

분산 멀티미디어 환경에서 실행되는 오류 분류 시스템의 성능 분석

고 응 남

천안대학교 정보통신학부

요 약

분산 멀티미디어 응용에 대한 요구는 정교한 QoS 관리를 위해서 필요하다. 분산 멀티미디어 시스템 견지에서 보면, QoS의 가장 중요한 범주는 시간성, 볼륨, 신뢰성이다. 본 논문에서는 결함 허용을 통해서 신뢰성을 향상시키는 방법에 대해서 기술한다. ECA는 분산 멀티미디어에서 하나의 소프트웨어 오류를 자동적으로 분류할 수 있는 시스템이다. 본 논문은 규칙-기반 DEVS 모델링과 시뮬레이션 기법을 사용하면서 분산 멀티미디어 상에서의 오류 분류 시스템의 성능 분석을 설명한다. DEVS에서 하나의 시스템은 시간, 입력, 상태, 출력 및 함수들을 가지고 있다.

Performance Analysis of Error Classification System on Distributed Multimedia Environment

Eung-Nam Ko

ABSTRACT

The requirement of distributed multimedia applications is the need for sophisticated QoS (quality of service) management. In terms of distributed multimedia systems, the most important categories for quality of service are a timeless, volume, and reliability. In this paper, we discuss a method for increasing reliability through fault tolerance. We describe the design and implementation of the ECA running on distributed multimedia environment. ECA is a system is able to classify automatically a software error based on distributed multimedia. This paper explains a performance analysis of an error classification system running on distributed multimedia environment using the rule-based DEVS modeling and simulation techniques. In DEVS, a system has a time base, inputs, states, outputs, and functions.

1. 서 론

멀티미디어 처리 기술은 매우 다양한 응용 분야에서 활용되고 있으며, 특히 휴먼-컴퓨터 상호 작용 기술과 결합되어 컴퓨터 사용자에게 배우기 쉽고 사용하기 쉬운 사용자 인터페이스를 제공하고 있으며, 실세계에 존재하는 각종 대화 비유를 컴퓨터에게 제공하고 있다. 교육 의료, 광고, 사업용 멀티미디어 응용을 기반으로 하여 멀티미디어는 사회 전반에 걸쳐 활성화되고 있다. 즉 멀티미디어는 교육, 원격 진료 등 다양하고 광범위한 분야에서 응용되고 있고 사회 전 분야에서 살펴볼 수 있다[1,2,3]. 분산 멀티미디어 정보 시스템에 대한 요구는 매우 빠르게 상업, 생산, 교육, CAD(Computer Aided Design)/CAE(Computer Aided Engineering), 의학, 기상 등 많은 분야에서 필요로 하고 있다[4,5]. 최근 들어 이러한 멀티미디어 시스템의 공동 작업 환경이 증가하고 있는데 반하여 이러한 시스템에서의 전체적인 망 관리, 특히 응용 S/W에 대한 결함을 발견 및 복구하는 연구는 미흡한 실정이다[6]. 분산 시스템은 하나의 노드 또는 자원에 결함이 발생해도 전체 시스템에는 큰 손실을 입히지 않는 특성을 가지기 때문에 결함 허용 시스템의 설계를 위한 좋은 조건을 가진다[6,7]. 따라서, 본 논문에서는 분산 멀티미디어 환경에서 응용 S/W의 결함을 미리 감지한 후에, 분류하여 알려주고 복구할 수 있는 에이전트 중에서 오류를 분류하는 에이전트를 DEVS(Discrete Event Specification) 형식론(formalism)의 계층적 구조를 이용하여 효율성을 분석, 비교한다. 본 논문의 구성은 2에서 관련 연구를 기술하고, 3에서는 제안하는 분산 멀티미디어 환경에서의 ECA를 기술하고, 4에서는

오류 유형 분류 시스템의 모델링, 5에서는 결론을 기술한다.

