴본다. JACE 시스템 구조는 (그림 1)에서 보는 바와 같이 크게 GC와 RPGS 그리고 API 모듈로 나뉘어진다.



(그림 1) JACE 시스템 구조

2.1 EVS(Extended Virtual Synchrony) 모델

JACE 시스템은 L. E. Moser에 의하여 정형화된 그룹통신 모델인 EVS[5]를 지원한다. EVS 모델은 ISIS 시스템의 VS 모델을 확장하여 분할된 모든 부분 구성요소에서 지속적인 실행을 지원한다. EVS 모델은 일시적으로 네트워크가 분할되고 재결합되는 상황, 응용프로세스가 실패하고 재참여하는 상황에서도 그룹 구성원간에 일관성 있는 메시지의 전달과 멤버십 관리를 보장하기 위하여 시스템의 상태를 두 가지 유형의 구성(Configuration) 상태로 정의한다. 이러한 구성 상태는 그룹통신 시스템이 실행되는 호스트의 멤버십과 식별자(unique identifier)로 표현된다.

(1) *정규구성*(Regular Configuration) 상태에서는 클라이언트의 서비스 요청 메시지나 응용서버의 응답 메시지와 같은 일반적인 메시지가 송수신 된다.

(2) *과도기구성*(Transitional Configuration) 상태는 네트워크의 단절로 인하여 그룹이 상호 통신할 수 없는 여러 분할영역으로 나뉘어지더라도 다음의 <표 1>에 기술된 메시지 전달방식을 올바르게 지원하기 위한 것이다. 과도기구성 상태가 포함하는 호스트 멤버십 정보는 이전의 정규구성 상태에서 새롭게 형성될 정규구성 상태로 옮겨가는 호스트 정보로 구성된다.

3. EVS를 지원하는 그룹통신 계층 : GC

그룹에 참여하는 구성원간에 일관성을 유지하기 위하여 메시지 순서화 기능과 멤버십 관리 기능이 요구된다. 이와 관련된 많은 연구가 진행되고 있는데 그 대표적인 방법으로는 중앙 집중적으로 그룹의 한 구성원이 멀티캐스트 메시지의 순서를 결정하고 멤버십을 관리하는 방법과 Totem과 RMP[12]와 같이 가상 링 구조를 기반으로 토큰을 순환시킴으로써 메시지의 순서를 결정하고 멤버십을 관리하는 방법이 있다. JACE 시스템은 보다 안정적이며 알고리즘 수행시 비교적 메시지 교환이 적은 링 기반 알고리즘을 사용하여 개발하였다.

JACE 시스템은 자신과 연결된 응용서버가 참여하는 그룹별로 GC 쓰레드(thread)를 생성하여 메시지의 순서화 기능과 호스트 멤버십 관리를 수행한다. 가상 링은 응용프로세스 그룹별로 생성되며 상호 통신할 수 있는 호스트로 구성된다.

3.1 메시지 순서화

그룹에 참여하는 응용프로세스는 메시지 전달의 신뢰성 및 순서보장의 요구사항에 따라 전달방식을 <표 1>과 같이 지정할 수 있다[5].

〈표 1〉 메시지의 전달 방식

전달방식	요구사항
Atomic delivery	임의의 한 구성원이 받은 메시지는 같은 그룹에 속한 모든 구성원에게 전송된다.
Causal delivery	임의의 메시지에 대한 응답 메시지가 결코 선행 메시지보다 먼저 전달되지 않는다.
Agreed delivery	그룹에 참여하는 모든 구성원에게 동일한 순서로 메시지가 전달된다.
Safe delivery	그룹에 참여하는 구성원이 실패하지 않는다면 메시지는 구성원 모두에게 전달된다. 네트워크 분할로 송신자와 분리되더라도 전달된다.

<표 1>의 전달방식을 만족시키기 위하여 일차적으로 전송 메시지에 대하여 순서화가 이루어진다. 메시지 순서화의 목적은, 동일한 그룹에 참여하고 상호 통신할 수 있는 호스트상의 GC로 구성되는 가상 링 위에 하나의 논리적인 토큰을 순환시키고 토큰을 받은

GC만이 자신의 호스트로부터 발생되는 메시지에 일련번호를 할당하여 멀티캐스트함으로써 일관된 전달 순서를 유지하는데 있다. 토큰은 최근 멀티캐스트된 메시지의 일련번호, 토큰 번호, 분실된 메시지의 재전송을 위한 재전송 요청 정보 등을 포함한다. 토큰을 전달받은 GC는 재전송 요청 정보를 검사하여 자신이 보유하고 있는 메시지를 재전송 한다.

3.2 호스트 멤버십 관리

GC 계층에서는 <표 1>의 메시지 전달방식을 지원하기 위하여 단일 가상 링 내에서 발생되는 메시지에 대하여 일련의 순서를 결정하고 가상 링에 참여하는 GC에 대하여 일관된 멤버십 관리를 할 수 있어야 한다. 토큰 분실은 호스트의 실패나 네트워크의 분할로 인하여 발생될 수 있다. 따라서 토큰분실이나 네트워크의 재결합이 발견되면 새로운 가상 링을 형성하기 위하여 호스트 멤버십 관리 알고리즘을 수행한다. 멤버십 관리의 목적은 새롭게 형성될 가상 링에 참여하는 모든 구성원들로부터 새로운 멤버십에 대하여 동의를 구하고 전산망의 결함으로 인하여 누락된 메시지들을 복원하는데 있다. 호스트 멤버십 관리 알고리즘은 새로운 가상 링을 형성하기 위하여 <표 2, 3>과 같은 특별한 유형의 메시지를 사용한다. <표 2>의 재구성 메시지는 일반 메시지와는 다르게 토큰을 갖고 있지 않더라도 멀티캐스트 될 수 있으며 분실되더라도 재전송되지 않는다. 재구성 메시지는 응용 프로세스에게 전달되지 않고 GC 계층에서 처리가 완료된다. 가상 링 식별자는 링을 대표하는 호스트 ID와 최대의 가상 링 번호로 구성된다.

