

상호운용성 보장을 위한 CORBA 기반의 전환 절차 개발 연구

강우석, 장창복, 조성훈, 김동혁, 이찬섭, 조완수*, 최의인
한남대학교 컴퓨터공학과
*국방과학연구소
e-mail:wskang@dblab.hannam.ac.kr

Development of System Migration Process for interoperability suaranty based on CORBA

Woo-Suck Kang, Chang-Bok Jang, Sung-Hoon Cho,
Dong-Hyuk Kim, Chan-Seob Lee, Wan-Soo Cho*, Eui-In Choi
Dept of Computer Engineering, Han-nam University
*Agency for Defense Development

요약

이질 분산 환경에서 분산 객체 미들웨어를 이용한 시스템 통합 기술은 시스템 서로 간에 상호운용 가능하도록 할뿐만 아니라 개발비용의 절감을 가져 올 수 있다. 현재의 군 정보체계는 하부 구조와 개발 플랫폼이 정의없이 정보 시스템 개발을 진행함에 따라 재사용성과 기술 공유가 미비한 실정이다. 따라서 이를 해결하기 위해서는 분산 객체 기술을 이용하여 개방 체계로 전환하기 위한 접근 방법이 필요하다. 따라서, 본 논문에서는 군 정보 시스템을 분산객체 미들웨어인 CORBA를 이용하여 통합할 수 있는 절차를 확보한 뒤 이를 설계하고 테스트함으로써 군 정보 시스템 자원과 C4I 시스템 개발 자원의 낭비를 절감시키고, 현재 시스템 간의 상호 운용성이 가능하도록 하였다.

1. 서론

네트워크의 발전과 컴퓨터 성능의 발달로 분산되어 있는 시스템간의 정보 공유 및 상호 운용이 필요하게 되었다. 하지만 서로 다른 환경으로 인해 시스템간에 상호 운용이 불가능함에 따라 이를 해결하기 위한 분산 객체 미들웨어인 CORBA가 기안되었다 [8]. 현재 군 정보 시스템은 공통적인 정보체계 기반 구조 정립 및 개발 플랫폼의 정의 없이 정보체계 개발 사업이 진행됨에 따라 정보체계 개발 사업 간의 기술 공유 및 체계 재사용이 매우 미흡하다. 따라서 기존의 스토브파이프(stovepipe) 방식의 정보체계로

는 분산 객체 기술을 이용한 개방 체계로 전환하고자하는 공통적 접근 방법이 개발되어 있지 못하다. 이를 해결하기 위해 분산 객체 미들웨어 환경에서 효율적인 정보 교환을 지원할 수 있는 하부구조인 CORBA를 이용한 시스템 통합 기술과 분산 객체 미들웨어에 의한 체계 전환 기술 및 절차 확보에 관한 연구가 필요하다[7].

본 논문에서는 이질적인 정보체계 및 C4I 체계를 상호연동하고, 국방 자원 투자 보호 및 분산 컴퓨팅 환경에서의 상호 운용성을 보장하기 위해 분산객체 미들웨어인 CORBA를 이용하여 클라이언트 계층과 이 기종 시스템 계층 간의 인터페이스를 설계하고 테스트하였다.

이 논문은 2001년도 국방과학연구소의 연구비 지원에 의하여 수행되었음.

2. 관련 연구

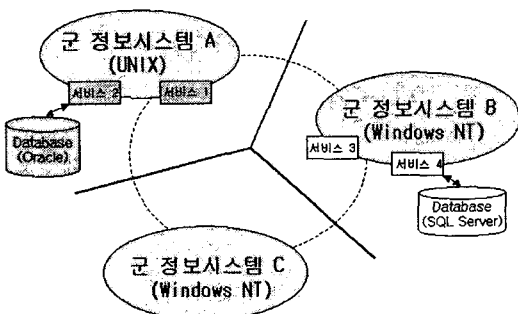
이질 분산 환경의 시스템 통합 기술로는 OMG의 CORBA, Microsoft의 DCOM, SUN의 RMI 등이 있다. 그중에서 분산객체 미들웨어로써 많이 사용되고 있는 CORBA는 OMG에서 제안한 객체지향 인터페이스 표준으로써 IDL(Interface Definition Language), IR(Interface Repository), Implementation Repository, DII(Dynamic Invocation Interface), Object adapter, ORB등으로 스펙이 구성되어져 있다[6].

마이크로소프트사의 DCOM은 인터넷이나, LAN, WAN 상의 여러 컴퓨터에 존재하는 객체들 간의 통신을 지원하기 위해서 COM(Component Object Model)을 확장한 것이다[4].