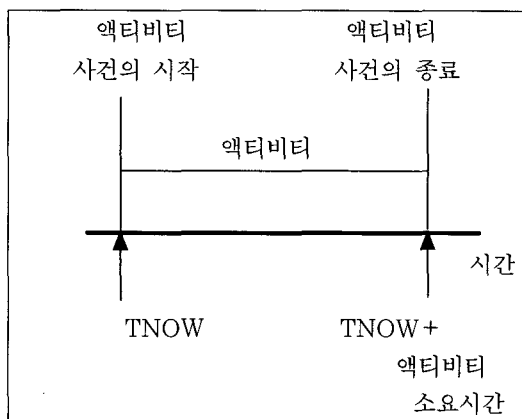
2. 관련 연구

이 장에서는 이산 사건 모델링(Discrete Event Modeling)과 DEVS의 형식론에 대해서 기술한다.

2.1 이산 사건 모델링

이산 사건 모델링(Discrete Event Modeling)의 목적은 엔티티(entity) 들이 포함된 액티비티(activity)들을 재생하여 시스템의 잠재적 행태와 성능에 대한 것들을 연구하고자 하는 것이다. 이러한 목적은 시스템의 상태들을 정의하고, 시스템을 어떤 상태에서 다른 상태로 옮기는 액티비티들을 구축함으로써 달성된다. 시스템의 상태는 엔티티들의 속성에 할당된 수치의 측면에서 정의가 된다. 모든 엔티티들이 어떤 상태에 대하여 정의된 속성값의 범위에 있을 때 시스템은 그 상태에 있다고 말한다. 따라서 이산사건 시뮬레이션은 시간에 따른 시스템 상태의 동적인 묘사이다[8].

이산 사건 모델링에서 사건과 액티비티간의 관계는 그림 1과 같으며, 액티비티가 시작하거나 끝나는 결정이 이루어지는 시점에 사건이 발생하고 있다. 프로세스(process)는 시간에 따라 정렬된 사건들의 순서이며, 여러 액티비티 들을 포함할 수 있다[8].



(그림 1) 사건과 액티비티 간의 관계

2.2 DEVS의 형식론

DEVS(Discrete Event System Specification)는 Bernard P. Zeigler에 의해 개발된 이산 사건 모델들의 계층 구조적 모듈화 방법을 제공하는 형식론이다. 시스템을 작은 모듈들로 나누고 그것들로 전체 시스템을 계층적으로 구성해 나간다. 각 모듈들은 원자(atomic) 모델로 표현되며 그것들의 계층적 구성은 커플(coupled) 모델로 표현된다. 모델링된 시스템의 시뮬레이션을 위해 추상화 시뮬레이터(Abstract simulator) 알고리즘이 제공된다. 추상화 시뮬레이터의 종류에는 시뮬레이터(simulator)와 협동자(coordinator)가 있으며 이것들은 각각 원자모델과 커플모델을 위한 시뮬레이터이다. 모델들과 추상화 시뮬레이터들은 일대일 대응 관계를 가진다. 즉, 하나의 모델은 하나의 추상화 시뮬레이터와 항상 쌍을 이루게 된다. DEVS 형식론에서 가장 기본이 되는 모델인 원자 모델은 다음과 같은 집합으로 표현된다 [9-13].

$$M = \langle X, S, Y, \text{int}, \text{ext}, \dots, \text{ta} \rangle$$

- X: 외부 입력 사건들의 집합
- S: 상태 변수들의 집합
- Y: 외부 출력 사건들의 집합
- int : 내부적 상태 변환 함수
- ext : 외부적 상태 변환 함수
- λ : 출력 함수
- ta : 시간 진행 함수

원자모델을 결합하여 새로운 결합 모델을 형성한다. 이 결합 모델은 또한 다른 모델의 구성 요소 모델이 될 수 있기 때문에 이것을 이용하여 복잡한 모델을 계층적으로 구성할 수 있게 된다. 결합 모델의 구조 표현은 다음과 같다[9-13].

