

# UML을 이용한 홈 네트워킹 지원 게이트웨이 관리 서버 설계 및 구현

## A Design and Implementation of the Management Server for the Gateway Supporting Home Networking Using the UML

권진혁(Jin-Hyuck Kwon)<sup>1)</sup> 민병조(Min-Byung Jo)<sup>2)</sup> 강명석(Myung-Seok Kang)<sup>3)</sup> 남의석(Eui-Seok Nahm)<sup>4)</sup> 김학배(Hag-Bae Kim)<sup>5)</sup>

### 국문요약

최근에 일반 가정에 두 대 이상의 컴퓨터를 네트워크에 연결하여 사용하고 있고, 인터넷이나 네트워크에 독립적으로 운용되던 가전 기기들이 네트워크와 밀접한 관계를 맺으며 발전하고 있다. 이는 과거에 서브 네트워크의 마지막 단계에 해당하던 가정 자체가 하나의 서브 네트워크로 되어가는 추세로 이어진다. 가정 내에 존재하는 다양한 통신규약들을 기존에 구축된 외부 네트워크(인터넷)에 연결하기 위해서는 홈 네트워크를 하나의 서브 네트워크로 하여 이를 외부 네트워크와 연결할 수 있는 RG(Residential Gateway)가 필요하다. 그러나 내장형 시스템인 RG의 하드웨어적인 한계와 IP 부족 문제를 안고 있고 이를 해결하기 위해 RG를 지원하는 관리 서버를 제안하였다. 관리 서버의 시스템 구조를 통해서 하드웨어 문제를 해결하고 각 기기들에 사실 IP를 할당하여 IP 부족 문제를 해결하였다. 원격에서도 웹 브라우저를 이용해서 도메인내의 모든 RG 및 정보가전들을 관리, 감시, 제어가 가능하고 HTTPS 프로토콜을 통해서 통신에 대한 보안 수준을 높이는 웹 기반의 접근 방법을 관리 서버에서 제공한다. <중략> 관리 서버의 UI는 UML을 이용하여 사용자의 역할에 따라 다음과 같이 3가지 모드로 접속하도록 설계하였다. 설계된 3가지 모드는 관리자 모드, RG 매니저 모드, RG 사용자 모드로 구성되었다.

관리 서버는 도메인 내의 RG를 관리하기 위한 프로토콜로 SNMP를 사용하고 있으며 관리 서버와 RG간의 폴링 방법에는 순차적 형태의 폴링 방법, 다중 프로세스를 이용한 폴링 방법, 다중 스레드를 이용한 폴링 방법이 있다. 본 논문에서는 각각의 폴링 방법에 대한 문제점을 제시하고 다중 스레드를 이용한 관리 서버와 RG간의 폴링 방법을 나타낸다.

### 영문요약

Recently, public home have used a more than two computer connected with network, and several home appliances using independently with internet or network are developing to be related closely with the network. Therefore, the home utilized for a simple terminal of the global network in the past is being expanded to another part of the sub network. For a variety of connecting home-area protocols with the existing external network, we require a new Residential Gateway(RG) that does not only make the home-area network operating in the sub network but also connects to the external network.

In this paper, RG has intrinsic limits against flexible service due to IP address assignment and hardware capacity. In order to solve this problem in the RG, we propose a Management Server(MS). The MS that offers the integrated managements and control services for a variety of devices connected the RG in the home-area. It can not only solve the dynamic IP address assigning problem but also assigns private IP addresses to the home network devices through the Network Address Translation(NAT). It also provides somewhat useful functions for the home network and the RG for other additional services. <중략>

The MS is using a SNMP protocol for managing the RG in the domain, a polling method of the MS and the RG compose a sequence polling method, a polling method using a multi-process and a multi-thread. In this paper, we introduce a problem with polling method separately, show a polling method between the MS and the RG using a multi-thread.

한글키워드(영문키워드): 홈 네트워크(Home Network), Residential Gateway, NMS, 정보가전(Intelligent Appliance), 미들웨어(Middleware), 사이버 아파트(Cyber Apartment)

- 1) 정회원 : 연세대학교 대학원 박사과정
- 2) 정회원 : 연세대학교 대학원 박사과정
- 3) 정회원 : 연세대학교 대학원 박사과정
- 4) 정회원 : 극동대학교 정보통신학부 전임강사
- 5) 정회원 : 연세대학교 전기공학과 부교수

논문접수 : 2004. 3. 23.  
심사완료 : 2004. 4. 2.

## 1. 서론

인터넷 이용의 폭발적 증가와 초고속 통신망의 보급으로 홈 네트워크가 일상생활의 중심으로 서서히 자리를 잡아가고 있다. 근래에 일반 가정에 두 대 이상의 컴퓨터를 네트워크에 연결하여 사용하고 있고, 네트워크에 독립적으로 운용되던 가전 기기들이 네트워크와 밀접한 관계를 맺으며 발전하고 있다. 이는, 서브 네트워크의 마지막 단계에 해당하던 가정 자체가 하나의 서브 네트워크로 되어가는 추세로 이어진다.

홈 네트워크의 특징은 일반 네트워크처럼 필요한 선로를 추가로 설비하는 것이 힘들기 때문에 기존의 가정에 설비되어 있는 통신선로(전화선) 혹은 전력선 통신(Power Line Communication)을 이용하여 구축한다. 또한, 새로 건설되는 사이버 아파트에는 기존에 제안된 프로토콜들을 만족하는 선로들을 구축하여 다양한 프로토콜들이 존재하게 된다. 이런 다양한 프로토콜들을 외부 네트워크에 연결하기 위해서는 게이트웨이가 필요하다. 그 중 유/무선 홈 네트워크에 디지털 TV, 냉장고, 세탁기, 전자레인지 등 정보가전 기기를 연동하는 RG(Residential Gateway)에 대한 연구가 활발히 진행되고 있다[1-8].

RG는 가정 내 각종 기기들을 전력선, 전화선, 무선 등을 사용하여 서로 연결하고 이를 외부에 구축되어 있는 ADSL 혹은 인터넷에 연결하기 위해 기존 게이트웨이와 마찬가지로 상호 프로토콜 변환하는[4] 동시에 하나의 서브 네트워크를 대표하는 라우터 역할을 수행한다[1-2]. 또한 RG는 기존의 백색가전이 정보가전으로 진화하는 과정에서 가정과 인터넷을 연결하기 위한 매개체 역할을 수행한다[8].

