

Publish/Subscribe 모델 기반 RPC 미들웨어 설계와 구현 Design and Implementation of Publish/Subscribe Model Based RPC Middleware

박 상 현*, 최 준 성*, 국 광 호***

Sanghyun Park*, Junesung Choi*, Kwangho Kook***

Abstract

Various types of middle-ware were created for integration between legacy systems and newly built systems. RPC(Remote Procedure Call), MOM(Message Oriented Middle-ware) and TM(Transaction processing Monitor) are the typical types of middle-ware. One of the most known MOM type middle-ware is PS(Publish/Subscribe). PS enables to create a system which has low coupling and high scalability. But PS based systems also have low cohesiveness. On the contrary, RPC has high cohesiveness but also has high coupling. This paper proposed design and implementation of hybrid model which offset disadvantages of RPC and PS.

요 약

다양한 형태의 미들웨어가 레거시 시스템과 신규 시스템의 통합을 위해 고안되었다. 미들웨어의 대표적인 형태로는 RPC(Remote Procedure Call), MOM(Message Oriented Middleware), TM(Transaction processing Monitor) 등이 있다. MOM 방식의 대표적인 모델인 PS(Publish/Subscribe)는 시스템의 결합도(coupling)가 낮고 낮은 결합도로 인해 높은 확장성(Scalability)를 얻을 수 있다는 장점이 있으나 한편으로는 응집도(cohesiveness) 또한 낮아진다는 단점이 있다. 이에 비해 RPC 방식은 응집도가 높으나 결합도 역시 높다는 단점이 있다. 본 논문에서는 RPC와 PS의 단점을 서로 상쇄시킬 수 있는 하이브리드 모델의 설계와 구현 방안을 제안한다.

Key words : middle-ware, publish, subscribe, remote procedure call, data distribution service

1. 서론

컴퓨팅 자원이 고가의 희소 자원이던 시기에는 컴퓨터의 유휴 시간을 활용하여 다수 단말을 운용하는 서버/클라이언트 방식으로 정보 시스템을 구축하였으나,

하드웨어의 기술의 발달로 컴퓨팅 자원이 저렴해지면서 서버 역할을 클라이언트가 분담하기 시작했다[1]. 1990년대에 이르러서는 인터넷 붐과 함께 서버/클라이언트의 역할 구분 없이 정보시스템이 동등한 수준의 노드로 구성되는 그리드(Grid) 방식이 등장하였다

* C4I R&D Center, R&D Division, Hanwha Thales Co. Ltd. (seanlab@gmail.com, 031-8020-7000)

** Seoul National University of Science & Technology

★ Corresponding author

※ Acknowledgment : This study was financially supported by Seoul National University of Science & Technology. Manuscript received Jul. 23, 2015; revised Aug. 11, 2015 ; accepted Aug. 17, 2015

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

[2]. 정보시스템 발전과정에서 많은 레거시 시스템이 발생했고, 레거시 시스템과 신규 시스템의 통합을 지원하는 다양한 모델의 미들웨어도 필요하게 되었다 [3]. 분산컴퓨팅 환경 미들웨어의 대표적인 방식에는 RPC(Remote Procedure Call), MOM(Message Oriented Middle-ware), TM(Transaction processing Monitor) 등이 있다[4]. TM은 분산시스템의 트랜잭션 감시 목적의 미들웨어로, 통신용 미들웨어인 MOM과 RPC와는 그 성격이 다르다[4]. MOM을 기반으로 구축된 시스템은 결합도(Coupling)가 낮아 구성요소간의 의존성을 떨어뜨리지만, 응집도(cohesiveness)가 낮기 때문에 유지보수성이 나빠진다는 단점이 있다 [5]. 반면, RPC 기반의 시스템은 응집도는 높지만 결합도 또한 높기 때문에 구성요소간의 의존성이 높아 시스템의 확장에 한계를 가지고 있었다[5]. 그러므로 이런 단점을 해결하기 위한 미들웨어가 필요하다.

본 논문에서는 MOM이 가진 저결합도와 RPC가 가진 고응집도를 겸비하고 있는 하이브리드 미들웨어를 설계/구현하고 제안 시스템의 성능을 검증하였다.