$$DN = \langle D, \{Mi\}, \{Ii\}, \{Zij\}, \text{select} \rangle$$

- DN: Diagraph Network
- D : 구성 요소 모델들의 이름의 집합
- {Mi}: i번째 구성 요소를 이루는 기본 모델들
- {Ii}: i번째 모델의 influencees 모델들의 집합
- {Zij}: i번째 모델의 출력을 j번째 모델의 입력으로 연결하는 함수
- select: 여러 구성 요소들이 같은 시간에 스케줄을 원할 때 그 중에서 하나를 선택하는 함수

3. 분산 멀티미디어 환경에서의 ECA

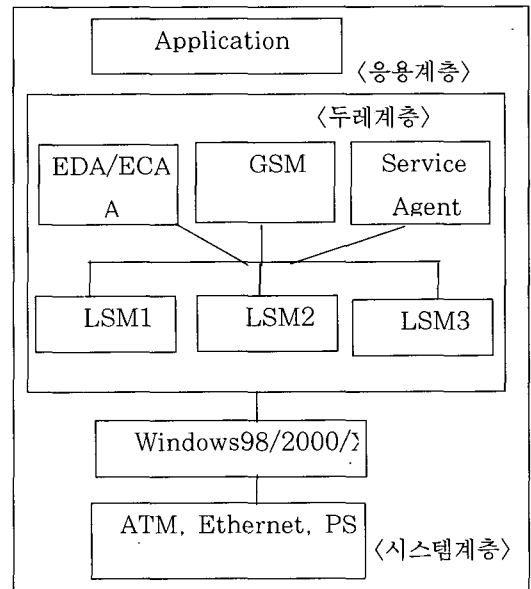
본 논문에서는 멀티 에이전트의 특성인 각 에이전트의 자치권을 보장하면서 서로간의 협력을 통해서 전체적인 망 관리, 특히 응용 S/W에 대한 결함을 발견하고 복구하는 에이전트 중에서 발견된 오류를 빨리 오류 유형을 찾는 에이전트인 ECA(Error Classification based on Agent)를 실세계의 환경으로 모델링 한다.

3.1 분산 멀티미디어 환경

두레는 상호 참여형 멀티미디어 응용 개발 환경으로 그림 2처럼 4계층으로 구성되어 있다. 기존의 단일 멀티미디어 응용 개발에 발생하는 미디어 제어와 세션 관리에 대한 개발 과정의 비용을 줄임으로 상호 참여형 멀티미디어 응용의 개발을 가능하게 지원하는 미들웨어이다. 통신 계층은 분산 처리 환경의 메시지 전송을 담당하는 계층으로 MS-WINDOWS의 socket system을 활용한다. 즉, socket 함수의 family로는 AF_INET를, type은 비연결형 또는 연결형을 사용한다. 시스템 계층은 기본적으로 기존의 운영체제(예: Windows 95/98/2000/XP) 기능을 사용한다 [2]. 두레 계층은 상호 참여형 멀티미디어 일반적인 응용을 개발하기 위해서 설계된 프레임워크이다. 두레에서 제공되는 서비스 에이전트에는 여러 개의 기능들을 가진다. 이 에이전트들은 상호 협력작업을 지원하기위한 것으로서 세션 관리 에이전트, 접근/동시성 제어 에이전트, 오디오 혹은 미디어 자원의 공유를 가능하게 하는 미디어 제어 에이전트, 공동 작업 시 공동작업 공간(화이트보드 등)에서의 동일한 화면을 보게 하여 동시작업을 가능하게 하는 커플링 에이전트, 전자우편 혹은 인터넷등 외부 네트워크와 접속을 담당하는 메일링 에이전트, 전체 세션에서 발생하는 세션의 종류, 이름, 참여자 명단, 통신의 양을 관리 하는 세션 감시 에이전트, 상용의 프리젠테이션 도구나 저작도구 등으로 개발된 소프트웨어를 공유하여 사용할 수 있게해 주는 응용공유 에이전트 등이있다. 또 이들의 외곽에는 통신 에이전트가 있어 여러 가지 통신 프로토콜을 지원 한다.