최근 가전 기기는 가전업체를 중심으로 네트워크 기능을 가진 정보가전으로 진화하고 있다. 이러한 가전 기기들을 통한 홈 네트워크화와 더불어 디지털 콘텐츠 서비스 제공에 대한 요구가 증가되고 있으나, IP 부족 문제와 RG의 내장형(embedded) 시스템이라는 하드웨어적인 문제 등으로 인하여 서비스의 한계를 드러내고 있다[4].

본 논문에서 내장형 시스템인 RG의 하드웨어적인 한계와 IP 부족 문제를 해결하기 위해 RG를 지원하는 관리 서버를 제안한다. 관리 서버는 현재 신축되는 대규모 아파트 단지를 중심으로 각 가정에 개별적으로 설치된 RG를 통합 관리하는 서버로써 공동으로 설치 운영하는 구조를 갖는다. 각 가정 내의 정보가전 및 HA 기기들에 사설 IP를 부여하고 RG를 NAT(Network Address Translator)를 이용하여 제어함으로써 현재 IP 문제를 해결함과 동시에 RG의 기능을 확대하고 보다 많은 부가 서비스를 제공할 수 있다. 또한, 관리 서버는 가정에 보다 확장된 서비스와 기기들을 관리할 수 있는 통로를 제공하며, RG 및 정보가전 기기들의 소프트웨어 업그레이드를 수행할 수 있고 기기들의 상태를 수시로 점검하여 수정하거나 재설정할 수 있다. 사설 IP 상에서도 정보가전 기기들을 외부에서 액세스할 수 있고, 관리 서버를 하나의 단지 혹은 논리적으로 구분되는 하나의 서브 네트워크에 배치하면 향후 관리 및 유지보수가 간편해진다.

관리 서버의 UI는 UML을 이용하여 사용자의 역할에 따라 다음과 같이 3가지 모드로 접속하도록 설계하였다. 설계된 3가지 모드는 관리자 모드, RG 매니저 모드, RG 사용자 모드로 구성되었다. 각각의 모드별로 로그인, 기본 루틴 및 에러처리 과정 등을 클래스 다이어그램을 통해 설계하였다.

관리 서버는 도메인 내의 RG를 관리하기 위한 프로토콜로 SNMP를 사용하고 있으며 관리 서버와 RG간의 폴링 방법에는 순차적 형태의 폴링 방법, 다중 프로세스를 이용한 폴링 방법, 다중 쓰레드를 이용한 폴링 방법이 있다. 본 논문에서는 각각의 폴링 방법에 대한 문제점을 제시하고 다중 쓰레드를 이용한 관리 서버와 RG간의 폴링 방법을 나타낸다. 순차적 형태의 폴링 방법을 사용하면 다음과 같은 문제점이 발생한다. 관리 서버가 RG와 폴링할 때 응답이 오지 않거나, 너무 많은 객체를 폴링하게 되어 데이터를 수신하는 동안에 다음 RG와의 폴링 간격을 만족하지 못하는 경우가 발생하고, 관리 서버가 RG와 통신하는 중에 에러가 발생하면 다른 RG와의 연결 상태도 유지할 수

없으므로 관리 서버의 신뢰성에 영향을 미치게 된다. 다중 프로세스를 이용한 풀링 방법은 자식 프로세스를 생성하여 부모 프로세스의 데이터 공간, 힙(heap), 스택 등의 복사본을 사용하기 때문에 RG의 수가 많을 때 관리 서버에서 많은 메모리를 사용하게 된다. 또한 프로세스 간의 의존성이 존재하기 때문에 하나의 프로세스에 문제가 발생하면 수많은 좀비 프로세스가 발생하기 때문에 시스템에 문제를 발생 시킨다. 다중 쓰레드를 이용한 풀링 방법은 쓰레드 간의 메모리는 전역 메모리를 이용하여 상호 공유하고, 타이머 인터럽트를 이용하여 각각의 쓰레드는 풀링 간격을 보장 받는다.

본 논문의 구성은 다음과 같다. 2장에서는 관리 서버의 구성에 대해서 설명하고, 3장에서는 UML을 이용하여 관리 서버의 설계 및 구현에 대해서 기술한다. 4장에서는 관리 서버와 RG간의 풀링 방법에 대해서 설명하고 마지막으로 5장에서는 결론을 맺고 향후 연구 방향을 제시한다.

## 2. 관리 서버의 구성

본 논문에서 제안하는 관리 서버는 그림 1과 같이 RG, Gate Keeper, VOD 서버 등 각종 서버를 대규모 아파트 단지 혹은 주택단지에 공동으로 설치 운영하는 구조를 제공한다. 관리 서버는 한 지역 내에 설치된 RG 및 RG와 연동된 각 가정 내의 다양한 정보가전 및 HA 기기들을 통합 관리 및 제어한다. 또한 관리 서버와 같은 지역 네트워크상에 연결된 RG, 정보가전 및 HA 기기들을 외부 네트워크(Internet, Wireless)의 유/무선 기기를 통해서 감시, 관리, 제어할 수 있도록 해주는 역할을 한다.

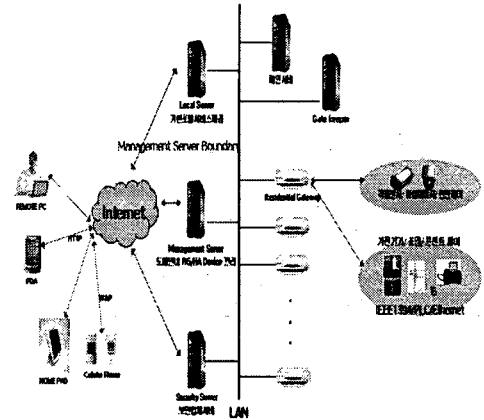


그림 1 관리 서버 배치 구조

웹 브라우저를 통해 원격에서도 관리 서버의 모든 기능을 관리, 감시, 제어가 가능하고 HTTPS 프로토콜을 통해서 통신에 대한 보안 수준을 높이는 웹 기반의 접근 방법을 제공한다[4]. 관리 서버가 있는 도메인내의 RG에 대한 각종 네트워크 사용량에 대한 통계, 접속자 수, 장애 발생 수, 접속 시간 별 트래픽 변화 등 다양한 형태의 통계 데이터를 표와 그래프로 조회할 수 있는 기능을 제공한다. 또한 RG에서 발생하는 각종 로그 정보 등은 관리 서버의 데이터베이스에 저장하고 해당 데이터를 조회할 수 있다. 관리 서버의 시스템 구조는 그림 2에 보여 지듯이 NMS(Network Management System), HAS(Home Automation System), 관리 기능 부분으로 구성되었다.

