

소프트웨어 디자인 패턴을 적용한 실시간 분산 시뮬레이션을 위한 데이터 전달처리 시스템 설계

석진원*, 유인태**

요약

일반적으로 초고속 네트워크에서 실행되는 분산 시스템의 데이터 전달처리 효율은 시스템 구조 및 데이터의 전달처리 시스템에 의존한다. 분산 환경을 이용한 실시간 분산 시뮬레이션 시스템은 데이터 전달처리의 실시간성과 시스템의 신뢰성 보장을 위하여 데이터 전달처리 시스템에 의하여 요구된 성능을 만족하고자 하였다. 그러나 실시간 시뮬레이션 시스템에 적용된 클라이언트/서버 기반의 데이터 전달처리 시스템은 시스템의 안정성 및 시스템의 변경에 따른 확장성과 유지보수성 확보가 어려웠다. 따라서 기존의 데이터 전달처리 시스템의 문제점을 해결하기 위하여 새로운 데이터 전달처리 시스템이 필요하다. 본 논문에서는 기존의 실시간 시뮬레이션 시스템을 분석하여 시스템 개선방향을 제시하고, 시스템 확장성, 상호운용성, 재사용성 및 유지보수성을 위하여 소프트웨어 디자인 패턴을 적용한 새로운 실시간 데이터 전달처리 시스템을 제안한다.

Data Transmission Processing System Design for Real-Time Distributed Simulation by Using Software Design Patterns

Jinweon SUK*, Intae RYOO**

Abstract

Usually, The data transmission processing efficiency of the distributed system running on high speed networks depends on the system architecture and the data transmission processing system. In order to secure the real-time rate and the system reliability, the real-time distributed simulation system on the distributed environment has tried to satisfy the performance required by the data transmission processing system. However, the client/server-based data transmission processing system in the real-time simulation system has been difficult to satisfy the system stability, extensibility and maintainability, especially when system changes. So, it is natural to study another improved data transmission processing system to solve the problems at the existing real-time simulation system. After analyzing the existing real-time simulation system, this paper will propose the improved real-time data transmission system by using Software Design Pattern, which enhances extensibility, interoperability, reusability and maintainability of the system.

Keywords : Real-Time Simulation, Distributed System, Data Transmission Processing

1. 서론

초기에 고정된 공간에서 제한적으로 사용되던

컴퓨터 시스템은 네트워크 기술의 급속한 발전으로 지리적, 공간적 한계를 극복한 분산 시스템으로 발전하였으며, 이러한 분산 시스템 기술과 환경은 시뮬레이션 분야에 적용되어 컴퓨터 시뮬레이션 기술의 획기적인 발전으로 이어져 오늘날 초고속 네트워크 기반의 실시간 분산 시뮬레이션 시스템으로 확장되었다.

실시간 분산 시뮬레이션 시스템은 대규모 군사용 시뮬레이션 시스템이나 복잡한 무기체계의 개발 및 운용, 최근의 우주개발사업, 그리고 공

※ 공동저자(co-author) : 석진원, 유인태

접수일:2009년 12월 09일, 완료일:2009년 12월 30일

* 경희대학교 컴퓨터공학과 (sjw0176@unitel.co.kr)

** 경희대학교 전자정보대학 (itryoo@khu.ac.kr)

▣ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2009-(C1090-0903-0011))

항, 철도 및 항만 등의 초대형 교통 시스템과 국토개발사업, 원자력 발전과 같은 중대한 위험 시스템 및 국가재난관리 시스템 등에 활용되고 있다[9][10].

실시간 시뮬레이션 시스템은 그 목적상 초고속 네트워크를 기반으로 실시간성과 고신뢰성 그리고 대량의 데이터 전달처리 능력과 함께 빠르게 변화하는 정보기술과 시뮬레이션 기법을 위하여 시스템의 확장성, 상호운용성, 사용자 및 운영자에게 편리한 사용성과 높은 유지보수성을 제공해야 한다. 실시간 시뮬레이션 시스템에서 가장 중요한 부분은 시뮬레이션 적용분야의 전문지식과 최신의 시뮬레이션 기법을 반영한 실시간 시뮬레이터와 초고속 네트워크에서 갑자기 폭주하는 데이터를 시스템의 각 구성요소에 안정적으로 전달하여 실시간성을 보장해주는 데이터 전달처리 시스템이다[12].