〈표 2〉 재구성 메시지의 구조

정 보	설 명
Type	재구성 메시지를 나타낸다.
Sender_id	재구성 메시지를 전달한 송신자의 호스트 ID
Host_set	송신자가 고려하는 새로운 가상 링의 멤버십(호스트 ID들의 집합)
Ring_seq	송신자에게 알려진 가상 링 식별자들 중에서 최대의 가상 링 번호

<표 3>의 *구성 변경 메시지*는 가상 링을 구성하는 호스트의 멤버십에 관한 정보를 포함한다. 구성 변경 메시지는 정규구성 변경 메시지와 과도기구성 변경 메시지로 구분되며 이전의 정규구성에서 과도기구성으로의 변경이나 과도기구성에서 새로운 정규구성으로의 변경을 나타낸다. 구성 변경 메시지는 일반 메시지와 다르게 각 GC 계층에서 지역적으로 생성되어 멀티캐스트되지 않고 직접적으로 RPGS 계층에 전달되어 프로세스 멤버십 관리와 상태전이 알고리즘을 수행시킨다.

〈표 3〉 구성 변경 메시지의 구조

정 보	설 명
Regular_id	정규구성 메시지의 경우 : 새롭게 형성되는 구성요소의 정규구성 ID 과도기구성 메시지의 경우 : 이전의 정규구성 ID
Seq_number	정규구성 변경 메시지의 경우 : 0 과도기구성 변경 메시지의 경우 : 이전의 정규구성에서 전송된 메시지의 최대 일련번호
Transition_id	정규구성 변경 메시지의 경우 : 이전의 과도기구성 ID 과도기구성 메시지의 경우 : 새롭게 형성될 정규구성 직전의 과도기구성 ID
Membership	구성 변경 메시지가 개시하는 구성요소의 멤버십

GC는 아래와 같이 6개의 상태로 정의된다.

(1) 초기상태 : 처음 생성된 GC는 초기상태에 있으며 응용프로세스의 그룹참여 요청이나 그룹생성 요청을 *그룹위치정보 관리자*에게 전달한다.

(2) 표준상태 : 가상 링을 이루는 호스트 멤버십에 변화가 없는 상태를 말하며 이 상태에 있는 GC는 토큰을 받고 자신의 호스트에서 발생되는 메시지에 대하여 일련의 순서를 결정하여 멀티캐스트 한다.

(3) 수집상태 : 호스트의 실패나 일시적인 네트워크 분할로 인하여 토큰분실이 발견되면 새로운 가상 링을 형성하기 위하여 호스트 멤버십 정보를 수집한다.

(4) 교섭상태 : 새로운 가상 링에 대하여 멤버십의 동의를 시도한다.

(5) 합의상태 : 새로운 가상 링의 멤버십에 대하여 동의가 이루어진 상태이며 토큰이 순환하게 될 경로가 결정되고 손실된 메시지에 관한 정보가 교환된다. 토큰이 순환하는 경로는 우선적으로 가상 링의 대표자가 위치한 도메인 상의 호스트를 대상으로 먼저 경유하고 도메인 밖에 위치한 외부 참여자의 경우 호스트 식별자의 오름차순에 따라 가상 링을 순환하게 된다.

(6) 복원상태 : 손실된 메시지에 대하여 복원 작업을

수행한다.

3.3 그룹위치정보 관리자

그룹위치정보 관리자[13]는 그룹통신의 범위를 인터넷 환경으로 확장하기 위하여 요구되며 그룹에 참여하는 구성원이 실행되는 호스트 정보를 제공한다. 그룹위치정보 관리자는 각 응용프로세스 그룹에 대하여 <표 4>와 같은 정보를 관리한다.

〈표 4〉 그룹위치정보

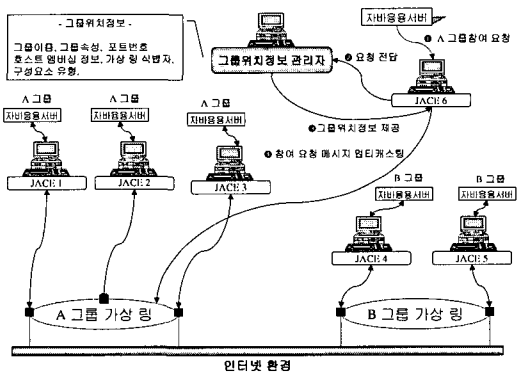
그룹위치정보	설 명
Name	응용프로세스가 참여하는 그룹이름을 나타낸다.
Creator	그룹을 생성한 응용프로세스의 ID와 호스트 위치정보를 포함한다.
Msg port	그룹별로 다르게 주어지며 메시지가 전달되는 통신 포트번호를 나타낸다.
Token port	그룹별로 다르게 주어지며 토큰이 전달되는 통신 포트번호를 나타낸다.
Component	주구성요소 또는 부구성요소로 지정된다.
Membership	가상 링에 참여하는 호스트 멤버십 정보를 포함한다.
Ring id	가상 링의 식별자(링을 대표하는 호스트 ID와 링 일련번호로 구성된다.)
Property	history-sensitive 그룹 또는 history-free 그룹으로 지정된다.