### 2.1 Network Management System(NMS)

관리 서버의 NMS는 서버 설정과 서버에 등록된 RG 및 기기들에 대한 관리, 감시를 하고 관련 소프트웨어 버전을 관리할 수 있는 기능을 제공한다. 관리 서버는 각 사용자에게 대한 정보와 각 세대별로 설치된 RG, 정보가전 및 HA 기기들에 대한 정보를 바탕으로 접근 권한 테이블(Access Control List)을 제공한다. 그림 3은 접근 권한 테이블에 대한 관리자와 사용자 별로 설정 과정을 나타낸다. 각 기기들의 동작 상태를 조회함으로써 관리자는 필요에 따라 각

기기들의 접근 권한을 그룹 별, 사용자 별로 설정할 수 있다.

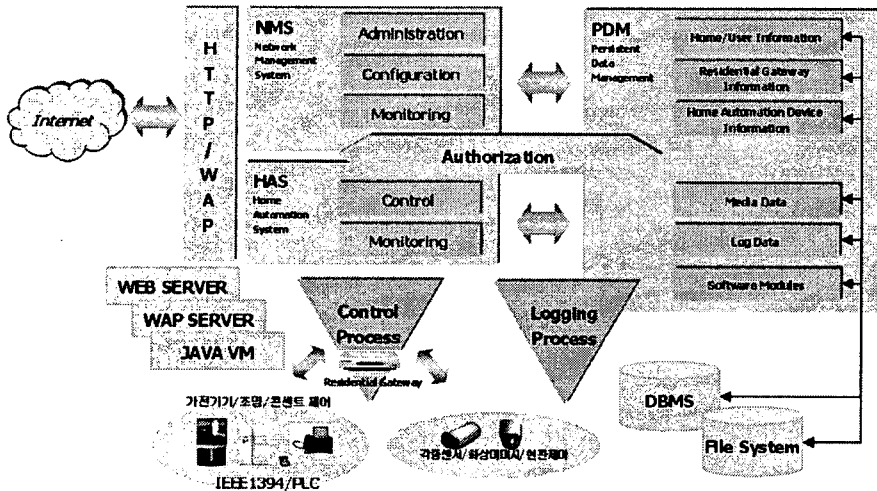


그림 2 관리 서버 시스템 구조

• 주어진 기간 내에 발생한 에러 유형 통계

관리자는 NMS를 이용하여 관리 서버에 사용자 등록 및 접근 권한 테이블을 설정할 수 있다. 세대 내 동작중인 각 기기들을 조회하는 기능을 제공함으로써 관리자는 용도에 따라 각 기기들에 대한 접근 권한을 사용자, 그룹 별로 따로 설정한다. 접근 권한 테이블 설정의 경우 매우 민감하고 복잡하기 때문에 다양한 환경에 대한 기본 값을 미리 제공하여 사용자 편의성을 높인다. 감시 기능을 통해서 관리자는 도메인에 존재하는 RG, 정보가전 및 HA 기기들에 대한 동작 상태를 감시할 수 있다. 이때 통계 정보를 얻어오기 위해서 관리 서버는 주기적으로 RG에 원하는 정보를 요청한다. RG에서 전송된 데이터는 데이터베이스나 파일에 저장되어 있다가 나중에 사용자가 원하는 시점에서 다양한 형태의 통계정보를 표 또는 그래프로 나타낸다. RG로부터 얻은 데이터로 날짜, 이벤트 타입, 기기, 세대별로 조회가 가능하고 이 데이터를 근거로 다음과 같은 통계 자료를 얻을 수 있다.

- 유형별 이벤트 발생 빈도 통계
- 세대별로 자주 사용하는 서비스 유형 통계

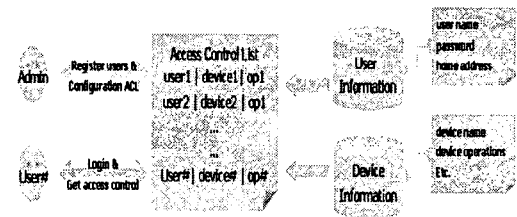


그림 3 접근 권한 테이블의 관리자와 사용자별 설정 과정

기기의 에러 발생이나 보안 장비에 침입 감지 등의 긴급 상황이 발생할 경우 정해진 보안 업체로의 통보 및 관련 유지보수 업체로의 A/S 요청 메시지를 전송한다. 관리 서버는 Master-Slave 구조를 바탕으로 중앙에서 탐다운 방식에 의한 관리 및 제어한다.

### 2.2 Home Automation System(HAS)

사용자는 HAS를 이용하여 RG에 연결된 각종 기기에 대한 제어를 할 수 있다. 관리 서버는 HAS에 자동으로 인식되거나 또는 수동으로 설정된 장치들에 대한 정보를 바탕으로 사

용자가 쉽게 제어할 수 있는 GUI 환경을 제공한다. GUI를 이용한 제어 기능은 각 기기들에 따라 다르게 설정되기 때문에 정적 형태의 기능을 제공한다. 또한 OSGi 프레임워크를 사용할 경우 해당 기기들을 제어할 수 있는 번들(Bundle)을 동적으로 다운로드 받는다. PC, Home PAD, PDA, Cellular Phone 등을 통해 가정 내의 정보가전 및 HA 기기들을 제어할 수 있는 각각의 GUI를 제공하며, 관리 서버는 HTTP 및 WAP 프로토콜을 지원한다. 제어하는 기기들에 따라 원격에서 제어할 수 있는 기능을 제한한다. 예를 들면 방법, 방재 기기들에 대해서는 별도의 핫라인으로 해당 보안업체나 관공서 등에 통보가 갈 수 있도록 구축하였다.

### 2.3 관리 기능

#### 2.3.1 Persistent Data Management(PDM)

관리 서버의 PDM을 통해 도메인 내의 RG, 정보가전 및 HA 기기들에서 발생하는 모든 이벤트 및 로그 데이터를 저장 관리한다. 데이터에는 가정 및 사용자 정보, RG 정보, 정보가전 및 HA 기기 정보, 미디어 데이터, 로그 메시지가 포함된다. 내장형 시스템인 RG는 각종 데이터(로그 메시지, 화상 이미지 등)를 저장할 공간이 별도로 없기 때문에 관리 서버에서 이를 위한 별도의 데이터베이스 공간을 제공한다. 사용자는 관리 서버의 NMS를 이용해서 저장된 데이터를 조회할 수 있다.