본 논문에서는 기존의 대규모 군사용 실시간 분산 시뮬레이션 시스템(또는 위게임 시뮬레이션 시스템, 이후로 “실시간 시뮬레이션 시스템”이라 함)에서 사용되는 데이터 전달처리 시스템의 한계와 문제점을 알아보고, 소프트웨어 디자인 패턴을 적용하여 실시간 시뮬레이션 시스템의 목적을 충족시킬 수 있는 데이터 전달처리 시스템을 제안한다. 제약사항으로 연구분야의 관련 자료 및 디자인 패턴 적용사례가 많지 않아 기존 시스템을 기준으로 문제점 도출 및 개선방안을 제시하고 연구결과를 비교해 본다.

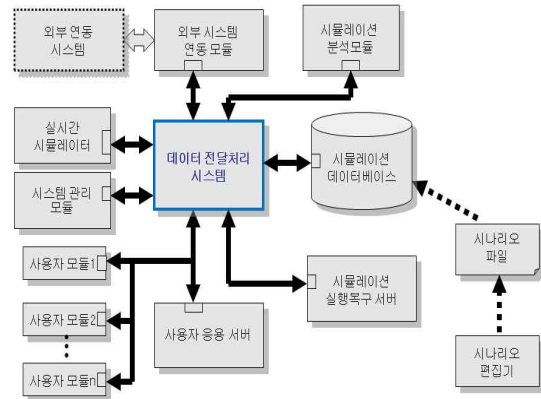
본 논문은 2장에서는 기존 시스템에 대하여 알아보고, 3장에서는 기존 시스템의 한계와 문제점에 대하여 소프트웨어 디자인 패턴의 적용방안을 찾아본 뒤, 4장에서는 디자인 패턴을 적용한 새로운 구조의 데이터 전달처리 시스템을 제안하고 기존 시스템과 비교한다. 5장에서는 연구결과를 뒤돌아보고 향후 연구방향을 제시한다.

2. 기존 데이터 전달처리 시스템의 문제점 및 개선방향

2.1 실시간 시뮬레이션 시스템 구성

기존의 실시간 시뮬레이션 시스템은 (그림 1)과 같이 전통적인 클라이언트/서버 구조의 형태를 취하고 있다. 기존의 실시간 시뮬레이션

시스템은 실시간 시뮬레이터, 시스템 관리 모듈, 시뮬레이션 데이터베이스, 시뮬레이션 실행복구 서버, 사용자 응용 서버, 사용자 모듈, 시나리오 편집기, 외부 시스템 연동모듈 및 시뮬레이션 분석 모듈 등으로 구성되며, 시뮬레이션 시스템 구성요소들을 하나의 통합된 구조로 만들기 위하여 데이터 전달처리 시스템이 필요하다[5][7][8][14].

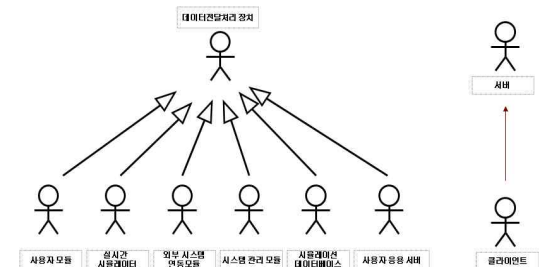


(그림 1) 기존의 실시간 시뮬레이션 시스템

2.2 기존 데이터 전달처리 시스템 문제점

기존 데이터 전달처리 시스템은 (그림 3)의 '1'과 같은 전통적인 2계층 클라이언트/서버 구조이며, 운영의 핵심이 되는 실시간 시뮬레이터를 포함한 대부분의 구성요소들이 클라이언트 영역에 포함되고, 이들 간의 연결을 관리하고 데이터의 전달처리를 보장하는 데이터 전달처리 시스템이 서버 영역에 위치하여 동작하는 구조이다.

기존 데이터 전달처리 시스템은 (그림 2)와 같이 통신 구조상 데이터 전달처리 시스템을 서버로 구성하고, 실시간 시뮬레이터 및 시뮬레이션에 참여하는 구성요소들을 클라이언트로 구성하였다.