3.2 ECA

본 논문에서 제안하는 ECA는 여러 기능의 에이전트가 존재하며 발견된 오류를 자동적으로 유형 분류하여 복구하는 Multi-agent 시스템이다. 이들 에이전트 간의 상호 협조를 통하여 비정상적으로 종료되어 inactive되어 있는 응용 S/W를 발견하고 이들 agent 간의 상호 협조를 통하여 자동적으로 active시켜주는 intelligent agent이다.



(그림 2) 두레의 구조

ECA(Error Classification Agent)와 상호 연관이 되는 에이전트는 EDA(Error Detection Agent), ERA(Error Recovery Agent) 및 SMA(Session Management Agent)이다. SMA는 다양한 서비스 객체들의 상호 작용에 의해서 지원된다. 최초 응용이 실행되면 응용은 데몬 객체의 존재를 찾아 응용으로써의 등록을 요구

한다. 세션 생성을 위한 모든 필요한 준비가 끝나게 되면 세션 관리자는 응용 인터페이스를 통해 응용에게 세션이 생성되었음을 알려준다. 이후 응용은 세션에 대한 모든 서비스를 세션 관리자를 통해 요구할 수 있게 된다. EDA는 오류를 감지하는 핵심 agent로 fail detect 정보 흐름은 packet 정보를 보내어 그 응답 결과 상태를 분석하여 오류의 발생 여부를 분석한다. 세션의 복원을 진행하기 위해서는 먼저 에러 감지를 하기위한 방법이 필요하다. MS-Windows 98/2000/XP의 시스템에서 에러 감지를 위한 방법은 실행된 프로세서의 상태를 보관하는 프로세서 데이터 베이스를 주기적으로 검사하는 방법이다. 그러나 이것은 두레를 이용한 세션의 상태와는 무관한 프로세서까지 데이터베이스를 검사해야 한다는 단점이 있다. 그래서 두레를 이용한 시스템에서는 데몬이나 세션 매니저가 생성한 프로세서에 대한 정보를 결합 감지기에게 통보하여 결합 허용 시스템에서 세션과 직접 연관된 프로세서만을 주기적으로 폴링(polling)하여 에러 감지를 수행한다. ECA는 사용자와 결합 감지 및 복구를 위한 EDA 및 ERA와 같이 에이전트를 기반으로 한 응용 분야와의 상호 작용을 위한 인터페이스로서의 역할을 하는 에이전트이다. ERA는 비정상적인 상황의 통보에 대하여 복원의 과정을 수행한다.

4. 오류 유형 분류 시스템의 모델링

분산 멀티미디어 환경 시스템 중에서 4번째 계층인 응용 계층과 3번째 계층인 두레 계층 사이의 관계만 국한 시킨다.

4.1 설계 요소(Design constraints)

오류 감지, 유형 분류 및 복구 계층에서는 감지 및 복구 기능은 제외시킨다. 즉, ECA에 대한 모델링을 통해서 본 시스템에 대한 범위를 한정한다. ECA에 대한 설명과 분석을 위해서 필요한 정의 및 표기는 다음과 같다.

(정의 1)

본 논문에 관계되는 Agent의 집합은 다음과 같다.

$$ECA = \{ECA_1, ECA_2, \dots, ECA_n\} \quad (n \in \mathbb{N})$$

(ECA: 오류 유형 분류 Agent 들의 집합)

(정의 2)

실제 환경 상태 P, 즉 오류 감지 또는 복구 대상이 되는 미디어, 미디어 인스턴스 및 응용 프로그램들의 집합은 다음과 같다.

$$P = \{p_1, p_2, \dots, p_n\} \quad (n \in \mathbb{N})$$

P_i 는 t_i 와 t_j 사이의 시간 간격에서 실행하는 프로세스 들의 집합으로 정의한다.

즉, $P_i = \{p_i \mid t_i \leq p_i < t_j\}$ 이다.