#### 2.3.2 소프트웨어 관리

관리 서버는 라이선스를 위한 인증 서버 역할을 한다. 소프트웨어 버전 체크 및 업그레이드 시 RG에 대한 인증을 거치고, 반대로 RG 측면에서는 관리 서버에 대한 인증을 받기 때문에 외부로부터의 해킹을 방지한다. 효과적인 업그레이드를 위해 모듈 별로 버전을 가져갈 수 있는 구조로 설계되었고, 각 버전별로 이력을 관리한다. RG의 버전을 체크해서 최신 버전의 다운로드와 필요한 경우 이전에 사용하던 버전을 다시 다운받을 수 있도록 하였다.

#### 2.3.3 보안 관리

관리 서버는 외부 인터넷 망에서 RG로 직접 접근하는 것을 2차적으로 차단하는 기능을 제공함으로써 보안수준을 높여준다. 1차적인 보안은 같은 서브 네트워크 내의 방화벽이 해주게 된다. 보안에 대한 요구 수준에 따라 침입 탐지시스템(Intrusion Detection System)을 설치함으로써 각 가정의 RG에 대한 불법적인 접근을 사전에 탐지할 수 있도록 한다. 웹으로 접근하는 경우 HTTPS 프로토콜을 사용함으로써 통신상의 데이터를 암호화하며, 텔넷과 같은 접근도 Secure Shell(SSH)을 탑재함으로써 사용자의 로그인ID, 비밀번호를 암호화한다.

### 3. 관리 서버의 설계 및 구현

RG와 통신하는 관리 서버는 그림 5와 같은 구조로 설계되었다. 운영체제로는 Linux, Window NT/2000, Solaris 등 특정한 운영체제에 관계없이 독립적으로 동작되도록 설계되었다. 본 논문에서는 리눅스 기반으로 관리 서버를 구현하였고, 리눅스 커널 2.4.2 버전과 웹 서버로는 아파치 1.3.2를 사용하였다. 그림 5에서 보듯이 운영체제 위에 JVM이 올라가고 관리 서버에서 JSP 파일들을 처리하기 위해 JSP 컨테이너인 자카르타 톰캣 4.0.4을 사용하였다. 그림 6은 사용자로부터 관리 서버에 요청이 들어오고 JSP 컨테이너가 JSP 서블릿을 생성하여 실행하는 과정을 나타낸다.

관리 서버에 사용자의 요청이 들어오면 우선 요청에 해당하는 클래스 파일이 있는지 찾아본다. 만약 클래스 파일이 있다면 해당 파일은 바로 메모리에 적재되고, 적재된 파일은 웹 브라우저를 통해서 사용자에게 보여진다. 사용자가 요청한 클래스 파일이 존재하지 않는다면 JSP 컨테이너는 해당 요청을 서블릿으로 변환하고 컴파일한다. 컴파일 된 클래스 파일은 다시 메모리에 적재되고 사용자에게 보여지게 된다. 관리 서버의 UI는 UML을 이용하여 사용자의 역할에 따라 다음과 같이 3가지 모드로 접속하도록 설계하였다. 설계된 3가지 모드는 관리자 모드, RG 매니저 모드, RG 사용자 모드로 구성되었다.

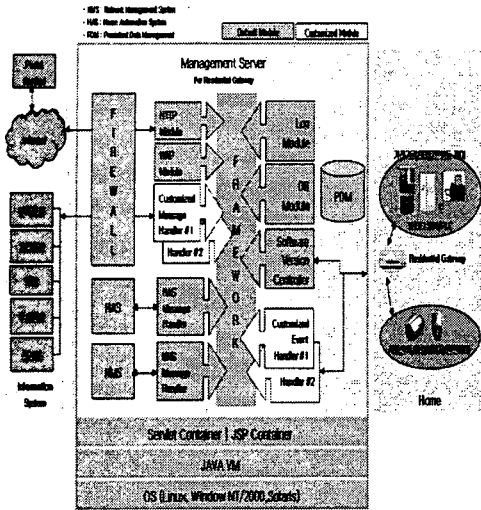


그림 5 관리 서버의 핵심 구조

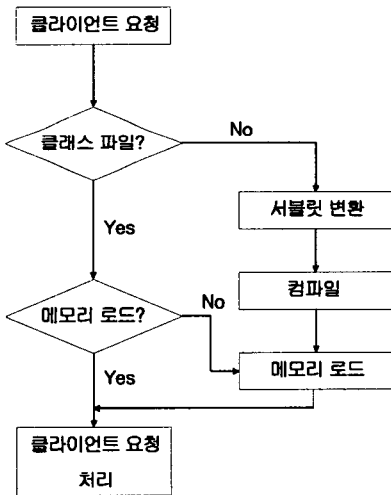


그림 6 관리 서버의 요청 흐름

### 3.1 관리자 모드의 인터페이스

#### 3.1.1 로그인

관리자가 main.htm에 접속하여 ID와 비밀번호를 입력하고, 관리 서버에 접속을 시도하면 login.jsp에서 인증에 대한 절차를 실행한다. login.jsp는 세션을 검증하고, 데이터베이스에 있는 사용자 정보와 웹 접속자가 입력한 ID, 비밀번호가 같은지 체크하기 위해 비밀번호를

암호화한다. 인증을 위해서 UserManager를 통해 UserInfo 즉, 사용자 정보를 데이터베이스로부터 가져와 접속자가 입력한 사항과 비교한다. 접속자가 정당한 사용자라면 RoleManage를 통해 RoleInfo를 가져오고 사용자에게 부여된 권한과 처음 열릴 페이지(Dfscr), 스크린 정보를 가져와 세션에 저장하여 이후 사용에 필요한 작업을 수행하고 Html 페이지로 넘어간다. 만약 인증에 실패하거나 중간에 Exception이 발생하면 error.jsp를 통해 에러 메시지를 접속자에게 제공한다. 그림 7은 관리자 모드의 로그인 클래스 다이어그램을 나타낸다.

#### 3.1.2 에러 처리

JSP에서는 에러 처리를 원하는 JSP 페이지로 넘길 수 있다. 관리 서버의 사용자 인터페이스에서는 각 모드별로 error.jsp를 두었으며, 이 페이지는 각 모드별로 넘어오는 에러를 처리한다. 이를 위해 관리 서버 내부에서 이용하는 MsrgException을 정의하였다. MsrgException의 invalidCode는 해당 세션을 invalid 시킬지 여부를 나타내며, error.jsp에서는 일반 에러 메시지를 사용자에게 제공할지 혹은 로그인 화면으로 나가서 사용자에게 에러 메시지를 보여줄지 여부를 이 코드를 통해 알 수 있다.