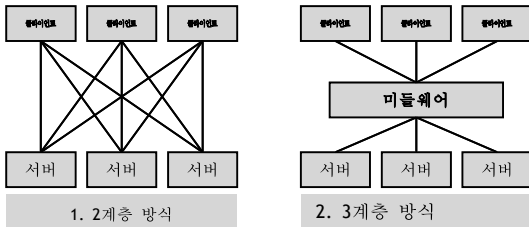


(그림 2) 기존의 데이터 전달처리 시스템 구조

이러한 기존 데이터 전달처리 시스템의 구조적인 문제는 데이터 전달처리 시스템의 통신장애와 같은 응급상황 발생 시 각 구성요소에 전달되어야 할 데이터의 신뢰성 및 처리율이 현저히 낮아지는 현상이 일어나게 되고, 최악의 경우에는 운영의 핵심인 실시간 시뮬레이터가 재시작 하게 되는 논리적 모순을 포함하게 된다는 것이다. 이 때 시스템의 재시작 순서는 먼저 서버 형태의 구성요소(실시간 시뮬레이터, 사용자 응용 서버, 시뮬레이션 데이터베이스 등)부터 재접속시키며, 그리고 기타 구성요소(사용자 모듈 등) 들을 연결시킨다. 이는 시스템 운영상 많은 문제를 내재하게 된다. 또한 구현상의 문제로는 시스템 콜 성능이 현저히 저하되는 다중 Wrapper 형태의 ACE_Wrapper의 사용 등도 원인이 된다[1][3][13][15].

2.3 데이터 전달처리 시스템 개선방향

(그림 3)은 분산시스템의 일반적인 통신 구조를 나타내었다. (그림 3)의 ‘1’에는 2계층(2-Tier) 방식으로 기존 데이터 전달처리 시스템에서 채용한 일반적인 시스템 구조이며, ‘2’의 3계층(3-Tier) 방식은 퍼사드 형태로 일반적인 미들웨어 방식으로 제안된 시스템에 적용하였다[9].



(그림 3) 클라이언트/서버 구조

본 논문에서 제안하는 데이터 전달처리 시스템은 기능상 서버(Server), 브릿지(Bridge), 클라이언트(Client)로 구성되어지며, 논리적으로 서버와 브릿지의 서버 군(Group)과 브릿지와 클라이언트의 클라이언트 군으로 분류가 가능하도록 한다. (그림 4)는 제안하는 데이터 전달처리 시

스템의 기본 구조를 유스케이스(Use Case) 다이어그램을 나타낸 것으로 3계층 방식을 기반으로 한다[13].



(그림 4) 3계층 클라이언트/서버 구조

제안하는 실시간 시뮬레이션 시스템을 위한 데이터 전달처리 시스템은 3계층의 클라이언트/서버 구조를 사용하며, 데이터 전달처리 시스템의 구성요소들을 서버 계층(실시간 시뮬레이터, 시뮬레이션 데이터베이스 등), 인터페이스 계층(데이터 전달처리 시스템), 클라이언트 계층(사용자 모듈과 같은 다른 구성요소)으로 구성하였으며, 개선된 데이터 전달처리 시스템 설계목표는 실시간 분산 시뮬레이션 시스템의 개발 및 유지보수에서 소프트웨어 디자인 패턴을 적용함으로써 소프트웨어 재사용성, 유지보수성, 확장성 및 상호운용성을 향상시키고, 시스템 안정성 및 신뢰성을 확보할 수 있도록 하며, 네트워크 처리효율을 향상시키고 시스템 및 구성요소 간의 독립성을 확보할 수 있게 한다.

3. 소프트웨어 디자인 패턴의 적용

3.1 디자인 패턴

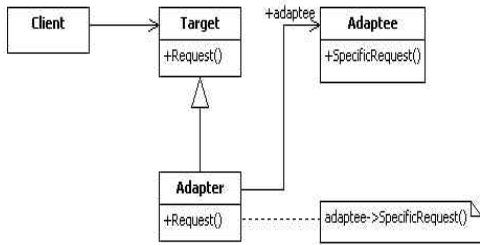
소프트웨어 디자인 패턴 개념은 원래 건축분야에서 1970년대 말에 크리스토퍼 알렉산더(Christopher Alexander)가 정립하여 발표한 개념으로써 소프트웨어 분야에서는 워드 커닝햄(Ward Cunningham)과 켄트 벡(Kent Beck)이 1980년대 말에 스몰토크(Smalltalk) 언어를 사용한 프로젝트에서 5가지의 패턴 언어를 개발하여 사용한 것이 시발점이다. 그리고 1990년대 초에는 에릭 감마(Erich Gamma), 리처드 헬름(Richard Helm), 존 블리지드(John Vlissides), 랄프 존슨(Ralph Johnson)에 의하여 디자인 패턴이 체계적으로 정리되어 책으로 발표되었는데 이들 팀의 이름을 따서 ‘GoF(Gang of Pattern) 패턴’이라고 불리워지며 23가지의 패턴으로 정리되어 있다. 그리고 GoF 디자인 패턴은 다시 생

성패턴, 구조패턴, 행위패턴으로 분류되어진다.

