

실시간 분산 시뮬레이션 시스템을 위한 통신 미들웨어의 구현 및 평가

석진원*, 유인태*, 나원식**

요약

최근에 시뮬레이션 범위의 확대, 고수준의 사용자 요구사항 증가, 시스템 간의 연동 요구로 시뮬레이션 시스템의 구조가 복잡해지고 있다. 따라서 기존의 통신 미들웨어 구조로는 다양한 요구사항의 수용이 어렵다. 본 논문의 선행 연구에서는 기존의 분산 시뮬레이션 시스템에서 실시간 데이터 전달기능을 가진 통신 미들웨어의 문제점을 분석하고, 3계층 클라이언트/서버 구조에 대한 개선된 통신 미들웨어를 디자인 패턴을 사용하여 제안하였다. 그러므로 본 논문에서는 제안한 통신 미들웨어를 개선 및 구현하고 실험을 통하여 기존 시스템과의 성능을 비교하였다.

Implementation and Evaluation of Communication Middleware for Real-Time Distributed Simulation System

Jin-Weon Seok*, In-Tae Ryoo*, Won-Shik Na**

ABSTRACT

Recently, the structure of the simulated system is becoming more complex due to the expanded range of simulations, increase in high-level requirements from users and integration requirement between different systems. Unfortunately, the existing structure of the communication middleware can hardly accommodate recent various requirements. In this paper, previous studies analyzed the problems of communication middleware of the existing distributed simulation system with real-time data transfer capability, and proposed an advanced communication middleware for the three-tier client/server architecture using design patterns. In this paper, the proposed communication middleware has been improvement and implemented, and verified through the tested with existing systems.

Key Words : Simulation, Communication Middleware, Data Transfer & Processing, Distributed System, Design Pattern

* 경희대학교 컴퓨터공학과(✉sjw0176@unitel.co.kr)

** 남서울대학교 교양과정부

· 제1저자(First Author) : 석진원 · 교신저자(Correspondent Author) : 유인태

· 접수일(2010년 8월 17일), 수정일(1차 : 2010년 9월 17일), 게재확정일(2010년 9월 22일)

I. 서 론

분산 시뮬레이션 시스템은 네트워크 기술의 급속한 발전으로 시뮬레이션 분야에 적용되어 나타난 초고속 네트워크 기반의 실시간 시스템이다. 주로 국방, 우주 개발 프로젝트, 공항 및 철도 그리고 원자력 분야 등의 고위험 시스템 등에서 주로 활용되고 있다[1][2].

대규모의 실시간 분산 시뮬레이션 시스템은 초고속 네트워크를 기반으로 높은 실시간성과 신뢰성과 대량의 데이터 전달처리 능력을 가진다. 그리고 정보기술과 시뮬레이션 기법의 변화에 따라서 시스템 확장성, 상호운용성, 편리한 사용성 및 높은 유지보수성을 제공해야 한다. 시뮬레이션 시스템의 핵심 요소는 분야 전문지식과 시뮬레이션 기술이 결합된 실시간 시뮬레이터와 순간적으로 증가하는 대량의 시뮬레이션 데이터를 안정적으로 전달하여 실시간성을 보장해주는 통신 미들웨어이다[3]. 최근의 대규모 국방 분야의 시뮬레이션 시스템 개발에서는 평균 응답시간이 3초 이내 또는 시뮬레이션 대상 객체가 1,000개 이상 등으로 시스템의 처리 능력을 명시하고 있고, 다른 시뮬레이션 시스템과 연동 요구가 계속 증가하고 있어 통신 미들웨어의 중요성은 더욱 커지고 있다. 그러므로 본 연구에서는 우선 기존의 대규모의 군사용 실시간 분산 시뮬레이션 시스템에 적용된 통신 미들웨어의 문제점을 알아본다. 동 연구 분야에 대한 관련 자료 및 디자인 패턴 적용 사례가 많지 않아 기존 운영 시스템 분석과 선행 연구된 디자인 패턴을 적용한 설계 결과를 보완하고, 이를 구현하여 성능을 평가해 본다[2][4][5].

본 논문 2장은 기존 시스템과 선행연구에 대한 분석을 통해 문제점을 도출하고, 3장은 통신 성능 향상을 위하여 도출된 문제점에 대하여 설계내역을 보완하며, 4장은 제한한 통신 미들웨어 시스템을 구현하고 실험을 통하여 기존 시스템과 비교해 본다. 결론에서는 연구결과를 요약하고 향후 연구방향을 제시한다.