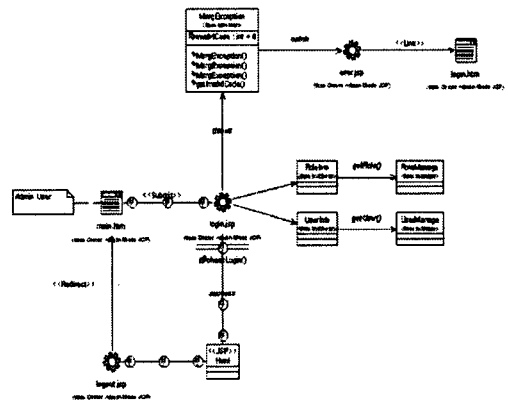


그림 7 관리자 모드의 로그인 과정

#### 3.1.3 관리 서버 설정

그림 8은 관리 서버의 설정 다이어그램을 나타낸다. 관리 서버의 설정 사항은 msrg.xml에 있다. 사용자 인터페이스에서는 이 설정 파일에 있는 사항을 관리자에게 제공하고 관리자는 원하는 경우 해당 설정을 변경할 수 있다. msrgconfig.jsp는 MsrgConfig 클래스를 통하여 MsrgInfo를 구성하고 이 beans에서 구성한 데이터를 JSP를 통해 사용자에게 제공한다. msrgconfig.jsp에서 제공하는 설정 사항을 보고 관리자는 원하는 경우 수정 버튼을 클릭하여 msrgconfigmod.jsp의 수정이 가능한 사용자 인터페이스로 들어갈 수 있다. 이 JSP에서 원하는 값을 수정한 후 적용 버튼을 클릭하면 해당사항이 바로 msrg.xml에 반영된다. 그러나 관리 서버는 설정 파일이 변경되어도 처음 시작할 때 구성된 환경 자체가 변경되는 것이 아니므로 설정 파일의 변경사항이 관리 서버 전체 환경에 영향을 주는 경우 관리 서버는 재부팅 되어야 한다. 따라서 설정 가능케 할 사항과 아닌 사항을 구분하여 사용자 인터페이스에서 제공한다.

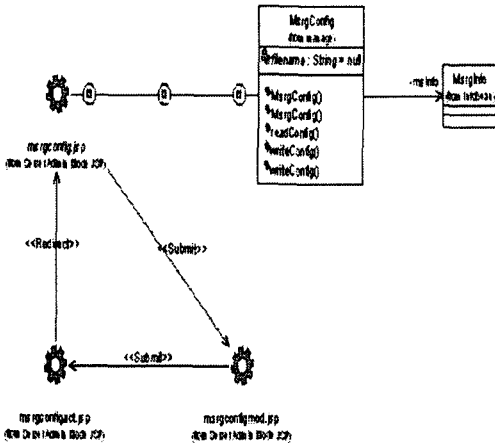


그림 8 관리 서버 설정 다이어그램

### 3.1.4 RG 등록/수정/삭제

RG의 등록, 수정, 삭제에 대한 사용자 인터페이스는 그림 9에 보여진다.

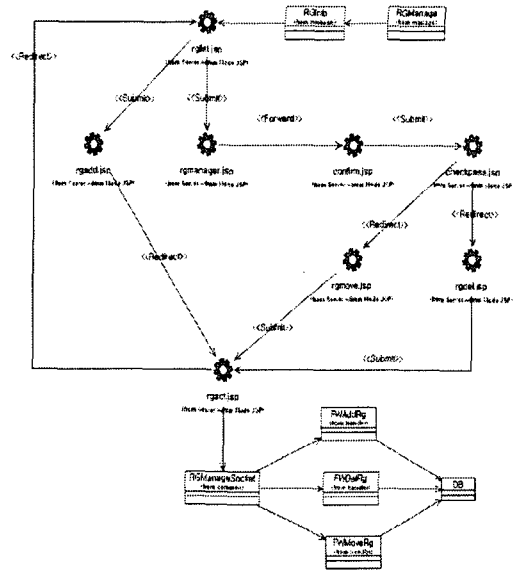


그림 9 RG의 등록, 수정, 삭제 과정

rglist.jsp에서는 RGManage를 통하여 전체 단지 혹은 관리자가 원하는 그룹에 속하는 RG의 리스트를 데이터베이스에서 가져와 관리자에게 보여준다. 새로운 RG의 등록, 기존 RG의 삭제와 등록된 RG에 대해서 상위 그룹의 수정 기능을 제공하며, 이 모든 작업은 manager를 통해서 이루어진다. 웹 환경에서 manager에 직접 접근할 수 있는 방법은 없으므로, 웹에서 manager에 소켓을 열어 통신하는 방법을 이용한다. RGManageSocket은 manager에서 제공하는 웹 포트에 소켓을 연결한다. JSP에서는 RGManageSocket의 addRg(), delRg(), moveRg()를 통해 원하는 작업을 수행한다. manager는 addRg()에 대해서는 FWAddRg 핸들러를, delRg()는 FWDelRg, moveRg()는 FWMoveRg 핸들러를 통해서 manager 내부의 context에 등록된 RG 정보를 수정함과 동시에 데이터베이스의 RG 데이터를 업데이트한다.

### 3.2 RG 매니저 모드의 인터페이스

#### 3.2.1 로그인

관리자 모드의 로그인과 다른 점은 관리자 모드는 데이터베이스에서 데이터를 가져와 인

증을 거치지만 RG 관리자 모드는 RG로부터 데이터를 가져온다는 것이다. RG 관리자가 main.htm 화면에 ID, 비밀번호, 동/호수를 입력하면 login.jsp에서는 해당 동/호수에 따라 데이터베이스에서 RGManage를 통해 RGInfo를 가져오고, 이 정보를 RGSocket에 파라미터로 넣어주어서 실제 해당 RG와 통신하도록 한다. 정상적으로 인증을 거치면 HTML 페이지로 이동되고, 인증에 대한 에러가 발생하면 error.jsp에서 다시 로그인 페이지로 전환된다. 그림 10은 관리자 모드의 로그인 클래스 다이어그램을 나타낸다.