II. 관련 연구

1. PS 통신 모델

MOM 미들웨어 세부 종류에는 P2P(Point-to-Point), MQ(Message-Queue), PS(Publish/Subscribe) 등이 있으나, P2P와 MQ는 서버-클라이언트 방식으로 동작하기 때문에 분산 시스템 내의 모든 노드를 독립적인 개체로 간주하는 PS 모델 기반의 분산 시스템에 비해 결합도가 높다.

PS(Publish/Subscribe)는 토픽(Topic)을 중심으로 익명의 정보 생산자(출판자 : Publisher)와 정보 소비자(구독자 : Subscriber)를 연결하는 통신 모델이다[6]. PS 모델에서는 출판자가 데이터를 송신한다는 의도를 선언한 후 데이터를 출판하고, 구독자가 데이터를 수신하겠다는 의도를 선언하면 출판된 데이터가 출판자에서 구독자에게로 전달된다. PS 모델은 실시간성을 필요로 하는 정보 전달에 적합하다[6-7]. PS 모델은 그 이름에서 알 수 있듯이 데이터의 송신자와 수신자의 역할이 분명하게 나뉘어 있어 시스템의 결합도(coupling)가 낮고, 낮은 결합도로 인해 높은 확장성(Scalability)를 얻을 수 있다는 장점이 있으나 한편으로는 응집도(cohesiveness) 또한 낮아진다는 단점이

있다[5]. PS 모델은 금융 시스템, IoT(Internet of Things), 로봇, 무기 체계 등 실시간 데이터 공유 기능이 요구되는 곳에 사용되고 있다[8-9].

[그림 1]은 PS 통신모델의 구조를 보여주고 있다. PS 방식의 통신 미들웨어로는 DDS(Data Distribution Service), JMS(Java Messaging Service), MQ(Message Queue) 등이 있다[7-8].

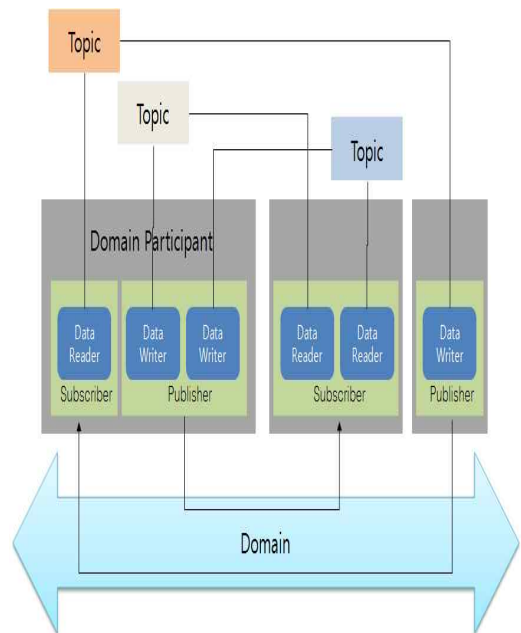


Fig. 1. PS Communication Model

그림 1. PS 통신 모델

2. RPC 모델

RPC(Remote Procedure Call)는 프로세스간 통신 방식 중 하나로써 소프트웨어가 네트워크 너머에서 동작하는 소프트웨어의 프로시저를 호출할 수 있도록 한다[6-8]. 프로그래머가 RPC를 이용하여 분산 시스템의 클라이언트 소프트웨어를 작성할 때는 로컬 프로세스의 프로시저를 호출하는 것과 동일한 방식으로 코딩을 한다[3,6-8]. 클라이언트가 호출하는 프로시저의 매개변수를 네트워크를 통해 전달하고 반환되는 값을 받아오는 일은 RPC 라이브러리 또는 미들웨어가 이를 대신하여 수행하게 된다[6]. 서버 소프트웨어 역시 로컬 프로세스에 있는 호출자가 이용하는 것과 동일하게 함수를 구현해 놓으면 RPC 라이브러리가

클라이언트로부터 온 프로시저 호출정보를 이용하여 대리 호출을 수행한 후, 그 결과를 다시 클라이언트에게 반환한다[6].