본 논문에서는 GoF 디자인 패턴 중 데이터 전달처리 시스템의 구조적 특성을 파악하여 구조패턴과 행위패턴 중에서 3가지 패턴을 추출하여 적용하였다. 이들은 어댑터(Adapter) 패턴, 브릿지(Bridge) 패턴, 퍼사드(Facade) 패턴으로써 다음으로 적용된 디자인 패턴에 대한 기본 개념, 패턴 선정 이유 및 적용 과정, 그리고 데이터 전달처리 시스템에 디자인 패턴을 적용한 결과와 특성을 설명하였다[3][5][8].

3.2 어댑터(Adapter) 패턴의 적용

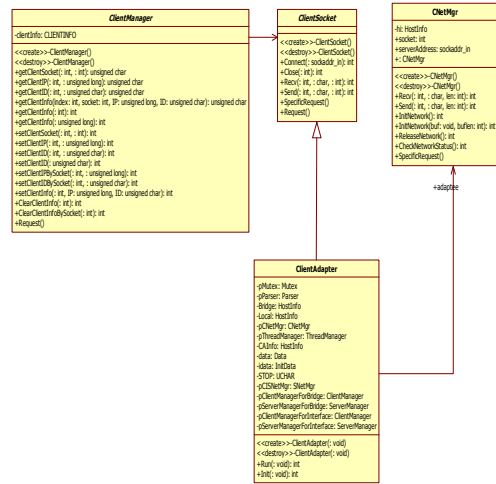
먼저 어댑터 패턴은 전원 소켓의 어댑터와 같이 ‘소프트웨어 구현에서 하나의 클래스 인터페이스를 클라이언트에서 사용하고자 하는 다른 인터페이스로 변환하는 패턴’으로써 어댑터 패턴을 이용하면 인터페이스 호환성 문제로 인하여 같이 사용할 수 없는 클래스들을 사용할 수 있게 해준다. (그림 5)는 어댑터 패턴의 기본적인 클래스 다이어그램이다[11].



(그림 5) 어댑터 패턴의 클래스 다이어그램

어댑터 패턴은 데이터 전달처리 시스템에서 서버 어댑터(Server Adapter)와 클라이언트 어댑터(Client Adapter) 부분에 적용하였으며 (그림 6)은 어댑터 패턴의 적용결과이다.

이들 서버 어댑터와 클라이언트 어댑터의 주된 설계 관점은 각 체계와의 유연한 연결 및 통신능력 보장에 있다. 여기서 유연한 연결이라는 것은 시스템의 각 구성요소와 연결할 때 별도의 부가적인 구성없이 연결 및 해제가 용이함을 의미하여, 어댑터는 실시간 시뮬레이션이 수행 중에 발생하는 네트워크 부하를 분산시킬 수 있도록 하였다.



(그림 6) 어댑터 패턴의 적용 결과

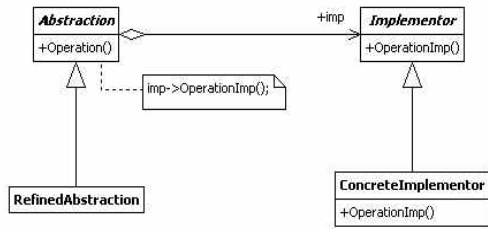
다음으로 어댑터 패턴에 대응되는 클라이언트 어댑터의 구현 클래스로 <표 1>과 같으며, 이들은 각 구현 클래스 중 클라이언트 어댑터 클래스의 경우로 시스템의 각 구성요소의 연결 상태를 관리하고, 각 구성요소에서 전송하는 데이터를 내부적으로 분석하여 최종 목적지까지 안전하고 정확하게 전송할 수 있다.