3.3.1 그룹생성 요청(Group Create Request)을 받는 경우

그룹위치정보 관리자가 그룹생성 요청을 받으면 요구된 그룹의 등록여부를 확인한다. 동일한 이름의 그룹이 이미 등록되어 있다면 그룹생성 요청은 무시되고 그룹생성을 요청한 송신자에게 이러한 사실을 알려주기 위하여 예외(exception)를 반환한다. 만일 요청한 그룹이 등록되어 있지 않다면 그룹위치정보 관리자는 그룹위치정보 저장소에 새로운 그룹정보를 등록한다. 등록되는 그룹위치정보는 그룹이름, 그룹 생성자, 구성요소 유형, 호스트 멤버십, 새로운 링 식별자, 생성된 그룹의 통신 포트번호, 그리고 그룹의 속성 등에 관한 정보가 있다.

3.3.2 그룹참여 요청(Group Join Request)을 받는 경우

그룹위치정보 관리자가 (그림 2)와 같이 그룹참여 요청을 받으면 우선 참여하고자 하는 그룹이 그룹위치정보 저장소에 등록되어 있는 지 검사한다. 등록되지 않았다면 그룹참여를 희망하는 송신자에게 관련된 그룹정보가 없다고 알려준다. 만일 요구된 그룹이 이미 등록되어 있다면 기존의 그룹이 사용하고 있는 통신포트번호, 기존 그룹의 호스트 멤버십, 그룹의 속성 및 그룹의 생성자에 관한 정보를 반환한다.



(그림 2) 그룹위치정보 관리자

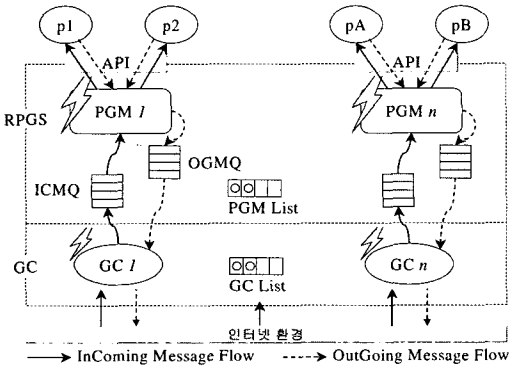
3.3.3 멤버십변경 요청(Membership Change Request)을 받는 경우

그룹위치정보 관리자가 멤버십변경 요청을 받으면 우선 변경하고자 하는 그룹이 그룹위치정보 저장소에 등록되어 있는지 검사한다. 등록여부를 검사할 때 그룹이름과 함께 가상 링의 식별자를 비교함으로써 멤버십변경 요청이 반복적으로 처리되지 않도록 한다. 그룹정보 저장소에 요구된 그룹이 등록되지 않았다면 새로운 그룹위치정보를 추가한다. 기존에 등록된 그룹이라면 멤버십변경 요청 메시지에 포함된 정보를 이용하여 그룹위치정보를 갱신한다.

4. 프로세스 그룹서비스 계층 : RPGS

RPGS 계층에서는 각 응용프로세스 그룹을 관리하기 위하여 프로세스 그룹관리자(Process Group Manager, PGM)를 두고 있다. (그림 3)에 보여준 두 개의 PGM은 자바응용서비스를 위한 프로세스 그룹 관리자이며 각각 자신의 그룹에서 발생되는 메시지에 대하여 상태전이 기능과 응용프로세스 멤버십 관리 등의 역할을 수행한다.

RPGS는 자바응용서비스의 프로세스 그룹 생성 요구에 대하여 하나의 PGM을 생성하고, PGM이 관리하



(그림 3) JACE 시스템의 상세구조

는 그룹 자바응용서비스 구성원 모두가 탈퇴하여 구성원이 없을 경우에 PGM은 제거된다. ICMQ(InComing Message Queue)는 GC로부터 순서가 결정된 메시지를 저장하는 큐로 사용된다. OGMQ(OutGoing Message Queue)는 응용서버로부터 전달받은 메시지를 임시보관하기 위한 큐로 사용된다. ICMQ와 OGMQ 큐는 단일 판독자와 기록자를 위한 동기화 큐로서 전형적인 생산자와 소비자 문제에 대한 알고리즘을 이용하여 구현하였다.

4.1 프로세스 그룹 관리자 : PGM

자바응용서비스는 RPGS와 연결을 성립한 이후 create() 메쏘드를 호출하여 프로세스 그룹을 생성할 수 있으며 응용프로세스 그룹을 관리하기 위한 PGM이 생성된다. PGM은 자바응용서비스를 위한 프로세스 그룹 관리자이며 각각 자신의 그룹에서 발생되는 메시지에 대하여 상태전이 기능과 응용프로세스 그룹의 멤버십 관리를 수행한다. 새로운 자바응용프로세스는 join() 메쏘드를 호출함으로써 기존의 그룹에 참여할 수 있으며, 프로세스 그룹에 참여했던 자바 응용서비스는 leave() 메쏘드를 호출함으로써 참여했던 그룹을 탈퇴할 수 있다. leave() 메쏘드는 join()과 함께 그룹의 멤버십을 변경시키는 기본적인 메쏘드로서 자바응용서비스가 join() 이나 leave()를 호출하였을 때 PGM은 프로세스 그룹의 멤버십을 변화시키고 그 사실을 그룹내의 모든 멤버에게 전달된다. 이와 같은 프로세스 멤버십을 변화시키는 메시지는 하부의 GC를 통하여 원격지에 있는 다른 서버들에게 신뢰성 있게 전달되므로 일관된 프로세스 멤버십을 유지할 수 있게 된다.