RPC 모델을 취하는 미들웨어에는 CORBA(Common Object Request Broker Architecture), RMI(Remote Method Invocation), .NET Remoting, XML-RPC, thrift, protocol buffer 등이 있다[6-8]. RPC 모델은 프로시저 호출자에게는 클라이언트 스텝(Stub)을, 프로시저에게는 서버 스텝을 대리자로 할당함으로써 단일 소프트웨어의 구조를 그대로 분산 시스템에 적용한다[3,6]. 이 모델에서는 클라이언트 프로그래머가 호출할 프로시저가 로컬에 구현되어 있다고 간주하고 코딩을 수행하며 서버 프로그래머는 호출자가 로컬에 있다고 간주하여 구현하므로, 시스템의 응집력(cohesiveness)은 상승하나 결합도(coupling) 역시 상승한다[4,5]. [그림 2]은 RPC 통신 모델의 구조를 나타낸다. PS 모델과 RPC 모델 모두 결합도와 응집도 간의 트레이드 오프(Trade-Off)를 해야 한다[5].

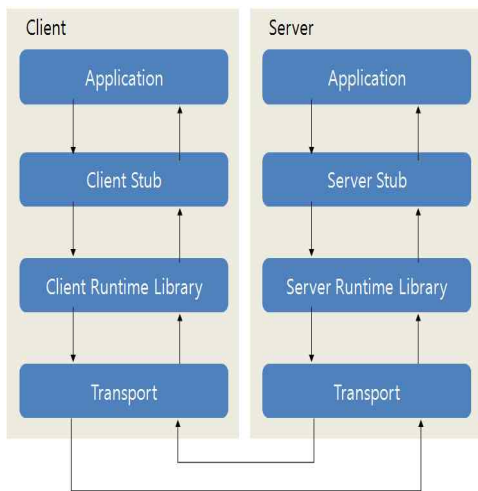


Fig. 2. RPC Communication Model
그림 2. RPC 통신 모델

III 제안 시스템 구현

1. PS-RPC 미들웨어의 구조

PS 모델은 다양한 성격의 임무를 수행하는 서브 시스템들에게 실시간으로 데이터를 공유해야 하는 시스템에서는 낮은 결합도를 제공하는 PS 모델이 적합하

나, 미사일 발사 명령, 금융 거래 체결과 같은 치명적 임무 기능 구현에는 높은 응집도를 제공하는 CORBA가 적합하다[10]. PS-RPC는 PS로 구성되어 있는 분산 시스템에 부분적으로 응집도를 높여야 하는 일부 구간에 RPC를 적용할 수 있도록 제안하는 하이브리드 형태의 미들웨어이다. PS-RPC 미들웨어는 프로시저 호출을 DDS Topic으로 변환하는 스텝(Stub) 계층과 Topic을 네트워크를 통해 송수신하는 DDS 계층으로 구성하게 된다. 스텝 계층은 원격 프로시저를 구현하는 서버 스텝과 이 원격 프로시저를 호출하는 클라이언트 스텝으로 나뉜다. 클라이언트 스텝은 프로시저 호출 요청 토픽을 송신하는 출판자와 호출 응답 토픽을 수신하는 구독자를 가지며, 서버 스텝은 프로시저 호출 요청 토픽을 수신하는 구독자와 호출 응답 토픽을 송신하는 출판자를 가진다. [그림 3]은 PS-RPC의 구조를 나타낸다.

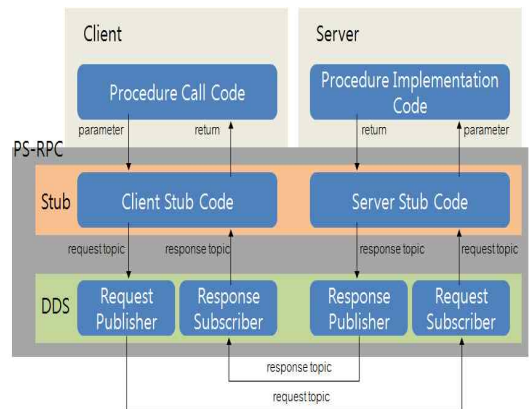


Fig. 3. PS-RPC Communication Model
그림 3. PS-RPC 통신 모델

이들 스텝 코드는 [그림 3]에서 나타나는 바와 같이 PS-RPC 모델러에 의해 IDL(Interface Definition Language) 파일과 함께 생성된다. 클라이언트 소프트웨어에서는 PS-RPC 모델러가 생성한 코드에서 클라이언트 스텝과 IDL을 이용하고 서버 소프트웨어에서는 서버 스텝과 IDL을 이용해서 원격 프로시저를 구현한다.