II. 관련 연구

2.1 기존 시스템

기존의 실시간 분산 시뮬레이션 시스템은 그림 1과 같이 전통적인 클라이언트/서버 구조로 구성되며, 시스템을 하나의 통합된 구조로 운영하기 위해서는 통신 미들웨어가 필요하다[6][7].

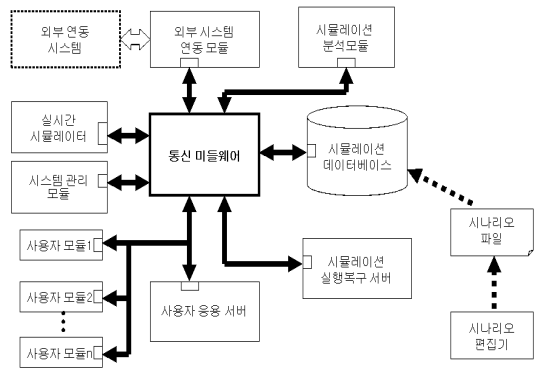


그림 1. 기존 시스템의 일반적인 구조
Fig. 1. General Structure of Existing System

2.2 기존 시스템 및 관련 연구의 문제점

기존 시스템의 통신 미들웨어는 그림 2와 같이 2계층 클라이언트/서버 구조로 구성되었으며, 중요한 실시간 시뮬레이터를 포함한 대부분의 구성요소들이 클라이언트 영역에 위치한다. 이들의 접속 관리 및 실시간으로 데이터 전송은 통신 미들웨어가 서버 영역에 위치하여 동작하는 구조이다[1][4].

구조적인 측면에서 보면, 기존의 통신 미들웨어는 통신 장애와 같은 응급상황이 발생하면 각 구성요소에 전달되어야 할 데이터의 신뢰성 및 처리율이 현저히 저하되는 현상이 발생한다. 최악의 경우에는 중요한 실시간 시뮬레이터가 재시작 되어

야 하는 논리적 모순을 가지고 있다. 시스템의 제작 순서는 먼저 서버 형태의 구성요소 (실시간 시뮬레이터, 사용자 응용 서버, 시뮬레이션 데이터베이스 등)부터 재접속시킨 후 기타 요소 (사용자 모듈 등)들을 연결시키는데 이것은 시스템 운영상 많은 문제를 초래한다.

그리고 구성요소들의 접속관리 측면에서는 단일 접속 구조를 가지고 있어 처리해야 할 데이터가 증가하면 병목현상이 발생하여 급격히 성능이 저하되는 문제를 가지고 있다.

구현 측면에서는 시스템 호출 (Call) 성능이 떨어지는 다중 Wrapper 형태의 ACE_Wrapper 사용이 주요 원인이 되었다. 선행 연구에서 디자인 패턴을 적용한 통신 미들웨어 설계는 확장성, 재사용성 및 유지보수성 등에 치우쳐 구조적인 개선은 되었으나 성능에 대한 개선이 미흡하였다.

그러므로 본 논문에서는 기존 시스템과 선행 연구의 문제점을 보완하여 성능 측면에서도 만족할 수 있는 개선된 통신 미들웨어를 제시하고 구현하여 성능 평가를 하였다[4][5][7][8][9].

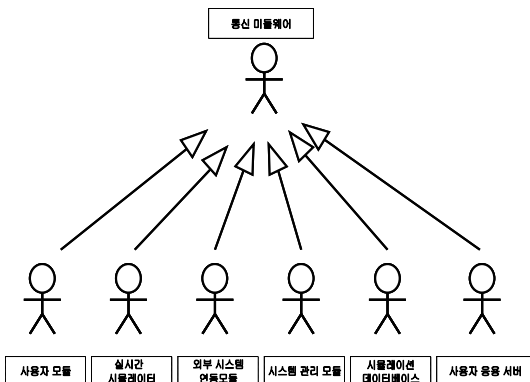


그림 2. 기존의 2계층 통신 미들웨어 구조
Fig. 2. Structure of Existing 2-Layers Communication Middleware

III. 개선된 구조의 통신 미들웨어

3.1 기본 구성

제안한 통신 미들웨어는 3계층의 클라이언트/서버 구조로써 그림 3은 개선된 통신 미들웨어의 논리적 구조를 나타낸다. 여기서 시스템의 확장성 및 위치투명성을 위하여 서버 계층 (실시간 시뮬레이터, 시뮬레이션 데이터베이스 등), 인터페이스 계층 (통신 미들웨어), 클라이언트 계층 (사용자 모듈과 같은 다른 구성요소)으로 나누어 구성하였다.