4.2 주구성요소(primary component)의 선택방법

응용프로세스 그룹은 네트워크의 분할이 발생될 때 상호 통신할 수 없는 두 개 이상의 구성요소로 나뉘어질 수 있다. 서로 다른 구성요소에 존재하는 동일 그룹의 구성원에게 서로 다른 순서로 메시지가 전달된다면 구성원 사이에 불일치가 발생되어 심각한 문제를 초래할 수 있다. 그룹 구성원간에 일관성을 유지하기 위하여 하나의 구성요소만이 메시지 최종 전달순서를 결정할 수 있다. 이와 같이 메시지의 순서를 결정할 수 있는 구성요소를 주구성요소라고 하며 나머지 구성요소들을 부구성요소라고 한다. 네트워크의 분할 이후 재결합이 발생되어 주구성요소와 부구성요소가 상호 통신 가능하게 되면 주구성요소에서 결정된 메시지의 순서에 따라 부구성요소에 존재하는 그룹의 구성원들에게 메시지가 전달되어 일관성을 유지하게 된다. 본 시스템에서는 기본적으로 그룹을 생성한 구성원이 참여하는 구성요소를 주구성요소로 선택한다.

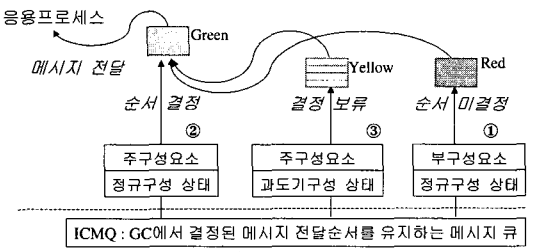
4.3 상태전이(State Transfer)

프로세스 그룹의 구성원간에 동일한 상태를 유지하기 위하여 구성원에게 전달되는 메시지의 순서는 항상 동일해야 한다. 3.1절에서 살펴본 바와 같이 GC에서 일차적으로 메시지 순서화가 이루어진다. PGM은 GC에서 결정된 순서를 기초로 하여 상태전이를 위한 메시지 순서화 작업을 수행한다. 상태전이를 위한 메시지의 순서화 작업은 네트워크의 분할과 재결합이 발생되어 동일한 그룹이 나뉘어지고 재결합되는 상황에서도 그룹의 모든 구성원에게 동일한 순서로 메시지를 전달할 수 있도록 지원한다. 본 시스템에서는 기본적인 상태전이를 지원하기 위하여 컬러 모델[14]을 채택하였다.

4.3.1 컬러(color) 모델

PGM은 (그림 4)의 컬러 모델을 이용하여 이차적인 메시지 순서화 작업을 수행한다. PGM은 컬러 모델을 적용한 메시지 큐를 가지고 있다. GC로부터 순서화된 메시지는 다음과 같이 세 가지의 색으로 분류되고 컬러 큐에 저장된다.

(1) red : PGM이 부구성요소에 속해 있을 때 전달된 메시지들은 red 메시지로서 전역순서가 결정될 수 없으므로 red 메시지를 위한 컬러 큐에 삽입된다 (①). red 메시지들은 전역순서가 결정되지 않았으



(그림 4) 상태전이를 위한 컬러 모델

므로 PGM이 관리하는 응용프로세스에게 전달되지 않는다.

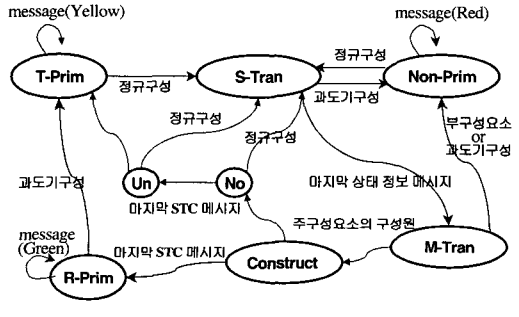
(2) green : PGM이 주구성요소에 속해 있으면서 정규구성 상태에서 메시지를 전달받은 경우 메시지는 green 메시지를 위한 컬러 큐에 저장된다(②). green 메시지들은 현재 PGM에 의해 전역순서가 결정되었으며 현재 PGM이 관리하는 응용프로세스들에게 전달된다.

(3) yellow : PGM이 주구성요소에 속해 있으면서도 과도기구성 상태에서 메시지를 받는 경우에는 yellow 메시지를 위한 컬러 큐에 저장된다(③).

네트워크의 분할로 인하여 각 PGM은 서로 다른 상태의 메시지를 보유하게 된다. 네트워크가 재결합되는 경우 각 PGM은 자신의 컬러 큐의 상태정보를 교환한다. 이러한 상태정보를 이용하여 상태전이가 수행된다.