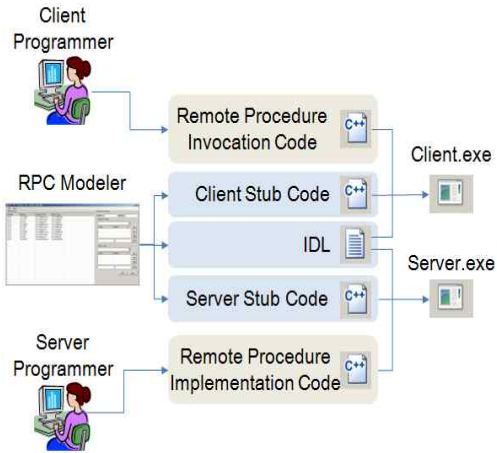


Fig. 4. PS-RPC base distributed system build process
 그림 4. PS-RPC 기반의 분산 시스템 구축 과정

2. PS-RPC 모델러 프로시저 설계

[그림 5]는 PS-RPC 모델러의 프로시저 등록 창이다. 상단의 [Module]은 프로시저가 선언될 클래스의 이름, [Procedure]는 프로시저의 이름을 나타낸다. 하단에 있는 [Request Topic] 항목은 프로시저의 입력 매개변수이자 요청 메시지의 각 필드를 입력하는 기능을 하며, [Response Topic] 항목은 프로시저의 출력 매개변수이자 응답 메시지의 각 필드를 입력하는 기능을 한다.

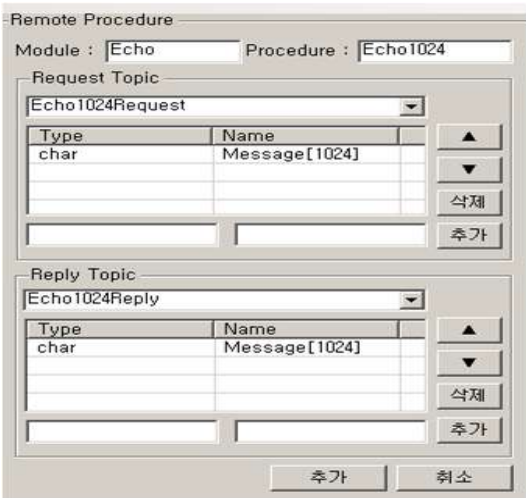


Fig. 5. Remote procedure modeler
 그림 5. 원격 프로시저 모델러

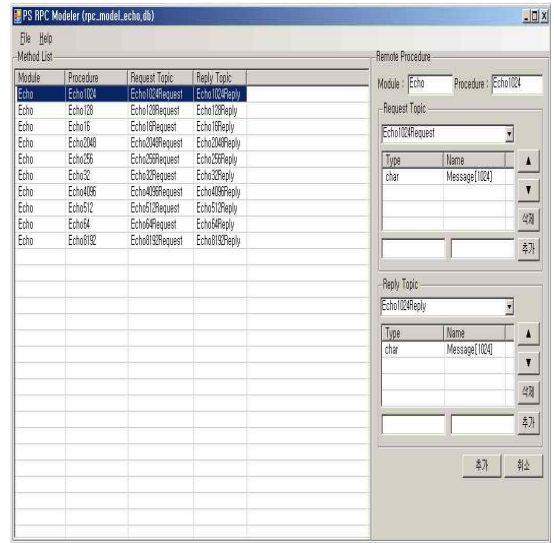


Fig. 6. Remote procedure list of PS-RPC Modeler
 그림 6. PS-RPC 모델러의 원격 프로시저 목록

입력 완료된 원격 프로시저의 정보는 [그림 6]과 같이 PS-RPC 모델러의 주 윈도우에 나타난다.