본 논문에서 디자인 패턴의 적용은 제안하는 통신 미들웨어의 특성에 맞는 어댑터 (Adapter) 패턴, 브릿지 (Bridge) 패턴, 퍼사드 (Facade) 패턴을 선택하여 적용하였다[6][10].

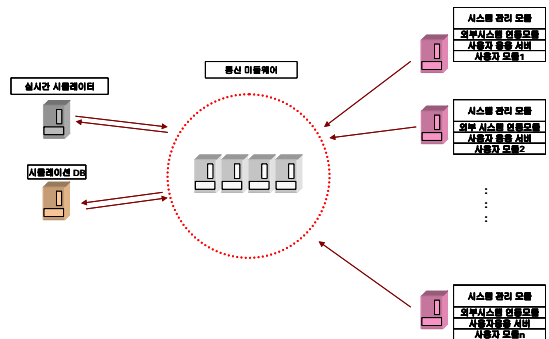


그림 3. 개선된 시스템의 논리적 구조
Fig. 3. Logical Structure of Advanced System

각 구성요소들은 서버 어댑터 (Server Adapter) 및 클라이언트 어댑터 (Client Adapter)와 유기적인 연결이 가능하도록 기존의 단일 접속 구조를 다중 접속 구조로 개선하여 클라이언트가 접속 시 병목현상 해소 및 시스템의 신뢰성을 보장하려고 하였다.

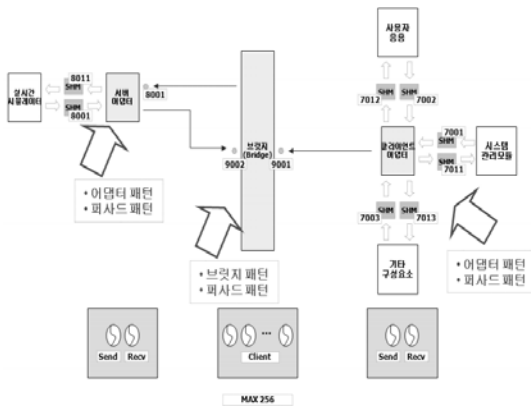


그림 4. 디자인 패턴이 적용된 물리적 구조
Fig. 4. Applying Physical Structure of Design Pattern

3.2 서버 계층

서버 계층에서는 서버와 클라이언트를 엄격하게 분리하기 위하여 어댑터 패턴과 피사드 패턴을 합성 적용하여 실시간 시뮬레이터 등의 다른 구성요소와 독립적인 구성이 가능하게 하고, 향후 확장 및 시스템의 재구성성이 용이하게 개선하였다[11].

서버 어댑터는 실시간 시뮬레이터와 결합되어 서버 기능을 담당하게 되고, 서버 어댑터의 경우 브릿지와 네트워크로 연결되어 논리적 및 물리적으로 분산이 가능한 구조로 구성하여 물리적으로 분산된 하나의 시스템 구조가 가능하다. 그러므로 실시간 분산 시뮬레이션에서 서버는 능동적으로 서버 어댑터에 연결하여 클라이언트들에게 데이터를 전송할 수 있게 되었다. 앞의 그림 4에서 좌측 부분은 서버 계층의 구조를 나타낸다.

3.3 인터페이스 계층

다음은 인터페이스 계층으로 클라이언트 어댑터에 대한 서버로 동작하며 클라이언트 군으로 구분이 가능하다. 또한 서버 어댑터와는 소켓 이중화로

구성하여 서버 군으로도 구분할 수 있다. 서버 어댑터와 클라이언트 어댑터는 브릿지와 유기적인 연동을 위하여 소켓으로 분리하여 단일 접속 구조를 다중 접속 구조로 개선하였다. 이것은 클라이언트 어댑터로부터 전송되는 데이터의 병목현상을 해소하고 동시에 분산 시스템 환경에 적용하기 위한 것이다. 앞의 그림 4의 가운데 부분의 브릿지는 브릿지 패턴과 피사드 패턴을 합성하여 적용한 서버 어댑터, 클라이언트 어댑터 및 브릿지로 구성된 인터페이스 계층의 구조이다.