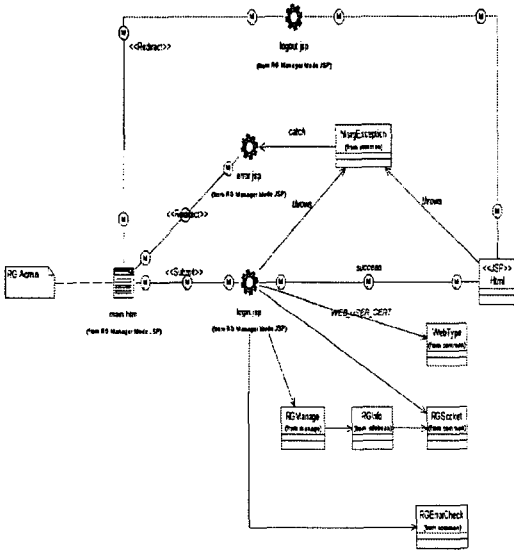


그림 10 RG 매니저 모드의 로그인 과정

### 3.2.2 RG와의 통신

RG와 웹에서 통신하는 방식은 RG 관리자 모드나 사용자 모드 역시 동일한 형식이다. WebType은 프로토콜 문서에 명시되어 있는 각 경우에 대한 타입을 정의하고 있다. RGSocket에 RGInfo와 타입을 지정해 주면 RGSocket은 manager를 통해서 RG와 통신하고, 그 결과 값을 받는다. 결과 값은 RGSocket에서 JSP에 Hash table로 넘기며, Hash table에 대해서 RGErorCheck 클래스를 통해 RG에서 에러가 리턴 되었는지 여부에 따라서 필요

한 Exception을 JSP에 보내고 올바른 값인 경우에는 JSP 페이지에서 필요한 화면을 관리자에게 제공한다.

### 3.2.3 기본 루틴

RG 관리자 모드에서는 RG로부터 모든 데이터를 받아 단순히 관리자에게 편리한 UI를 제공하고 있다. 그 기본적인 루틴은 그림 11과 같다.

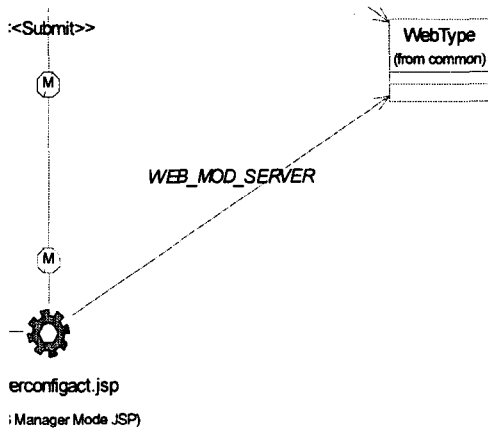


그림 11 RG 관리자 모드의 기본 루틴

가장 기본적인 형식은 그림 8의 서버 설정에 대한 루틴과 같다. 메뉴에서 서버 설정을 클릭하면 serverconfig.jsp에서는 WEB\_REQ\_SERVER 라는 타입으로 RG에서 데이터를 받아오고 이를 화면에 뿌려준다. 필요한 경우 관리자는 해당 설정을 수정할 수 있으며, 실제 수정에 대한 부분을 RG에 적용하기 위하여 serverconfigact.jsp는 다시 WEB\_MOD\_SERVER 타입으로 RG에 접속하여 데이터의 변동사항을 적용하고, serverconfig.jsp 페이지로 전환한다. RG 관리자 모드의 경우, 서버 설정이나 인터넷 환경 설정 등 몇 가지의 작업에 대해서는 설정 적용 후 RG를 리부팅 해줘야한다.

### 3.3 RG 사용자 모드의 인터페이스



### 3.3.1 로그인

RG 사용자 모드에서는 RGUserLogging이라는 클래스를 이용하여 로그인한다. 이 클래스는 모바일 사용자 로그인과 동일한 루틴을 이용하기 위해 만들어진 클래스이다. 로그인 역시, RG를 통해 인증이 되므로 RGSocket을 이용하여 WebType의 WEB\_USER\_CERT 프로토콜 타입으로 RG에 인증을 요청한다. 정상적으로 로그인 되면 JSP 페이지에 연결되고, 에러가 나는 경우 MsrgeException을 발생시키며, 정상적인 로그아웃은 logout.jsp를 통해 이루어진다. 그림 12는 RG 사용자 모드의 로그인 클래스 다이어그램을 나타낸다.

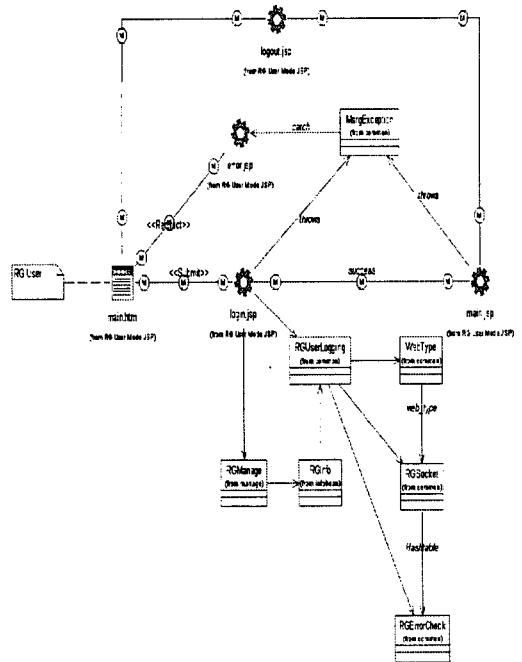


그림 12 RG 사용자 모드의 로그인 과정

### 3.3.2 기본 루틴

RG 사용자 모드의 JSP는 해당 페이지의 기능에 따라 RG로 요청을 전달하고 그 결과를 전달받는다. 이를 위해 RGSocket, WebType 클래스가 필요하다. JSP 다이어그램의 모든 기능들은 그림 10과 같이 RG 매니저 모드와 같은 루틴을 RG와 통신하기 위해 사용하며, 이하 모든 JSP 다이어그램에서는 그 WebType만 보여준다. RGSocket은 JSP에서 전달하는 RGInfo를 기반으로 해당 RG와 Manager가 통신할 수 있도록 한다. WebType은 JPS에서 해당 페이지의 기능에 따른 프로토콜 타입을 정의하고 있다. FromWebWorkRg는 Manager의 핸들러로써 RG와 실제 통신하기 위한 소켓을 열고 RG와의 통신 데이터를 RGSocket에 전달한다. RGErrorCheck는 RG로부터 전달받은 데이터가 정상적인지와 에러가 있는지 여부를 판단하여 에러인 경우 JSP에 에러 메시지를 전달한다.