(정의 3)

E_i 는 t_i 와 t_j 사이의 시간 간격에서 발견되는 오류(error)들의 집합으로 정의한다.

$$E_i = \{e_i \mid t_i \leq e_i < t_j\}$$

F_i 는 t_i 와 t_j 사이의 시간 간격에서 발생하는 오류의 원인이 되는 결함(fault)들의 집합으로 정의한다.

$$F_i = \{f_i \mid t_i \leq f_i < t_j\}$$

S_i 는 t_i 와 t_j 사이의 시간 간격에서 SM이라는 세션 관리자를 이용하여 필요한 응용을 등록시키는 데 발생하는 시간의 집합으로 정의한다.

$$S_i = \{s_i \mid t_i \leq s_i < t_j\}$$

4.2 모델링(Modeling)

4.2.1 제안하는 알고리즘

ECA는 EDA에서 접수된 오류와 프로세스 데이터 베이스를 이용하여 자동 분류할 수 있는 지적 대리인이다. 분류된 데이터를 가지고 지식베이스를 통해 분석적인 정보를 생성하고 학습의 기능도 가지며 집합으로 표시하면 다음과 같다.

Set of Classification = {Set of error, Set of fault, Set of error classifier}

여기에서

Set of error = {E, D}

(E: 발생하는 오류,

D: 발생하는 도메인 위치)

Set of fault = {H, C, F}

(H: 검출된 오류,

C: PDB에서 찾은 error code 즉, GetExitCodeProcess 실행 후에 return되는 코드 값,

F: 오류의 원인이 되는 결함)

Set of error classifier = {Addr_ECA, Func_ECA}

(Addr_ECA: ECA의 주소 정보,

Func_ECA: ECA의 기능)

t_i 와 t_j 사이의 시간 간격에 활동하는 ECA_i의 기능은 다음과 같다.

$$\text{Function of ECA}_i = \{ p_i \times e_i \times s_i \rightarrow f_i \mid p_i \in P_i, e_i \in E_i, s_i \in S_i, f_i \in F_i \}$$

오류 검출된 프로세스의 오류 유형을 알기 위하여 프로세스데이터베이스를 검색하면 알 수 있다. 윈도우 함수 중 GetExitCodeProcess를 사용할

때 리턴코드 값에 의해서 알 수 있다. MS-Windows 95/98/NT의 시스템에서 오류 검출을 위한 방법은 실행된 프로세서의 상태를 보관하는 프로세서 데이터 베이스를 주기적으로 검사하는 방법이다. 그러나 이것은 두레를 이용한 세션의 상태와는 무관한 프로세서까지 데이터베이스를 검사해야 한다는 단점이 있다. 그래서 두레를 이용한 시스템에서는 데몬이나 세션 매니저가 생성한 프로세서에 대한 정보를 결합 검출기에게 통보하여 내결함 시스템에서 세션과 직접 연관된 프로세서만을 주기적으로 폴링하여 오류검출을 수행한다. 프로세스 데이터베이스의 구조 중 프로세서의 살아있는 상태를 알려면 종료 상태의 내용을 보고 알 수 있다. 예를 들면, 하나의 프로세스가 여전히 활동하고 있으면 STILL_ACTIVE(헵사 코드값: 103)이다. 이 값을 알려면 API 함수 중 GetExitCodeProcess를 사용했을 때 리턴(return)되는 코드값으로 알 수 있다.

4.2.2 제안하는 방식과 기존 방식의 효율성 비교

본 논문에서는 오류 검출된 프로세스의 오류 유형을 알기 위하여 프로세스데이터베이스를 검색하면 알 수 있다. 검색에 있어서 전체 응용에 대해서 폴링 방식을 사용하는 방식과 폴링을 사용하되 기존 시스템의 정보를 갖고 있는 SM이라는 세션 관리자를 이용하여 등록되어 있는 필요한 응용만 찾아서 폴링 시간을 줄이는 방식 2가지를 비교하였다.

(1) 기존 방식

DEVS 형식론에서 원자 모델(atomic model)을 결합하여 새로운 커플 모델(coupled model)을 형성한다[9,10,11]. 기존 방식에서 상태 변수는 표1과 같고 component interaction은 표2와

같다.