3.4 클라이언트 계층

클라이언트 계층은 여러 클라이언트 요소들과 결합되어 클라이언트 어댑터 기능을 수행한다. 클라이언트 어댑터는 브릿지와 네트워크로 연결되며 논리적 및 물리적으로 분산될 수 있는 구조로 되어 있다. 클라이언트 어댑터는 서버 어댑터와 구조가 유사하며 피사드 패턴과 어댑터 패턴을 합성하여 적용하였다. 각 구성요소 간에 데이터 전달을 위하여 IPC 및 소켓으로 연동할 수 있도록 하였으며, 각 구성요소의 종류와 상관없이 물리적으로도 분리될 수 있도록 하였다. 앞의 그림 4의 우측 부분은 클라이언트 계층의 구조를 나타낸다.

IV. 제안 시스템의 구현 및 실험

4.1 구현 환경

제안한 실시간 분산 시뮬레이션 시스템을 위한 3계층 통신 미들웨어의 개발 도구 및 실행 환경은 선마이크로시스템즈사의 유닉스 서버 운영체제 환경에서 표준 개발도구(GNU gcc 3.4.2, GNU Make 3.8.0, Standard C++, Standard Socket and Thread, Solaris 9)들을 이용하였으며, 클라이언트에서는 마

이크로소프트사의 윈도우즈 XP 운영체제 환경을 적용하였다.

4.2 실험 환경

기존 시스템과 제안 시스템의 실험 환경은 서버에는 유닉스 계열 운영체제 (솔라리스 9) 환경으로 그리고 클라이언트에는 윈도우 XP 환경에서 그림 8과 같이 구성하여 진행하였다.

구현된 통신 미들웨어의 실험 방법은 기존 시스템과 제안된 시스템의 클라이언트에 테스트 프로그램을 설치하여 10 KB 크기의 메시지를 생성하여 서버의 실시간 시뮬레이터에 1000 개의 메시지를 송신하여 전송에 소요된 시간과 실시간 시뮬레이터 (서버)에서 메시지 수신 후 수신 메시지를 생성하여 테스트 프로그램 (클라이언트)으로 전송하여 수신된 시간을 측정하였다.

표 1. 기존 및 제안 시스템의 성능 실험 결과

Table 1. Performance Test Result of Existing and Proposed System

구분	시행횟수	단위 : usec									
		1차	2차	3차	4차	5차	6차	7차	8차	9차	10차
기존의 시스템	1회	14,159,887	14,181,487	19,220,659	13,730,544	13,190,595	21,543,032	13,844,242	13,009,505	14,190,322	14,050,889
	2회	15,260,046	14,051,264	13,336,978	13,629,791	14,867,180	14,912,824	13,356,008	13,881,947	14,129,326	14,549,090
	3회	14,190,384	13,083,836	14,101,039	13,159,551	14,048,587	13,487,651	15,008,072	13,298,496	13,840,223	13,418,600
	4회	14,629,150	20,903,884	14,856,374	13,229,619	13,239,328	14,181,839	13,429,738	13,913,234	13,798,804	14,264,785
	5회	14,089,850	13,613,400	14,294,269	13,781,705	21,630,559	14,116,753	13,322,827	13,530,204	19,901,345	13,304,701
	6회	14,659,516	14,322,525	15,239,518	15,320,704	14,378,915	14,981,687	14,082,313	14,773,486	14,869,427	14,812,412
	7회	15,270,836	13,912,543	15,069,010	19,667,971	13,589,626	13,603,168	13,396,517	14,125,842	13,359,396	22,787,754
	8회	13,999,263	14,053,529	13,729,933	13,542,757	16,179,554	15,122,266	13,639,342	13,870,107	14,938,581	14,174,506
	9회	14,340,619	14,557,662	13,142,587	13,436,044	14,183,577	13,838,855	13,252,285	13,631,642	13,892,895	13,527,555
	10회	17,508,946	15,582,124	13,446,535	18,052,495	15,324,873	12,859,772	13,757,681	13,699,141	13,668,181	13,568,004
	평균시간	14,810,830	14,826,225	14,643,690	14,755,118	15,063,279	14,864,785	13,708,903	13,773,360	14,658,850	14,845,830
MB/s	0.7079792	0.7072441	0.7160599	0.7106524	0.696114	0.7054095	0.7648869	0.7613073	0.7153194	0.7063101	
제안된 시스템	1회	2,019,863	918,401	1,279,351	3,119,644	1,118,662	2,020,292	2,408,999	2,512,056	2,109,958	1,655,241
	2회	1,255,726	3,761,746	1,667,219	3,789,861	1,831,094	2,511,578	3,618,461	1,858,118	2,894,306	1,004,926
	3회	1,772,852	1,728,333	1,519,867	1,129,355	1,399,507	1,348,191	1,800,041	1,669,930	1,921,027	2,496,628
	4회	770,776	1,411,615	1,511,163	1,543,298	1,115,810	2,519,001	2,459,633	2,040,702	1,233,067	2,748,411
	5회	1,310,897	1,648,415	1,940,056	2,149,028	2,394,280	1,312,119	1,347,798	1,870,670	2,330,689	1,557,021
	6회	1,662,346	1,200,767	2,539,826	2,677,465	1,380,144	2,188,286	2,612,264	2,187,705	1,519,327	2,188,694
	7회	1,976,843	1,543,139	2,087,653	1,942,216	1,381,405	2,589,344	1,909,659	1,942,736	2,810,308	1,860,623
	8회	2,459,321	1,809,984	1,492,420	1,681,455	4,428,413	1,872,554	2,983,856	1,028,244	1,241,051	2,218,986
	9회	1,689,138	1,746,362	1,716,903	1,477,964	2,650,341	2,467,436	1,086,529	2,287,844	3,307,688	1,268,073
	10회	1,841,988	2,821,792	1,519,618	1,860,799	1,748,944	1,841,096	2,059,237	1,420,642	2,165,106	2,372,661
	평균시간	1,675,955	1,859,055	1,727,408	2,137,109	1,945,860	2,066,990	2,228,628	1,881,865	2,153,253	1,937,126
MB/s	6.256588	5.6403688	6.0702292	4.9065174	5.3887536	5.0729619	4.7050299	5.5720053	4.8697303	5.4130489	