4.3.2 컬러 모델을 적용한 PGM의 상태도

(그림 5)는 컬러 모델을 적용한 PGM의 상태도이다. (그림 5)에서 보는 바와 같이 PGM은 크게 여섯 개의 상태로 구분되며 각 상태의 의미와 알고리즘은 다음과 같다.



(그림 5) 컬러 모델을 적용한 PGM의 상태도

(1) Non-Prim 상태 : Non-Prim 상태는 PGM이 부구성요소에 속해 있는 경우이다. PGM이 S-Tran 상태나 M-Tran 상태에서 GC로부터 네트워크의 분할로 인하여 과도기구성 메시지가 전달되었을 때 이 상태로 전이되며 네트워크의 복구로 인한 정규구성 메시지가 전달되면 S-Tran 상태로 변경되고 상태전이를 시작하게 된다. 일반 메시지가 전달되었을 경우는 메시지의 전역순서를 결정할 수 없으므로 컬러 큐의 red 영역에 저장한다.

(2) R-Prim 상태 : PGM이 주구성요소에 속해 있으면서 네트워크의 구성이 정규구성인 경우이다. 전달되는 모든 일반 메시지에 대하여 PGM은 전역순서를 결정할 수 있으므로 green으로 마크하고 현재 PGM이 관리하는 응용프로세스들에게 전달한다. 과도기구성 메시지가 전달되면 네트워크의 분할이 발생한 것이므로 T-Prim 상태로 변경된다.

(3) T-Prim 상태 : PGM이 주구성요소에 속해 있으면서 네트워크의 구성이 과도기구성인 경우이다. GC는 네트워크의 분할 및 복구가 일어난 후 이전의 정규구성에서 전달하지 못한 일반 메시지를 전달하기 위하여 과도기구성 요소 변경 메시지를 전달한다. 전달되는 모든 일반 메시지는 이전의 정규구성에서 발생했지만 아직 PGM으로 전달되지 못한 메시지로서 모두 컬러 큐의 yellow 영역에 저장된다.

(4) S-Tran 상태 : PGM은 상태전이를 수행한다. 분산되어 있는 각 PGM은 자신의 상태정보를 서로 다른 PGM들에게 전달하고 PGM은 현재 구성요소 내의 모든 PGM에 대한 상태정보를 수집한다. PGM은 수집된 상태정보를 바탕으로 가장 많은 green 메시지를 보유하고 있는 PGM을 선택한다. 선택된 PGM은 재전송될 필요가 있는 green 메시지를 전송 후 M-Tran 상태로 변경된다.

(5) M-Tran 상태 : 이전상태인 S-Tran 상태에서 수집한 상태정보를 바탕으로 실제 메시지를 재전송하는 상태이다. PGM이 받는 일반 메시지는 다른 PGM으로부터 재전송 되는 메시지로서 이 메시지를 재전송 한 PGM의 상태정보 메시지에 따라 색깔 정보를 결정한다. 그리고 각 서버는 자신이 재전송할 순서인지 여부를 결정하고 차례로 메시지를 재전송 한다. 재전송이 끝나면 자신의 상태정보를 갱신하고 주구성요소가 될 수 있는지를 결정한다.

(6) Construct 상태 : 모든 PGM이 상태전이를 완료하

였는지 확인하는 상태이다. 상태전이를 끝낸 PGM은 STC(State Transfer Complete) 메시지를 다른 PGM에게 전달한다. 모든 PGM이 STC 메시지를 전달한 사실이 확인되면 R-Prim 상태로 변경된다.

5. 프로그래밍 인터페이스

자바응용서비스 및 클라이언트의 개발을 위한 프로그래밍 인터페이스를 개발함으로써 분할 가능한 이기종 분산환경에서의 자바응용서비스를 쉽게 작성할 수 있도록 하였다.

5.1 JACE API

프로세스 그룹의 형태로 서비스를 제공할 수 있는 자바응용서비스의 개발 및 클라이언트 개발을 위하여 시스템이 제공하는 API는 아래와 같다. JACE API는 자바응용서비스의 개발을 위하여 <표 5>의 JACEServerAPI와 클라이언트 개발을 위한 <표 6>의 JACEClientAPI로 구성된다.

〈표 5〉 JACE 서버 API

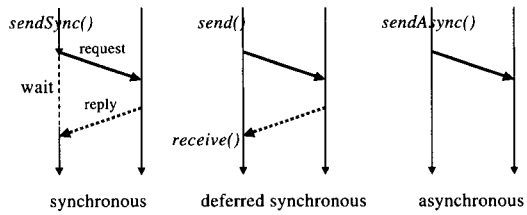
서버 API	메쏘드	설 명
JACEServerAPI	*connect*	JACE 시스템에 연결
	disconnect	JACE 시스템과의 연결 종료
	create	프로세스 그룹을 생성
	createJoin	기존 그룹이 없을 경우 : 그룹을 생성 기존 그룹이 있을 경우 : 그룹에 참여
	join	프로세스 그룹에 참여
	leave	프로세스 그룹으로부터 탈퇴
	reply	응답 메시지 전송
	event handler	클라이언트 요청 메시지 수신

〈표 6〉 JACE 클라이언트 API

클라이언트 API	메쏘드	설 명
JACEClientAPI	*connect*	JACE 시스템에 연결
	disconnect	JACE 시스템과의 연결 종료
	send *sendAsync* *sendSync*	프로세스 그룹으로 서비스 요구 메시지 전송
	receive	프로세스 그룹으로부터 응답 메시지 수신

5.2 메시지 교환

자바응용서비스가 프로세스 그룹의 형태로 서비스를 제공하기 위해서는 서버와 클라이언트간에 서비스를 요구하고 그 결과를 되돌려 줄 수 있는 방법이 제공되어야 한다. JACE에서는 해당 그룹에게 이벤트를 포함한 메시지를 전달하는 형태로 이루어지며 세 가지 유형의 이벤트 전달 방법을 정의하였다. (그림 6)은 이들의 차이점을 보여주고 있다.