(표 1) 기존 방식에서 상태 변수

model	state variable	purpose
EF (<i>genr</i>)	poll_int	polling interval
RA1	ra1_re_time app_count ra1_re_t_a	response time 두레 관련 응용 갯수 반응시간 누적
UA1	ua1_re_time app_count ua1_re_t_a	response time 두레 무관 응용 갯수 반응시간 누적
ED1	ra1_re_t_a ua1_re_t_a tat1_t_a	ra1 반응시간누적 ua1 반응시간누적 ra1+ua1 누적

(표 2) 기존 방식에서 component interaction

순서	component interaction	message
step1	genr→ RA1 genr→ UA1	polling 발생 시점을 준다. RA1, UA1는 자신의 반응시간을 계산한다.
step2	RA1→ ED1 UA1→ ED1	RA1, UA1는 자신의 ED1의 polling에 대한 반응시간을 준다. ED1은 RA1, UA1로부터의 반응시간을 누적한다. 간격 step1-step2 반복
step3	ED1→ EF1	ED1는 EF1에게 RA1, UA1 반응시간 누적 총합을 준다 (종료시점).

(2) 제안된 방식

DEVS 형식론에서 원자 모델(atomic model)을 결합하여 새로운 커플 모델(coupled model)을 형성한다. 제안된 방식에서 상태 변수의 정의는 표3과 같고 component interaction은 표4와 같다.

(표 3) 제안된 방식에서 상태 변수의 정의

model	state variable	purpose
EF(<i>genr</i>)	poll_int	polling interval
RA2	ra2_re_time app_count ra2_re_t_a	response time 응용 갯수 반응시간 누적
ED2	ra2_re_t_a sm_t_a tat2_t_a	ra2반응시간누적 SM 정보등록 시간 ra2시간누적 + SM정보등록시간

(표 4) 제안된 방식에서 component interaction

순서	component interaction	message
step1	genr→ RA2	polling 발생 시점을 준다. RA2는 자신의 반응시간을 계산한다.
step2	RA2→ ED2	RA2는 자신의 ED2의 polling에 대한 반응시간을 준다. ED2는 RA2로부터의 반응시간을 누적한다. 간격 step1-step2 반복
step3	ED2→ EF2	ED2는 EF2에게 RA2 반응시간 누적과 SM정보등록시간 총합을 준다(종료시점).

(3) 시뮬레이션 결과

분산 멀티미디어 환경에서 실행되는 결합 허용 시스템의 기능 중에서 결합 오류 유형 분류시 성능 분석을 DEVS 형식론을 이용하여 살펴 보았다. 즉, simulation model을 통한 관측 목표와 관측 값 계산에 관련된 변수를 상태 변수로 가지는 모델이다. 기존 방식에서 만일 오류의 유형을 찾기 위하여 프로세스 데이터 베이스를 이용하여 프로세스 들을 찾는데 걸리는 시간을 t 라고 하면 한번 폴링 시간은 $2t$ 가 된다. 세션 관련 프로세스의 갯수가 m 개, 세션과 무관한 프로세스의 개수

를 r 이라고 하면 기존 방법의 오류 유형 찾기 시간의 기대값 (Γ_2)은 다음과 같다. $\Gamma_2 = 2t^*(m+r)$. 세션 등록 시간을 s 라고 하면 제안된 방법의 오류 유형 찾기 시간의 기대값 (Γ_2)은 다음과 같다.

$$\Gamma_2 = 2t^*m + s$$

그러므로 $r > m$ 인 경우, 즉, 두레에 관련되지 않는 응용 프로그램의 실행 갯수가 많을수록 제안된 방식은 효율적이다. 단, 반대인 경우에는 기존 방법이 효율적일 수 있다.