<표 1> 어댑터 패턴의 대응 클래스

어댑터 패턴	구현 클래스
Client	ClientManager
Target	ClientSocket
Adapter	ClientAdapter
Adaptee	CNetMgr

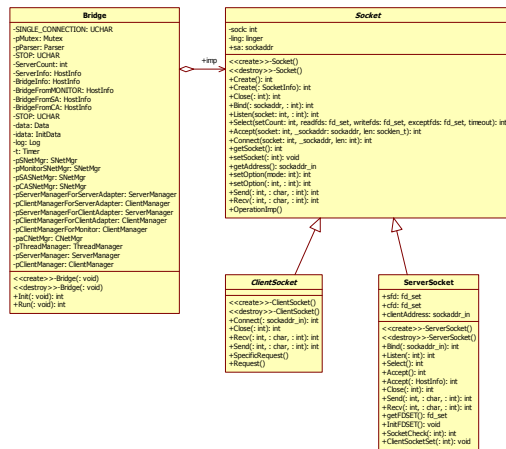
3.3 브릿지(Bridge) 패턴의 적용

브릿지 패턴은 ‘구현뿐만 아니라 추상화된 부분까지 변경시켜야 하는 경우에 사용하는 패턴’으로써 이 패턴을 이용하면 추상화된 부분과 추상 클래스 및 인터페이스를 구현하는 클래스를 서로 다른 클래스 계층구조에 집어넣어서 그 둘을 모두 변경시키는 역할을 한다. (그림 7)은 브릿지 패턴의 기본적인 클래스 다이어그램이다 [11].



(그림 7) 브릿지 패턴의 클래스 다이어그램

브릿지 패턴은 데이터 전달처리 시스템의 인터페이스 계층에 적용되었으며 (그림 8)은 적용된 브릿지 패턴의 클래스 다이어그램이다. 브릿지 패턴의 설계관점은 시스템의 각 구성요소들이 독립적으로 운용할 수 있도록 하며, 대규모 데이터 처리 요청에 대하여 필요시 병렬 시스템으로 구성이 가능하도록 설계가 되었다는 점이다. 적용된 브릿지 패턴의 경우 각 어댑터와 유기적으로 연동이 가능하여 위치독립성 및 유지보수성을 향상시키며 장애발생 시 안정적으로 대응할 수 있다.



(그림 8) 브릿지 패턴의 적용 결과

데이터 전달처리 시스템에서 인터페이스 계층은 클라이언트 어댑터에 대한 서버로 동작되며, 이 경우 클라이언트 군으로 구분이 가능하다. 또한 서버 어댑터와는 소켓 이중화로 구성되어 있으며, 이 경우 서버 군으로 구분이 가능하다. 이러한 시스템 구성은 각 구성요소로 하여금 전체 시스템에 대한 위치 투명성을 제공하며, 각 구성요소에서는 시스템 구성 변경 및 구조의 개선과

는 독립적으로 구성되어 운영될 수 있어서 시스템의 확장성 및 상호운용성을 향상 시키게 된다.

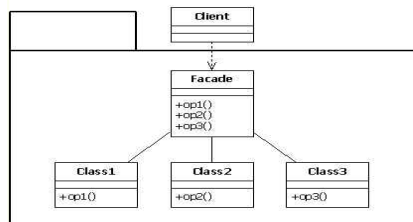
다음은 브릿지 패턴에 대응되는 브릿지의 구현 클래스를 <표 2>에 나타내었다. 각 구현 클래스 중 ClientSocket, ServerSocket 클래스의 경우 데이터의 분류 및 전송을 담당한다. 또한 브릿지 클래스의 경우 서버 어댑터와 이중화 연결을 지원하며, 클라이언트 어댑터의 복수 연결 및 브릿지 간의 복수 연결 기능을 제공하여 병렬 분산 구조로 구성이 가능하게 하였다.

<표 2> 브릿지 패턴의 대응 클래스

어댑터 패턴	구현 클래스
Abstraction	Bridge
Implementor	Socket
ConcreteImplementor	ClientSocket ServerSocket