(그림 6) 동기화 유무에 따른 이벤트 발생 유형

sendSync()는 메시지를 전달하고 결과가 반환될 때까지 호출 프로세스는 블록되었다가, 결과 이벤트가 반환된 이후에 다시 수행을 시작할 수 있다. send()는 이벤트를 포함한 메시지를 전달한 후 블록되지 않고 수행을 계속하다가 자신이 원할 때 receive()를 호출함으로써 응답 이벤트가 포함되어 있는 메시지를 받아올 수 있다. receive()를 호출하였을 때 만약 반환 메시지가 아직 도착하지 않았다면 호출 프로세스는 반환 메시지가 도착할 때까지 블록 된다.

sendAsync()는 비동기 메시지의 전송 방법이다. 호출 프로세스는 이벤트를 포함한 메시지를 전달한 후 그 결과의 반환 여부와 관계없이 계속해서 수행을 할 수 있다. 이 세 가지 메쏘드(method)는 클라이언트가 서버 그룹에 서비스를 요구하기 위하여 사용되며 이 이벤트는 서버로서 동작하는 자바응용서비스의 이벤트 핸들러에 전달되어 처리된다.

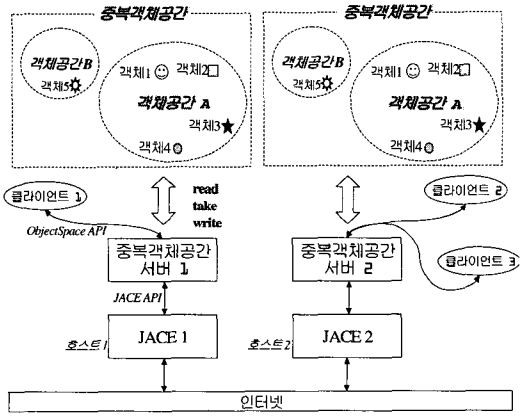
6. 응용 사례

인터넷상에서 효율적으로 공동작업을 수행하기 위하여 자원의 공유와 통신을 간편하게 할 수 있는 환경이 제공되는 것이 바람직하다. 이러한 요구사항을 만족시키기 위하여 공유객체공간을 제공하는 튜플스페이스 시스템[15]이 제안되었으며, 최근에는 썬 마이크로시스템즈사의 JavaSpace[16, 17]가 개발되었다. 그러나 기존의 개발된 시스템은 중앙 집중적인 구조로 수행됨으로써 네트워크가 단절되거나 서버가 실패하는 경우 지속

적인 서비스를 제공할 수 없게 된다.

6.1 중복객체공간(Replicated Object Space)

개발된 JACE 시스템을 이용하여 인터넷 환경에서 공유하고자 하는 객체를 (그림 7)과 같이 관련노드에 중복(replication)하여 저장하는 중복객체공간이 실험적으로 구현되었다. JACE 시스템과 함께 각 호스트 상에서 수행되는 중복객체공간 응용서버는 "ObjectSpace"라는 이름으로 프로세스 그룹을 형성하여 서비스를 제공한다.



※ 동일한 객체가 호스트1과 호스트2에 중복되어 저장된다.

(그림 7) 중복객체공간

6.2 중복객체공간의 기본 API

네트워크 응용프로그램은 <표 7>의 중복객체공간 기본 API를 이용하여 효율적으로 자원을 공유하고 통신할 수 있다.

〈표 7〉 중복객체공간 서비스의 기본 API

API	설 명
read(조건)	조건을 만족하는 임의의 객체를 읽어온다.
readAll(조건)	조건을 만족하는 객체 전체를 읽어온다.
take(조건)	조건을 만족하는 임의의 객체를 가져온다.
takeAll(조건)	조건을 만족하는 객체 전체를 가지온다.
write(객체)	공유할 목적으로 객체를 저장한다.
createObjectSpace (그룹이름)	특정 이름의 객체공간을 생성한다.
removeObjectSpace (그룹이름)	특정 이름의 객체공간을 삭제한다.

※ (조건)에는 찾고자하는 객체의 특성을 포함하는 템플릿 객체를 인수로 넣는다.