3. 토픽 IDL 및 클라이언트/서버 스텝 코드 생성

원격 프로시저 설계가 완료되면 [그림 7]과 같이 PS-RPC 모델러의 [File]->[Generate IDL & Stub]메뉴를 통해 IDL과 클라이언트/서버 스텝 코드를 생성한다.

4. Server Side 구현

서버 어플리케이션은 이 순수 가상 클래스를 상속하여 구현한 클래스의 인스턴스를 PS-RPC 모델러가 생성한 EchoServer의 생성자에 매개변수로 넘겨야 한다.

PS-RPC 모델러가 생성한 서버 스텝 코드에는 [표 1]과 같은 순수 가상 클래스 IEcho가 선언된다. IEcho 순수 가상 클래스는 서버의 원격 프로시저 구현체와 클라이언트의 호출자 사이의 인터페이스 역할을 한다.

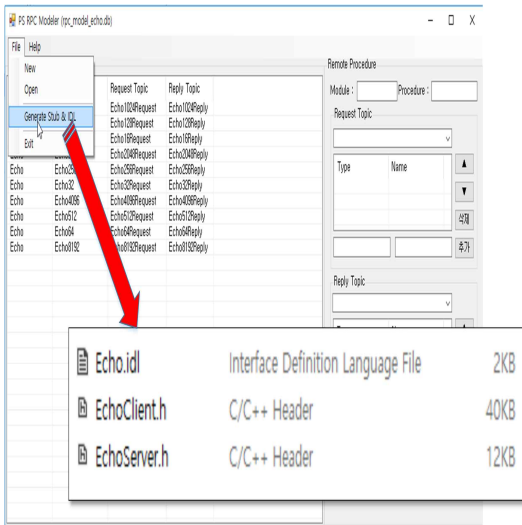


Fig 7 Client/Server Stub Code Generation
 그림 7. 클라이언트 서버 스텝 코드 생성

Table 1. PS-RPC Modeler Generated Server Stub
 표 1. PS-RPC 모델러가 생성한 서버 스텝 코드

```

INTERFACE IEcho
{
    Echo1024( Echo1024Request )
}

CLASS EchoServer
{
    RegisterServant( IEcho )
        Echo = IEcho
    EXIT

    OnMessageEcho1024( Echo1024Request )
        Echo1024Reply =
        Echo.Echo1024( Echo1024Request )
        DDS_SEND Echo1024Reply
    EXIT
}
    
```

IEcho 순수 가상 클래스를 상속하여 원격 프로시저를 구현하는 파생 클래스(EchoImpl)와 이 파생 클래스의 인스턴스를 생성하고 호스팅하는 서버 어플리케이션의 코드는 [표 2]와 같다.

Table 2. Implementation Of Server Application
 표 2. 서버 응용 프로그램의 구현 코드

```

CLASS EchoImpl IMPLEMENTS IEcho
{
    Echo1024( Echo1024Request )
        Echo1024Reply = Echo1024Request

    RETURN with Echo1024Reply
}

SERVER-APPLICATION()
    EchoServer.RegisterServant( EchoImpl )

    START_DDS
    RUN_EVENT_LOOP
    STOP
    
```

5. Client Side 구현

클라이언트는 원격 프로시저와 원격 프로시저를 호스팅하는 기능의 구현이 요구되는 서버와는 달리 클라이언트의 요구사항은 원격 프로시저 호출 뿐이다. 토픽을 발행하고 구독하는 코드는 PS-RPC 모델러가 자동으로 생성한 스텝 코드 안에 감춰져 있다. [표 3]의 코드는 클라이언트에서 로컬 함수를 호출하는 것과 다름없이 서버에 구현되어 있는 Echo1024() 프로시저를 호출하는 과정을 보여준다.

Table 3. Implementation Of Client Application
 표 3. 클라이언트 응용 프로그램의 구현 코드

```

CLIENT-APPLICATION()

    START_DDS

    Echo1024Request = "HELLO"

    Echo1024Reply =

        Echo1024( Echo1024Request )

    STOP
    
```

IV 제안 시스템의 성능 분석

제안하는 PS-RPC 미들웨어의 성능 비교를 위한 시험 환경은 네트워크 I/O가 시험의 병목 구간이 되지 않도록 기가비트 인터넷으로 서버와 클라이언트를 연결하여 통상적인 분산 시스템을 기준으로 성능 실험하였다. 이때, 서버와 클라이언트 노드에는 동일한 사양의 하드웨어를 적용하였다. PS-RPC의 성능 비교를 위한 RPC 미들웨어는 PS-RPC가 기반하고 있는 DDS의 플랫폼 투명성, 언어 투명성, 벤더 투명성, 표준 기구(Object Management Group)에서 정립한 산업 표준 등의 조건을 동일하게 갖추고 있는 CORBA로 선정하였다.