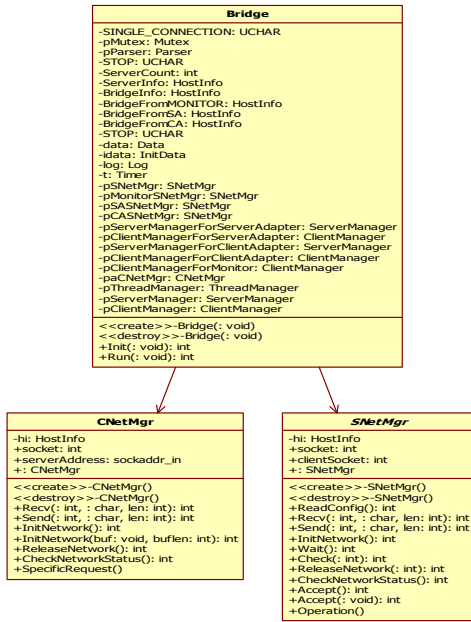
3.4 피사드(Facade) 패턴의 적용

피사드 패턴은 '서브 시스템을 통합하는 다수 객체들의 인터페이스 집합에 대해서 일관된 하나의 인터페이스를 제공할 수 있도록 하기 위한 패턴'으로 복잡한 서브 시스템에 대한 단순한 인터페이스 제공이 필요할 때 유용한 패턴이다. 피사드 패턴의 기본적인 클래스 다이어그램은 (그림 9)에 나타내었다[11].



(그림 9) 피사드 패턴의 클래스 다이어그램

데이터 전달처리 시스템에서 각 어댑터와 서버 및 클라이언트 간, 어댑터와 브릿지 간의 유연한 연결을 지원하기 위하여 (그림 10)과 같이 적용한다. 데이터 전달처리 시스템에서는 물리적인 형태의 피사드 패턴을 응용하여 각 구성요소 간에 독립적인 운영 및 설계가 가능하고 더불어 분산 시스템 환경에서 확장 및 시스템 재구성이 용이하도록 하였다.



(그림 10) 퍼사드 패턴의 적용 결과

다음은 퍼사드 패턴에 대응되는 브릿지의 구현 클래스를 <표 3>에 나타내었다. 각 구현 클래스 중 SNetMgr는 서버 어댑터와 CNetMgr는 서버 어댑터, 클라이언트 어댑터와 연동하는 구조로 구성되어 있다. Bridge 클래스의 경우 시스템 전반에 걸쳐 데이터 전달처리 시스템의 핵심 기능을 제공하도록 설계되어 있다. Bridge 클래스를 퍼사드 패턴을 이용하여 적용하게 되면 향후 시스템의 확장시 매우 용이하며 각 구성요소 간의 모듈 독립성이 증가하게 된다.

<표 3> 퍼사드 패턴의 대응 클래스

퍼사드 패턴	구현 클래스
Facade	Bridge
Class	CNetMgr, SNetMgr

4. 실시간 데이터 전달처리 시스템

디자인 패턴을 적용하여 실시간 분산 시뮬레이션 시스템에서 데이터 전달처리 시스템 설계 과정은 앞에서 제시한 설계 목표를 달성하기 위하여 1. 데이터 전달처리 시스템의 기능요소별

요구사항 파악, 2. 설계 방안(특성) 도출, 3. 관련 디자인 패턴 적용, 4. 기능요소별 적용결과 통합, 5. 설계 결과의 검증 및 보완하는 과정으로 진행하였다.

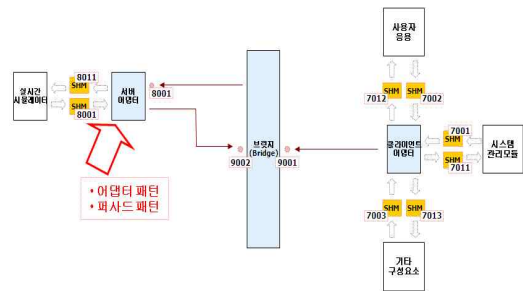
본 연구에서는 서버, 인터페이스, 클라이언트로 구성요소를 구분하여 진행하였으며, 다음으로 각 구성요소에서 디자인 패턴의 적용과정과 시스템의 논리적인 연결과정, 그리고 기존 시스템과 제안한 시스템을 비교하여 결과를 분석한다.

4.1 서버 계층

데이터 전달처리 시스템에서 서버와 클라이언트를 엄격하게 분리하기 위하여 서버와 데이터 전달처리 시스템을 어댑터 패턴과 퍼사드 패턴을 적용하여 실시간 시뮬레이터 등의 다른 구성요소와 독립적인 구성이 가능하게 하였다.

여기서 서버 어댑터는 실시간 시뮬레이터와 결합되어 서버 기능을 담당하게 되고, 서버 어댑터의 경우 브릿지와 네트워크로 연결되어 논리적이면서 물리적으로도 분산시킬 수 있는 구조로 되어 있다. 물리적으로 분산된 하나의 시스템 구조로 구성이 가능한 서버 어댑터는 그 구성상 어댑터 패턴과 퍼사드 패턴을 합성하여 적용하였다. 실시간 분산 시뮬레이션 환경에서 서버들은 능동적으로 서버 어댑터에 연결하여 클라이언트들에게 데이터를 전송하게 된다.