6.3 중복객체공간의 주요 구현코드

JACE API를 이용하여 실험적으로 구현한 중복객체공간의 주요 코드는 다음과 같다.

```
// 인터페이스 JACEApplication은 OnGroupEventListen 메쏘드를 포함한다.
public class ObjectSpaceImpl implements JACEApplication {
   // 객체저장소를 생성한다. Java의 Collection 클래스를 이용하여 구현하였다.
   ObjectRepository repository = new ObjectRepository();
   ProcessGroupManager osPGM = null;

 public ObjectSpaceImpl() {
   // 자바응용서비스 개발을 위한 Server API.
   osPGM = JACEServer.connect(this); // JACE 시스템에 연결.
   // 그룹을 생성하거나 참여한다.
   osPGM.createJoin("ObjectSpace", Protocol.HISTORY_SENSITIVE);
   osPGM.goEventLoop(); // 서비스를 요청하는 이벤트를 기다린다.
 }
 public void OnGroupEventListen(GroupEvent e, Object piggyBackInfo) {
   // 이벤트 유형에 따라서 중복객체공간 서비스를 제공한다.
   switch (e.getEventType()) {
     // 조건에 맞는 객체를 읽어온다.
     case ObjectSpaceEvent.Read :
       Entry myEntry = (Entry) repository.read(e);
       .....
       reply = new ObjectSpaceEvent(ObjectSpaceEvent.Read_Reply, arg);
       // 클라이언트 요청에 대한 응답 메시지를 전송한다.
       osPGM.reply(reply, piggyBackInfo, Protocol.SAFE_DELIVERY);
       break;
- 이하 생략 -
```

6.4 중복객체공간을 이용하는 응용프로그램의 주요 구현코드

다음은 중복객체공간 기본 API를 이용하여 구현한 예제 프로그램을 보여준다.

```
public class HelloEntry extends Entry { // 객체공간에 공유하고자하는 엔트리(Entry)
   int value;
   String content;
   public HelloEntry(String aMessage) {
         content = aMessage;
   }
   public HelloEntry(int aValue, String aMessage) {
         value = aValue;
         content = aMessage;
   } // .....
}
public class HelloObjectSpace {
   ObjectSpace osProxy = null;
   ProcessGroup connection = null;
   .....
   public HelloObjectSpace(String groupInfoManager) {
   /* 중복객체공간 서버가 'ObjectSpace'라는 그룹이름으로 동작하고 있을 때,
      인자값으로 주어진 그룹위치정보 관리자로부터 그룹위치정보를 얻은다. 이
      위치정보를 이용하여 중복객체공간 서버가 수행되는 호스트상의 JACE
      시스템과 연결한다. */
```

```
    connection = JACEClient.connect("ObjectSpace", groupInfoManager);
    osProxy = new ObjectSpace(connection);
}
public void go(){
    HelloEntry myEntry, resultEntry, Entry = null;
    // 'HelloSpace'라는 이름으로 객체공간을 생성한다.
    osProxy.createObjectSpace("HelloSpace");

    // 공유하고자 하는 객체를 생성한다.
    myEntry = new HelloEntry("HelloWorld"));
    // 'HelloSpace' 객체공간에 myEntry를 저장한다.
    osProxy.write("HelloSpace", myEntry);
    .....
    dataEntry = new HelloEntry(2000, "Cost");
    resultEntry = (HelloEntry) osProxy.read("HelloSpace", dataEntry);
}
public static void main(String args[]){
        HelloObjectSpace test_app = new HelloObjectSpace(args[0]);
        test_app.go();
}
- 이하 생략 -
```

7. 결 론

그룹통신 시스템은 VS 모델을 지원하는 시스템과 EVS 모델을 지원하는 시스템으로 구분된다. EVS 모델은 VS 모델을 확장하여 네트워크 분할과 재결합이 발생되더라도 그룹 구성원간의 일관성을 보장한다.

본 논문에서는 인터넷 환경에서 EVS 모델을 지원하는 JACE 그룹통신 시스템의 설계와 구현에 대하여 기술하였다. JACE 시스템은 각 그룹별로 가상 링을 형성하여 논리적인 토큰을 순환시키는 GC, 응용프로세스 그룹을 관리하는 RPGS 계층, 그리고 자바응용서비스와 클라이언트 개발을 위한 JACE API로 구성되어 있다. 개발된 JACE 시스템을 이용하여 인터넷 환경에서 자원의 효율적인 공유와 통신수단을 제공하는 중복 객체공간이 실험적으로 구현되었다. JACE 시스템이 가지는 특징은 다음과 같다.

첫째, JACE 시스템은 Java를 이용하여 개발되었기 때문에 특정 플랫폼에 독립적으로 실행될 수 있다. 둘째, 다양한 분야에서 제공되는 자바응용서비스의 견고성을 지원하는 그룹통신 시스템이다. 셋째, 프로세스 그룹의 성격에 따라 history-sensitive 그룹과 history-free 그룹으로 구별하여 그룹통신을 지원한다. 기존의 처리결과에 영향을 받는 history-sensitive 그룹은 주구성요소에 참여하는 구성원만이 서비스를 제공하고 history-free 그룹은 기존의 처리결과에 영향을 받지 않으므로 부구성요소의 구성원도 서비스를 제공한다. 넷째, JACE 시스템은 인터넷 환경에서 EVS 모델을 지원하는 그룹통신 시스템이다.

본 시스템은 신뢰성 있는 그룹통신을 지원하기 위한 미들웨어로 사용되는데 현재의 구현에서는 JACE 데몬 프로그램과 그룹위치정보 제공 서버가 시스템 사용자를 위한 GUI를 제공하고 있지 않다. 앞으로 환경설정 및 실행상태를 모니터링을 할 수 있는 GUI 환경을 제공할 계획이며, 인터넷상의 다양한 트래픽(traffic), 그룹에 참여하는 구성원의 지리적 밀도, 네트워크의 분할 회수, 메시지 분실 회수와 메시지 재전송 회수 등에 대하여 통계적인 자료를 산출하여 수행시간에 최적의 토큰 순환경로를 결정하고 재전송 되는 메시지의 수를 줄여 시스템의 성능을 향상시킬 계획이다. 또한 JACE 시스템을 내장한 웹서버를 개발함으로써, 인터넷상의 서로 다른 지역에서 동일한 서비스를 제공하는 웹서비스를 구축할 계획이다.