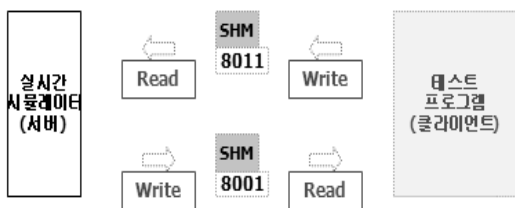


그림 8. 실험 환경 및 방법

Fig. 8. Test Environment and Method

실험은 모두 10 차례에 걸쳐서 기존 시스템과 제안 시스템에서 매차마다 10 회씩 실시 (전체 100 회) 하였다. 실험 결과는 표 1에 기존 시스템과 제안 시스템으로 분류하여 나타내었다.

4.3 결과 분석

본 연구에서는 기존의 대규모 시뮬레이션 시스

템의 통신 방식과 제안한 방식에 대한 실험 결과를 이용하여 평균 전송시간과 평균 전송량으로 구분하여 비교해 보았다. 그림 9는 실험결과에 대한 기존 시스템과 제안한 방법을 적용한 시스템의 성능을 비교한 결과를 보여준다. 여기서 제안한 시스템은 기존 시스템과 단순 성능 비교만으로도 약 6~7 배 정도의 시스템 성능이 향상된 것을 볼 수 있다.

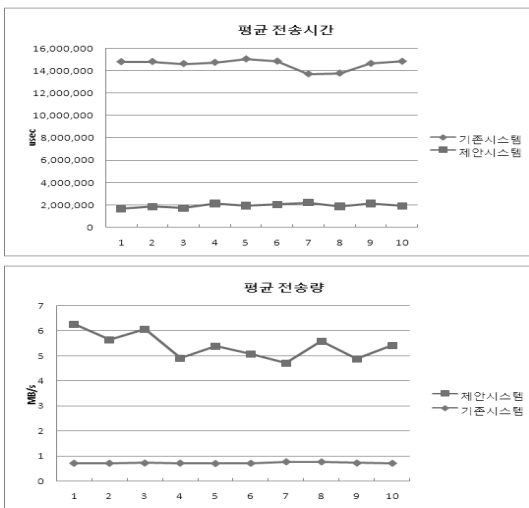


그림 9. 기존 시스템과 비교
Fig. 9. Compared with the Existing System

결과적으로 이러한 성능 비교를 통하여 제안한 시스템은 기존 시스템의 열악한 통신 구조를 개선 가능하다는 것을 알 수 있다. 그리고 기존 시스템의 통신 구조가 얼마나 열악한 시스템 구조인지, 그리고 얼마나 불안정한 시스템인지 역으로 유추해 볼 수 있다.