CORBA는 산업 표준인 동시에 동종 미들웨어 중에 성능이 가장 우수한 것으로 알려져 있다[10]. 성능 시험에 사용된 하드웨어 및 소프트웨어의 환경은 [표 4]와 같다.

표 4. PS-RPC 시험 환경
Table 4. PS-RPC Test Environment

Classification	Specification
CPU	Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz
RAM	DDR3 8GB
HDD	500GB
DDS	RTI NDDS 4.5d
CORBA	TAO 2.2a
OS	Windows 7 Professional SP1
Compiler	Visual C++ 2010 SP1

시험에 사용한 소프트웨어는 다음과 같이 제작되었다. 먼저 PS-RPC 모델러를 이용하여 입력 매개변수(요청 토픽)/출력 매개변수(응답 토픽)의 크기가 동일한 메아리 원격 프로시저 9개를 모델링하였으며, 원격 프로시저의 입력/출력 매개변수에 해당하는 토픽 크기는 각각 16 바이트, 32 바이트, 64 바이트, 128 바이트, 256 바이트, 512 바이트, 1024 바이트, 2048 바이트, 4096 바이트가 되도록 하였다. 이 원격 프로시

저 모델을 기반으로 IDL과 서버/클라이언트 스텝 코드를 생성하고 서버/클라이언트 응용 프로그램을 구현하였다.

제안하는 PS-RPC 미들웨어에 대한 성능 시험은 각 원격 프로시저의 호출 소요 시간을 50만회 측정하고 이에 대한 평균을 내는 방식으로 이루어졌다. 비교되는 두개의 환경에 대한 시험 결과는 [그림 8]과 같이 나타난다. 시험결과 제안하는 PS-RPC의 원격 프로시저 평균 호출 소요시간이 CORBA에 비해 평균4.87% 성능 향상을 보이는 것으로 확인되었다.

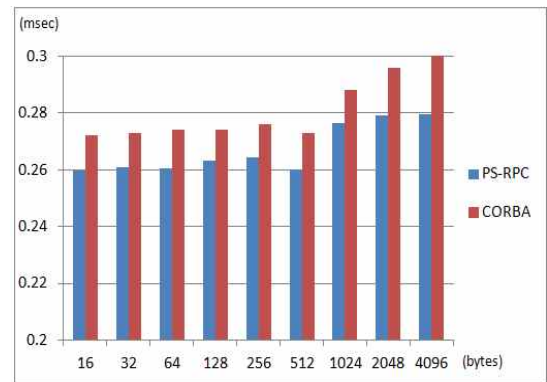


Fig. 8. Comparison Test Result of two models

그림 8. 비교 모델간 시험 결과

V 결론

제안하는 하이브리드형태의 PS-RPC 미들웨어는 PS 모델 기반에서 동작하는 RPC 미들웨어로써 낮은 결합도와 높은 응집도를 동시에 갖춘 분산 시스템 구축을 가능하게 한다. 또한 제안하는 하이브리드형태의 PS-RPC 미들웨어는 분산 시스템 개발자에게 높은 생산성을 제공한다.

제안하는 하이브리드형태의 PS-RPC 모델러는 분산 시스템을 시각적으로 모델링할 수 있도록 도우며, DDS 통신 인터페이스를 정의하는 IDL에서부터 서버와 클라이언트 응용을 구현하는데 필요한 스텝 코드를 생성한다. 분산 시스템의 응용 프로그램 개발자는 데이터의 송/수신에는 관여하지 않고, 오로지 시스템의 논리 구현에 집중하는 것이 가능하다.

향후에는 본 논문에서 제안하고 있는 PS-RPC 미들

웨어 모델의 활용과 관련하여 IoT(Internet of Things) 환경과 로보틱스 환경에서의 적용성에 대한 추가 연구가 필요하다.