참 고 문 헌

[1] K. Birman and T. Joseph. "Exploiting Virtual Synchrony in Distributed Systems." In *Proceeding of the ACM Symposium on Operating Systems Principles*, pp.123-138, November 1987.

[2] K. P. Birman, "Virtual Synchrony Model," In Reliable Distributed Computing with the Isis Toolkit, IEEE press.

[3] K. Birman and R. van Renesse. Reliable Distributed Computing with the ISIS Toolkit, Los Alamitos, CA., IEEE Computer Society Press, 1994.

[4] K. Birman, R. Cooper, B. Gleeson. "Design Alternatives for Process Group Membership and Multicast." TR91-1257, Cornell University Computer Science Department, Ithaca, NY. December 1991.

[5] L. E. Moser, Y. Amir, P. M. Melliar-Smith and D. A. Agarwal. "Extended Virtual Synchrony." In Proceeding of the 14th Internatinal Conference on Distributed Computing Systems, pp. 56-65, June 1994.

[6] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, and C. A. Lingley-Papadopoulos,

"Totem : A Fault-Tolerant Multicast Group Communication System," Communications of the ACM Vol.39 No.4, pp.54-63, 1996.

[7] D. Malki. Multicast Commnication for High Availability. Ph.D. thesis, Institute of Computer Science, The Hebrew University of Jerusalem, Israel, 1994.

[8] R Van Renesse, K. P. Birman, and S. Maffeis, "Horus : A Flexible Group Communication System," Communications of the ACM, Vol.39, No. 4, pp.75-83, 1996.

[9] 문남두, 최혁재, 유양우, 박양수, 이명준, "자바를 이용한 Extended Virtual Synchrony의 지원", 정보과학회 추계학술발표논문집 25권 2호, pp.409-411, 1998.

[10] 최혁재, 문남두, 김현규, 박양수, 이명준, "분할가능 분산환경에서의 신뢰성 있는 자바 프로세스 그룹 서비스", 정보과학회 추계학술발표논문집 25권 2호, pp.196-198, 1998

[11] 최혁재, 문남두, 박양수, 이명준. "자바응용서비스 개발을 위한 JACE 프로그래밍 인터페이스". 한국정보과학회 '99 봄 학술발표논문집26(1), pp.382-384. 1999.

[12] Weijia Jia, "Implementation of a Reliable Multicast Protocol," Software-Practice And Experience, Vol.27(7), pp.813-850, July 1997

[13] 문남두, 최혁재, 유양우, 이명준. "인터넷 환경을 지원하는 신뢰성 있는 그룹통신 시스템의 설계". 한국정보과학회 '99 봄 학술발표논문집26(1), pp. 283-285. 1999.

[14] Y. Amir. "Replication Using Group Communication Over a Partitioned Network." PhD thesis, Institute of Computer Science, The Hebrew University of Jerusalem, Israel, 1995.

[15] Antony I. T. Rowstron, Alan Wood : "An Efficient Distributed Tuple Space Implementation for Networks of Workstations." Euro-Par, Vol.I 1996. pp.510-513.

[16] "The JavaSpace Specifications," SunMicrosystems, http://chatsubo.javasoft.com, June 1997.

[17] Sun Microsystems,Inc. "JavaSpace Technology," "http://java.sun.com/products/javaspaces/index.html"

문 남 두

e-mail : dooya@cic.ulsan.ac.kr

1997년 울산대학교 전자계산학과 졸업(공학사)

1999년 울산대학교 전자계산학과 졸업(공학석사)

1999년~현재 울산대학교 컴퓨터정보통신공학과 박사과정

관심분야 : 분산객체 시스템, 네트웍 컴퓨팅, 인터넷 프로그래밍 시스템

안 건 태

e-mail : gtahn@cic.ulsan.ac.kr

1999년 울산대학교 전자계산학과 졸업(공학사)

1999년~현재 울산대학교 컴퓨터정보통신공학과 석사과정

관심분야 : 분산객체 시스템, 이동 에이전트 시스템, 그룹통신 시스템

유 양 우

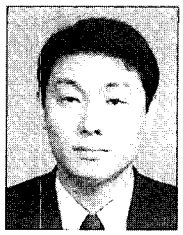
e-mail : soft@cic.ulsan.ac.kr

1995년 경일대학교 전자계산학과 졸업(공학사)

1997년 울산대학교 전자계산학과 졸업(공학석사)

1998년~현재 울산대학교 컴퓨터정보통신공학과 박사과정

관심분야 : 이동 에이전트 시스템, 그룹통신 시스템, 분산객체 시스템 등



이 명 준

e-mail : mjlee@uou.ulsan.ac.kr

1980년 서울대학교 수학과 졸업 (학사)

1982년 한국과학기술원 전산학과 졸업(석사)

1991년 한국과학기술원 전산학과 졸업(박사)

1982년~현재 울산대학교 컴퓨터정보통신공학부 근무 (현재 교수)

1993년~1994년 미국 버지니아대학 교환교수

관심분야 : 프로그래밍언어, 분산객체 프로그래밍 시스템, 병행 실시간 컴퓨팅, 인터넷 프로그래밍 시스템 등