5. 결론

본 논문에서는 DEVS 형식론을 사용하여 실제 계의 분산 멀티미디어 환경에서의 시스템을 모듈화 된 가장 작은 단위로 genr, trnsducer, RA, UA, ED로 나누었다. genr과 trnsducer 각각의 atomic model을 결합하여 EF coupled model을 형성한다. 또한 UA, RA, ED를 결합하여 DP라는 coupled model을 형성하고 EF-DP를 결합하여 coupled model을 형성하는 것이다. 본 논문에서는 오류의 유형을 찾기 위하여 프로세스 데이터 베이스를 이용하여 프로세스들을 찾았다. 전체 응용에 대해서 폴링 방식을 사용하는 방식과 폴링을 사용하지 기존 시스템의 정보를 갖고 있는 SM이라는 세션 관리자를 이용하여 등록되어 있는 필요한 응용만 찾아서 polling의 시간을 줄이는 방식 2가지를 비교하여 효율성 검토를 할 수 있는 시스템의 모델링을 제안하였다. 앞으로의 연구 방향은 이러한 시스템을 정형화하는 부분이 과제이다. 오류 분류뿐만 아니라 오류 복구 등에 대해서도 DEVS 형식론을 이용하여 분석하는 일 등이다. 또한 변수들의 값과 SM에 등록된 정보를 주는 시간을 통계화하고 일반화하여 모델링 및 시

물레이션 하는 연구 등이 있다.

참고문헌

- [1] 오승준, "멀티미디어 정보 처리와 응용", 정보 과학회지 제 9권 제 3호, 1991년 6월, pp.19-25.
- [2] Eung-Nam, Dae-Joon Hwang, "Implementation of a Fault-Tolerant System Running on DOORAE: FTSD". In proccdings of IEEE ISCE98, Taipei, Taiwan, October 19-21, 1998.
- [3] Eung-Nam, Dae-Joon Hwang, Jae-Hyun Kim, "Implementation of an Error Detection-Recovery System based on Multimedia Collaboration Works: EDRSMCW". In proccdings of IASTED MIC'99, Innsbruck, Austria, February 15-18, 1999, pp.100-103.
- [4] 전준걸, 황대준, "상호 참여를 위한 탁상회의 시스템의 구현", 95년 한국 정보 과학지 가을학술 발표 논문집 vol.22, No.2, pp.1041.
- [5] 박길철, 황대준, "네트워크 환경에서 멀티미디어 객체 동기화 모델 설계", 한국 정보처리 학회 제 1 권 2호, pp.568-571, 1994.10.
- [6] 김문희, "결합 허용 시스템의 설계 고려사항 및 동향", 정보과학회지, 제11권 제3호, pp.7, 1993.
- [7] 장순주, 임종규, 정구영, 구용완, "분산 시스템에서 결합 허용성을 위한 프로세스 이주 연구", 한국 정보 과학회지 가을 학술발표논문집 Vol.21, No2, pp. 132.1994.
- [8] 김영실, 백두권, 이산 사건 모델링과 시물레

이선, 정보과학회지 제 13권 제 4호, 1995년 4월, pp.9-10.

- [9] Bernard P. Zeigler, "Object-Oriented Simulation with hierarchical, Modular Models", Academic Press, 1990.
- [10] Bernard P. Zeigler, "Multifaceted Modeling and Discrete Event Simulation, Orlando, FL: Academic, 1984.
- [11] Bernard P. Zeigler, "Theory of Modeling and Simulation, John Wiley, NY, USA, 1976, reissued by Krieger, Malabar, FL, USA, 1985.



고 응 남

1984년 연세대 수학과(이학사)
1991년 숭실대 정보과학 대학원 전산공학과 (공학석사)
2000년 성균관대 대학원 정보공학과(공학박사)

1983년-1993년 대우통신 컴퓨터개발부 선임연구원
1993년-1997년 동우대학 전자계산과 교수
1997년-2001년 신성대학 컴퓨터계열 교수
2001년-현재 천안대학교 정보통신학부 교수 및 멀티미디어 기술사
관심분야: 인터넷, 멀티미디어, CSCW, 결합허용, 에이전트 및 게임 등