### 3.3.3 메모/주소록 이용

memolist.jsp에서는 WEB\_READ\_FILE 타입으로 RG에 메모 데이터 파일을 요청하고, 이 파일의 내용을 파싱하여 그 메모 리스트를 submemolist.jsp를 통해 보여준다. 메모 리스트와 각 내용은 사용자의 세션에 저장하여, 추가, 수정, 삭제가 발생 시 즉각 세션의 내용을 업데이트하고, 각 기능이 수행된 결과 값을 WEB\_WRITE\_FILE 타입을 통해 RG에 저장한다. 주소록의 이용 역시 메모 이용과 동일한 루틴을 갖는다.

### 3.3.4 방문자 이미지 조회

방문자 이미지 조회는 visitorlist.jsp에서 WEB\_REQ\_FILELIST 타입으로 RG에서 방문자 리스트를 받아오는 것으로 시작한다. 사용자가 원하는 방문자 정보를 클릭하면 WEB\_REQ\_IMAGE 타입을 이용하여 RG에서 해당 사용자의 이미지를 받아오고, visitorlist.jsp에서는 RG에서 받은 이미지를 converter.Converter를 이용하여 animated GIF로 변형하여 사용자에게 제공한다. 사용자는

방문자를 확인 후, 해당 방문자에 대하여 삭제 기능을 수행할 수 있다.

#### 4. 관리 서버와 RG간의 폴링 방법

관리 서버는 도메인 내의 RG를 관리하기 위한 프로토콜로 SNMP를 사용하고 있으며 관리 서버와 RG간의 폴링방법에는 순차적 형태의 폴링 방법, 다중 프로세스를 이용한 폴링 방법, 다중 쓰레드를 이용한 폴링 방법이 있는데 본 논문에서는 다중 쓰레드를 이용하여 관리 서버와 RG간의 폴링 방법을 사용한다. 다음 절에서는 각각의 폴링 방법에 대한 문제점을 제시하고 다중 쓰레드를 이용한 관리 서버와 RG간의 폴링 방법을 나타낸다.

##### 4.1 순차적 형태의 폴링 방법

관리 서버는 기본적으로 관리하고자 하는 N개의 RG와 폴링을 통하여 원하는 데이터를 수집하여 연산한다. 관리 서버는 첫 번째 RG와 폴링한 후 그 다음 RG와 폴링하여 데이터를 수집하고, 이렇게 N번째 RG까지 폴링한 후 다시 첫 번째 RG와 폴링하게 된다. 순차적 형태의 폴링 방법을 사용하면 다음과 같은 문제점이 발생한다. 첫째, 관리 서버가 RG와 폴링할 때 응답이 오지 않거나, 너무 많은 객체를 폴링하게 되어, 데이터를 수신하는 동안 다음 RG와의 폴링 간격을 만족하지 못하는 경우가 발생하고, 관리 서버가 RG와 통신하는 중에 에러가 발생하면 다른 RG와의 연결 상태도 유지할 수 없으므로 관리 서버의 신뢰성에 영향을 미치게 된다.

##### 4.2 다중 프로세스를 이용한 폴링 방법

순차적 폴링 방법의 문제점은 하나의 관리 서버가 N개의 RG와 통신하여 생기는 문제점이다. 이를 극복하기 위해 관리 서버에 N개의 RG만큼 프로세스를 생성하여 각각의 프로세스가 각각의 RG와 통신하게 함으로서 각각의 RG에 부여된 폴링 간격을 보장하게 된다. 이는 다음과 같은 문제점을 안고 있다. 첫째, 관리 서버에서 자식 프로세스는 부모 프로세스의 데이터 공간, 힙(heap), 스택 등의 복사본을 유

지하게 됨으로 관리하고자 하는 RG의 수가 많을 때 관리 서버에서 많은 메모리를 사용하게 된다. 둘째, Linux에서 프로세스 수는 최대 256개로 정의되어있어 관리하고자 하는 RG의 수는 한계를 가질 수밖에 없다. 셋째, 관리자의 프로세스 간에는 서로의 의존성이 존재함으로 프로세스 관리가 어렵다. 각각의 프로세스는 각각의 RG와 통신하게 된다. 이때 중간 프로세스의 하나만 오류가 발생하여도 수많은 좀비(Zombie) 프로세스가 만들어지게 되어 시스템에 문제를 발생 시킨다.

##### 4.3 다중 쓰레드를 이용한 폴링 방법

본 논문에서 관리 서버와 RG간의 폴링 방법은 다중 쓰레드를 지원하며 쓰레드간의 메모리는 전역 메모리를 이용하여 상호 공유한다. Auto Discovery에서는 관리하고자 하는 RG를 주어진 IP 주소 범위 안에서 찾는다. 찾아진 IP 주소와 RG의 특성을 Thread Initiator에게 정보를 넘겨준다. Thread Initiator는 그 정보를 이용하여 해당 쓰레드를 생성한다. 이때 원하는 폴링 간격에 대한 정보를 쓰레드에게 넘겨준다. 해당 쓰레드들은 관리하고자 하는 객체 함수를 호출하면서 대상 RG와 통신하게 된다. 각각의 쓰레드들은 독립적으로 실행하면서 주어진 정보에 따라 해당 RG들과 통신하고 관리 객체를 수집하여 연산한다. Thread Initiator에서는 해당 RG에서 새로이 생성되는 이벤트가 들어오면 이때 새로운 쓰레드를 생성하고, 삭제 이벤트가 들어오면 해당 IP 주소에 관련된 쓰레드를 삭제하는 기능을 갖는다.

각각의 쓰레드는 타이머 인터럽트를 이용하여 폴링 간격을 보장 받게 된다. 이때 각각의 쓰레드간의 스케줄링 방법은 라운드 로빈 방식으로 한다. Thread Initiator에서는 쓰레드 생성 시기를 조절하기 위해 전체 RG 수를 감안하여 각각의 쓰레드를 생성하게 한다. 이는 각각의 쓰레드가 동시에 폴링 할 수 있는데 이때 전역 메모리의 공유를 지키기 위해 상호 배제 메커니즘을 사용하지만 이는 프로세서의 성능을 저하시키므로, 가능한 한 이런 문제가 생기지 않도록 쓰레드 생성시기를 조절한다. Auto Discovery 기능 수행 후 RG의 이벤트를 수집

하기 위해 TRAP Daemon을 실행 시킨다. RG에서는 미리 정의된 TRAP 특성에 따라 이벤트를 발생시킨다.