V. 결 론

본 연구는 기존의 실시간 분산 시뮬레이션 시스

템의 통신 미들웨어 및 선행 연구 결과를 기반으로 구조를 개선하고 구현하여 성능실험을 통하여 기존 시스템과 제안 시스템에서 평균 전송속도와 평균 전송량을 비교해 보았다.

본 연구를 통하여 구현한 실시간 분산 시뮬레이션 시스템을 위한 통신 미들웨어는 실시간 데이터 전달처리 과정에서 시스템 구조 개선에 의한 병목 현상의 해소로 시스템 성능을 크게 향상시키고 있음을 확인할 수 있었다. 그러나 실시간 분산 시뮬레이션 시스템은 구조 특성상 아직 많은 부분에서 개선이 필요하다. 향후에는 시스템이 외부 변화에 적절하게 적응하고 안정성과 신뢰성 향상을 위하여 실시간 스케줄러 개선 및 구성요소 들의 접속 상황 인지방법 등에 대한 연구가 더 필요하다.

참고문헌

- [1] Amjad Umar, *Distributed Computing and Client Server Systems*, Prentice Hall Inc, 2005.
- [2] Bernard P. Zeigler, Herbert Praehoffer and Tag Gon Kim, *Theory of Modeling and Simulation : Integrating Discrete Event and Continuous Complex Dynamic Systems*, Academic Press, 2000.
- [3] Frederick Kuhl, Richard Weatherly and Judith Dahmann, *Creating Computer Simulation Systems*, Prentice Hall Inc, 1999.
- [4] 석진원, 유인태, "소프트웨어 디자인 패턴을 적용한 실시간 분산 시뮬레이션을 위한 데이터 전달처리 시스템 설계", *디지털콘텐츠학회논문지*, 제 10 권, 제 4 호, pp. 649-658, 2009.
- [5] 석진원, 나원식, "실시간 분산 시뮬레이션 시스템을 위한 통신 미들웨어", *한국지식정보기술학회 2010 춘계 학술 발표대회논문집*, 2010.
- [6] 이강택, "침입감내기술 기반의 보안시스템 설계 및 구현 : 위게임 시스템 중심으로," *경기대학교박사학위논문*, 2006.
- [7] 임치훈, "위게임 시스템을 위한 디자인 패턴의 합성 및

- 적용," *한밭대학교정보통신대학원석사학위논문*, 2008.
- [8] 구성환, "무선PBX 시스템을 위한 객체기반 통신망 관리 시스템의 설계 및 구현," *서강대학교석사학위논문*, 2004.
- [9] 에릭 프리먼 외 3인 공저, 서환수 역, *Head First Design Patterns*, 한빛미디어, 2005.
- [10] <http://www.cs.wustl.edu/~schmidt/ACE-papers.ht ml>
- [11] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns : Elements of Reuseable Object-Oriented Software*, Person Education Inc, 1995.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2010-(C1090-1011-0001))

석진원(Jin-Weon Suk)



1990년 금오공과대학교 전자공학과 (컴퓨터공학) 졸업(공학사)
 1996년 국방대학원 전자계산학과 졸업(국방과학석사)
 2004년 경희대학교 일반대학원 컴퓨터공학과(박사수료)

1990년~2003년 : 국방부(공군) 군수 및 정보체계 담당
 2003년~2009년 : 포스데이터(주)/SKC&C(주) 부장
 2009년~현재 : (주)비전엔바이오테크 이사
 ※ 관심분야: 시뮬레이션, RFID/USN, 그리드 네트워크

유인태(In-Tae Ryoo)



1987년 연세대학교 전자공학과 (공학사)
 1989년 연세대학교 본대학원 전자공학과 (공학석사)
 1994년 연세대학교 본대학원 전자공학과 (공학박사)

1997년 The Univ. of Tokyo 전자정보통신전공 (Ph.D.)

1997년~1999년 : 삼성전자 정보통신총괄 선임연구원
 1999년~현재 : 경희대학교 전자정보대학 교수
 ※ 관심분야: 인터넷 기술, IPTV, 멀티미디어 통신

나원식(Won-Shik Na)



2005년 경희대학교 대학원 컴퓨터공학과 (공학박사)
 2001년 3월~2003년 2월 : (주)성신 섬유 전산실장
 2006년 3월~현재 : 남서울대학교 교양과정부 교수 (컴퓨터 계열)

※ 관심분야: 네트워크 보안, 무선 LAN, 의료정보, 전자제어