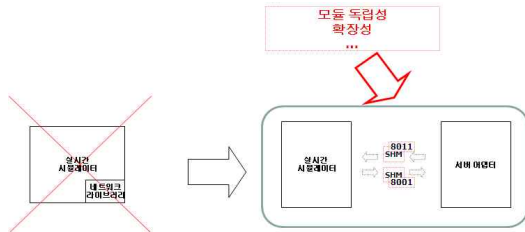
다음 (그림 11)은 디자인 패턴 중 어댑터 패턴과 퍼사드 패턴을 적용한 서버 계층을 나타낸다. 서버 계층과 클라이언트 계층에 적용된 어댑터 패턴은 퍼사드 패턴과의 합성에서 어댑터 패턴에 퍼사드 패턴의 특징을 내포하고 있어 별도의 클래스 다이어그램을 나타내지 않았다.



(그림 11) 서버 계층의 디자인 패턴 적용

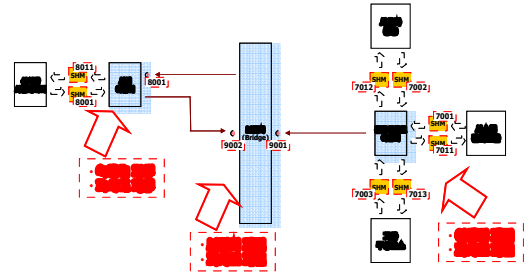
4.5 기존 시스템과 제안 시스템 비교

제안된 데이터 전달처리 시스템의 비교기준으로 국제표준으로 인정된 ISO/IEC 9126의 소프트웨어 품질특성을 적용하여 소프트웨어의 기능성(상호운용성), 신뢰성(회복성), 사용성(운용성), 효율성(시간반응성, 자원효율성), 유지보수성(변경성, 안정성), 이식성(적응성, 설치성) 측면에서 기존 시스템과 제안된 시스템을 비교해 보았다 [4][6]. 실시간 분산 시뮬레이션 시스템의 데이터 전달처리 시스템 관련한 기존 연구결과가 부족하여 부득이하게 기존 시스템과 비교하여 결과를 제시한다. 비교과정은 기존 시스템의 문제점, 제안 시스템의 설계목표와 제안된 시스템의 설계결과에 대하여 관련항목을 비교해 보았다. 비교 과정을 살펴보면, 기존 시스템은 네트워크 모듈을 라이브러리 형태로 각 구성요소에 구축함으로써 시스템 변경 시 각 구성요소별로 시스템을 변경해야 하고 연관된 기능들을 모두 변경해야 하는 번거로움이 있었다. 이는 시스템의 개발 및 운영에서 모듈의 독립성이나 상호운용성, 확장성, 재사용성 등을 크게 저하시킨다. 그러나 제안된 시스템의 각 구성요소들은 다른 구성요소의 위치와 무관하게 어댑터에 연결하는 것으로 동일한 서비스가 가능하다. (그림 16)에 라이브러리 형태의 데이터 전달처리 모듈을 어댑터로 분리하는 것을 나타내었다.



(그림 16) 어댑터에 의한 네트워크 모듈 분리로 사용성 및 이식성 향상

또한 다음 (그림 17)은 디자인 패턴이 적용된 데이터 전달처리 시스템으로 여기서 서버계층은 클라이언트 구성요소 입장에서 하나의 서버로 동작하는 구조로 구성되며, 이것은 서버 구성요소에서 위치독립성 향상으로 시스템 개발 및 운영에서 기능성과 유지보수성 등을 향상시킬 수 있다.



(그림 17) 디자인 패턴 적용에 의한 시스템의 기능성 및 유지보수성 향상

그러므로 제안된 데이터 전달처리 시스템은 기존 시스템과 비교하여 <표 4>와 같이 대부분의 비교항목에서 개선되어진 것을 볼 수 있다.