## 5. 결론

본 논문은 기존 홈 네트워크의 IP 문제와 내장형 시스템인 RG의 하드웨어 한계를 극복하기 위해 관리 서버를 제안하였다. 가정 내의 정보가전 및 HA 기기들에 사설 IP를 부여하고 RG를 NAT로 제어함으로써 IP 문제를 해결하였다. RG의 하드웨어 한계는 NMS, HAS, PDM으로 구성되어 있는 관리 서버의 시스템 구조를 통해서 극복한다. NMS는 관리 서버 설정 및 관리 서버에 등록된 RG, 정보가전 및 HA 기기들에 대한 관리, 감시 및 관련 소프트웨어 버전 관리 등을 할 수 있는 기능을 제공하고, 사용자는 HAS을 이용하여 RG에 연결된 각종 정보가전 및 HA 기기를 제어한다. 관리 서버의 PDM은 도메인 내의 RG, 정보가전 및 HA 기기들에서 발생하는 모든 이벤트 및 로그 데이터를 저장 관리한다. 원격에서도 웹 브라우저를 이용해서 도메인내의 모든 RG 및 정보가전들을 관리, 감시, 제어가 가능하고 HTTPS 프로토콜을 통해서 통신에 대한 보안 수준을 높이는 웹 기반의 접근 방법을 관리 서버에서 제공한다. 도메인내의 RG에 대한 각종 네트워크 사용량에 대한 통계, 접속자 수, 장애 발생 수, 접속 시간 별 트래픽 변화 등 다양한 형태의 통계 데이터를 표와 그래프로 조회할 수 있는 기능을 제공한다. 또한 RG에서 발생하는 각종 로그 정보 등은 관리 서버의 데이터베이스에 저장하고 해당 데이터를 조회할 수 있다.

관리 서버는 운영체제에 관계없이 독립적으로 동작이 가능하도록 설계되었고 본 논문에서는 리눅스 기반으로 구현되었다. 관리 서버의 UI는 UML을 이용하여 사용자의 역할에 따라 다음과 같이 3가지 모드로 접속하도록 설계하였다. 설계된 3가지 모드는 관리자 모드, RG 매니저 모드, RG 사용자 모드로 구성되었다.

관리 서버는 도메인 내의 RG를 관리하기 위한 프로토콜로 SNMP를 사용하고 있으며 관리 서버와 RG간의 폴링 방법에는 순차적 형태의

폴링 방법, 다중 프로세스를 이용한 폴링 방법, 다중 쓰레드를 이용한 폴링 방법이 있다. 본 논문에서는 순차적 형태의 폴링 방법과 다중 프로세스를 이용한 폴링 방법에 대한 문제점을 제시하고 다중 쓰레드를 이용한 폴링 방법을 해결책으로 제시하였다. 향후 한대의 관리 서버에 몇 대의 RG가 연결될 수 있는지, 각각의 폴링 방법에 대한 성능 평가가 이루어질 것이다.

## 참고 문헌

- [1] T. Saito, I. Tomada, Y. Takabatake, J. Ami, K. Teramoto, "Home gateway architecture and its implementation", consumer electronics IEEE Transactions on, Vol. 46, Issue. 4, pp. 1161-1166, Nov. 2000.
- [2] D. Valtchev, I. Frankov, "Service gateway architecture for a smart home", IEEE communications Magazine, Vol. 40, Issue 4, pp. 126-132, Apr. 2002.
- [3] K. Hofrichter, "The residential gateway as service platform", Consumer Electronics, International Conference on, pp.304-305, June 2001
- [4] J.k Park, I.S Han, J.H Kwon, J. Hwang, H.B kim, S.T Ahn, W. Chang, "Development of a Residential Gateway and a Service Server for Home Automation", Lecture Notes in Computer Science 2402, Springer-Verlag Berlin Heidelberg, pp. 137-149, Aug. 2002
- [5] Dietmar Loy, Atsushi Murase, Andreas Doederlein, "System Level Prototyping for Embedded Networking Applications", 12th International Workshop on Rapid System Prototyping, pp 12-16, 2001.
- [6] K.Y. Cho, K. H. Choi "A novel Architecture of Home Gateway for Efficient Packet Process", Knowledge Media Networking, Proceedings. IEEE

Workshop on, pp.63-67, July 2002,

- [7] J. C. Moon, H. S. Lim, S. J. Kang.  
 "Real-time event kernel architecture for home-network gateway set-top-box (HNGS)". Consumer Electronics, IEEE Transactions on, Vol 45, Issue. 3, pp. 488-495, Aug. 1999
- [8] Cucos. A, Corcoran. P.M, Bigioi. P, Hanrahan. R.B.W, "Home gateway architecture for generic multimedia services", Consumer Electronics, International Conference on, pp. 284-285, June 1999

권진혁:Kwon Jin Hyuck



2002년 국민대학교 전자공학과 졸업(학사)  
 2004년 연세대학교 대학원 전기전자공학과 졸업(공학석사)  
 2004년 - 현재 연세대학교 대학원 전기전자공학과 박사과정

관심분야: 지능형 홈 네트워크, Ad hoc Network. IEEE802.15

민병조:Min Byung Jo



1998년 연세대학교 전기공학과 졸업(학사)  
 2001년 연세대학교 대학원 전기컴퓨터공학과(공학석사)  
 2001년 - 현재 연세대학교 대학원 전기전자공학과 박사과정

관심분야: 인터넷 웹서버 기술, 임베디드 시스템, 정보보안

강명석:Kang Myung Seok



2001년 원광대학교 컴퓨터공학과 졸업(학사)  
 2003년 원광대학교 컴퓨터공학과 졸업(공학석사)  
 2003년 - 현재 연세대학교 대학원 전기전자공학과 박사과정

관심분야: RTOS, Fault-Tolerant, 지능형 홈

네트워크

남 의 석 Nahm Eui Seok



1991년 연세대학교 전기공학과과 졸업.  
 1993년 연세대학교 석사  
 1998년 연세대학교 박사  
 2003년~현재 극동대학교 정보통신 학부 전임강사.

관심분야는 시스템 제어 및 응용, 지능형 모델링.

김 학 배 : Hagbae Kim



1988년 서울대학교 전자공학과 졸업.  
 1990년 미국 미시간대학교 전기 및 컴퓨터 공학과 석사  
 1994년 미국 미시간대학교 전기 및 컴퓨터 공학과 박사

1996년~현재 연세대학교 전기전자공학과 부교수.

관심분야는 실시간 시스템, 인터넷 웹서버 기술, 디지털 시스템 고장포용 및 신뢰도 평가분야.