OMG's Second Workshop on Real-time And Embedded Distributed Object Computing, June 2001.

References

- [1] Oki, B., Pfluegl, M., Siegel, A. Skeen, D, "The Information Bus, An Architecture for Extensible Distributed Systems", SOSPI4, pp 58-68, Dec, 1993.
- [2] Zhiwei Xu, Wei Li, Li Zha, Haiyan Yu, Donghua Liu, "A Computer Systems Approach to Grid Computing", Journal of Grid Computing Volume 2, Issue 2, pp 109-120, 2004
- [3] Hennadiy P, Middleware : Past and Present Comparison, UMBC, 2004
- [4] Bernstein P, "Middleware: A model for distributed systems services", Communications of the ACM, pp. 86 -98, Feb. 1996
- [5] M Hitz, B Montazeri, "Measuring coupling and cohesion in object-oriented systems", Proc. Third Int'l Symp. Applied Corporate Computing, Oct. 1995.
- [5] Patrick Th. Eugster , Pascal A. Felber , Rachid Guerraoui, Anne-Marie Kermarrec, "The many faces of publish/subscribe", ACM Computing Surveys (CSUR), v.35 n.2, pp.114-131, June 2003
- [6] Brian N. Bershad , Thomas E. Anderson , Edward D. Lazowska, Henry M. Levy, "Lightweight remote procedure call", ACM Transactions on Computer Systems (TOCS), v.8 n.1, pp.37-55, Feb, 1990
- [7] Gerado Pardo-Castellote, "Introduction to DDS and Data-Centric Communications" Proceeding of the 23rd International Conference on Distributed, Computing Systems Workshops, 2003
- [8] Gerado Pardo-Castellote, "OMG Data-Distribution Service: Architectural Overview", Proc. 23rd Int'l Conf. Distributed Computing Systems. ICDCSW, 2003
- [9] Jesús Martínez Cruz, "A DDS-based middleware for quality-of-service and high-performance networked robotics", Concurrency and Computation: Practice and Experience Volume 24, Issue 16, pp. 1940 -1952, November 2012
- [10] Robert Kelly Louis DiPalma. "Applying corba in a contemporary embedded military combat system".

BIOGRAPHY

Sang-hyun Park (Member)



2002 : BS degree in Industrial Engineering, Suwon Univ.
 2002~2003 : Engineer, Hanmi Data
 2003~2005 : Engineer, AIM Systems
 2005~2011 : Senior Engineer, Newgrid
 2011~2015 : Senior Engineer,

Samsung Thales Co., Ltd.

2015~ : Senior Engineer, Hanwha Thales Co., Ltd.

Current research interests :

Middle-ware, Distributed Computing System

June-sung Choi (Life Member)



1994~1998 : BS degree in Industrial Engineering, Republic of Korea Air Force Academy

2008~2010 : MS degree in Computer Science, Korea National Open Univ.

2011~2013: PhD degree in Industrial Information Systems Engineering,

Seoul National Univ. of Science and Technology

1998~2008 : Major, Ministry of National Defense, Official, Ministry of Safety and Administration

2008~2015 : Manager, Samsung Thales Co., Ltd., Hanwha Thales Co., Ltd.

Current research interests :

Warfare System Penetration Test Methodology

Warfare System Security Architecture

Cyber-Warfare, Secure Coding, International Law applicable to Cyber-War, Social Engineering Attack

Kwang-ho Kook (Member)

1979 : BS degree in Industrial
Engineering, Seoul National Univ.
1981 : MS degree in Industrial
Engineering, Seoul National Univ.
1989 : Ph.D degree in Industrial
Engineering, Georgia Tech
1981~1983 : Professor, Industrial

Engineering, Chungju Univ.

1989~1993 : Senior Researcher, Electronics and
Telecommunications Research Institute(ETRI)

1993~2010 : Professor, Industrial Information Systems
Engineering, Seoul National Industrial Univ.

2010~ : Professor, Industrial Information Systems
Engineering, Seoul National Univ. of Science and
Technology

Current research interests :

Information Security, Convergence Security,
Industrial Security