<표 4> 기존 시스템과 제안 시스템 비교

비교 항목		기존 시스템	제안된 시스템
기능성	상호운용성	×	○
신뢰성	회복성	×	△
사용성	운용성	△	○
효율성	자원효율성	×	△
	시간반응성	△	○
유지보수성	변경성	×	○
	안정성	×	△
이식성	적응성	×	○
	설치성	△	△

5. 결론

지금까지 분산 환경을 기반으로 하는 기존의 실시간 시뮬레이션 시스템에서 데이터 전달처리 시스템의 문제점에 대하여 알아보고, 이에 대한 개선방안으로 소프트웨어 디자인 패턴 기법을 도입한 새로운 구조의 데이터 전달처리 시스템을 제안하였다.

본 논문에서 제안한 디자인 패턴을 적용한 데이터 전달처리 시스템은 고유의 실시간성 보장, 장치의 독립성 및 상호운용성의 증가로 실시간 시뮬레이션 시스템 개발 및 유지보수 시 각 구성요소에서 전체 시스템의 구성 및 환경에 대한 고려사항을 현저하게 감소시킬 수 있어 소프트

웨어의 재사용성, 유지보수성 및 확장성의 향상을 가져다 줄 수 있다.

향후에는 실시간 시뮬레이션 시스템의 데이터 전달처리 시스템 성능과 신뢰성 향상을 위한 실시간 스케줄러 및 구성요소 접속방법 등에 대한 추가적인 연구와 시스템 구현 및 소프트웨어 품질특성을 반영한 시스템 성능평가가 필요하다고 본다.

참 고 문 헌

[1] 구성환, "무선PBX 시스템을 위한 객체기반 통신망 관리시스템의 설계 및 구현", 서강대학교 석사학위논문, 2004.12

[2] 김용호, "HLA 기반 적응형 페더레이트 설계 및 구현", 충남대학교 박사학위논문, 2006.2

[3] 에릭 프리먼 외 3인 공저, 서환수 역, "Head First Design Patterns", 한빛미디어, 2005

[4] 양해술, 이하용, 박주석, "미들웨어 소프트웨어의 시험사례 연구", 한국콘텐츠학회 논문지 9권 7호, pp110-120, 2009.7

[5] 이강택, "침입감내기술 기반의 보안시스템 설계 및 구현 : 위게임 시스템 중심으로", 경기대학교 박사학위논문, 2006.8

[6] 이길섭, 이태공, "RUP 기반 CIS 소프트웨어 개발프로세스", 한국인터넷정보학회 논문지 5권 4호, pp95-113, 2004. 8

[7] 이창섭, "DEVS 기반의 실시간 위게임 시뮬레이션을 위한 저장 및 복구방법론", 한국과학기술원 석사학위논문, 2006.2

[8] 임치훈, "위게임 시스템을 위한 디자인 패턴의 합성 및 적용", 한밭대학교 정보통신대학원 석사학위논문 2008.2

[9] Amjad Umar, "Distributed Computing and Client Server Systems", Prentice Hall, Inc, 2005

[10] Bernard P. Zeigler, Herbert Praehoffer, Tag Gon Kim, "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems", Academic Press, 2000

[11] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns : Elements of Reuseable Object-Oriented Software", Person Education, Inc, 1995

[12] Frederick Kuhl, Richard Weatherly, Judith Dahman n, " Creating Computer Simulation Systems", Prentice Hall, Inc, 1999

[13] <http://www.cs.wustl.edu/~schmidt/ACE-papers.html>

[14] James F Dunnigan, "Wargames Handbook, Third Edition : How to Play and Design Commercial and Professional Wargames", Writers Club Press, 2000

[15] Stephen D. Huston, James CE Johnson, Umar Syyd, "ACE Programmer Guide : The Practical Design Patterns For Network And System Programming", Pearson Education, Inc. 2004



석진원

1990년 금오공과대학교 전자공학과 (컴퓨터공학) 졸업(공학사)
 1996년 국방대학원 전산학과 졸업 (국방과학석사)
 2004년 경희대학교 대학원 컴퓨터 공학과 수료(공학박사)

1990년~2003년 : 국방부(공군) 국방정보체계담당
 2003년 : 한국국방연구원 선임연구원
 2003년~2005년 : 포스테이타(주) 차장
 2005년~2009년 : SKC&C(주) 부장
 2009년~현 재 : (주)비전엔바이오테크 이사
 관심분야 : 시뮬레이션, RFID/USN, 그리드 네트워크



유인태

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~현 재 : 경희대학교 전자정보대학 교수
 관심분야 : 인터넷 기술, IPTV, 멀티미디어 통신