CORBA를 이용한 체계 전환 연구로는 MITRE의 DOMIS 프로젝트가 있다. DOMIS(Distributed Object Management Integration System)는 MITRE사에서 미 공군의 CTAPS(Contingency Theater Automated Planning System)의 문제점들을 해결하기 위해 분산 객체기술을 사용하여 하부시스템을 통합하고 DOM(Distributed Object Management) 구조로 변환하는 기술과 방법론을 개발한 프로젝트이다 [1, 2, 5, 7].

3. 상호운용을 위한 시스템 구조

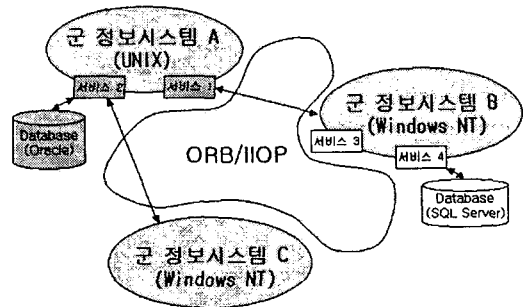
본 논문에서 상호운용을 위한 군 정보 시스템은 [그림 1]처럼 이질적인 운영체제와 데이터베이스로 구성되어있다. 따라서 이들 시스템들은 상호간 정보 교환 및 공유, 활용이 미약하고 상호 운용이 불가능하기 때문에 시스템에서 제공하는 유사한 기능들이 중복되어 개발된다는 문제점이 존재한다.



[그림 1] 군 정보 시스템 구조

이러한 문제점을 해결하기 위해서 군 정보 시스템

을 CORBA의 ORB와 IIOP, IDL등을 이용하여 [그림 2]처럼 상호 운용 가능한 구조로 전환하였다.



[그림 2] 상호 운용을 위한 시스템 구조

[그림 2]에서 서비스 1, 2, 3, 4는 각각 군 정보 시스템 A와 B에서 제공하고 있는 서비스를 나타낸다. 시스템 체계 전환 절차를 확보하기 위해 새로운 군 정보 시스템 C가 도입되었을 경우에는 기존 시스템이 제공하고 있는 서비스인 서비스 1과 서비스 2를 위해 어떤 절차나 동작 없이 해당 서비스를 호출하여 군 정보시스템 A와 B를 사용하기 위한 구조를 보여주고 있다.

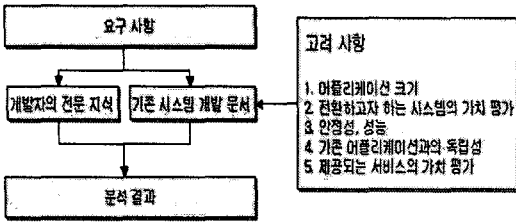
4. 상호 운용 절차

기존 군 정보시스템을 상호 운용 가능하도록 하기 위해서는 본 논문에서 제안한 절차는 다음과 같다.

1) 요구 사항 분석 및 CORBA 지원 제품 사용법 습득

기존 시스템을 통합하기 위해 체계 전환을 위해 필요한 요구 사항 및 통합 환경은 어떻게 되고 재사용할 서비스가 무엇인지를 분석한다. 이러한 분석을 통하여 체계 전환시 사용할 CORBA 지원 어플리케이션을 선택한다. 이러한 어플리케이션으로는 Orbix, Visibroker 같은 많은 제품이 존재하며 그 것들 중에서 개발자의 능력과 개발환경에 따라 제품을 선택한다.

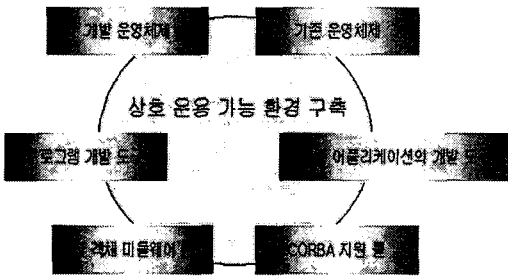
CORBA 호환 어플리케이션을 선택한 후에는 어플리케이션의 사용법을 습득한다. 기존 프로그램 개발자인 C++이나 JAVA를 알고 있더라도 CORBA 호환 제품을 이용하기 위해서는 제품의 사용 방법을 습득해야 하며 IDL를 통한 프로그래밍 방법을 알아야 한다.



[그림 3] 요구사항 분석

2) 개발 환경 구축

분산 이질 시스템에서 상호 운용이 가능하도록 정보 체계 전환 시스템을 구축하기 위해서는 개발자들의 능력과 개발환경에 맞는 시스템과 프로그래밍 언어를 선택한다. [그림 4]는 개발 환경을 구축하기 위해 고려되어야 할 사항들이다.



[그림 4] 개발 환경 구축

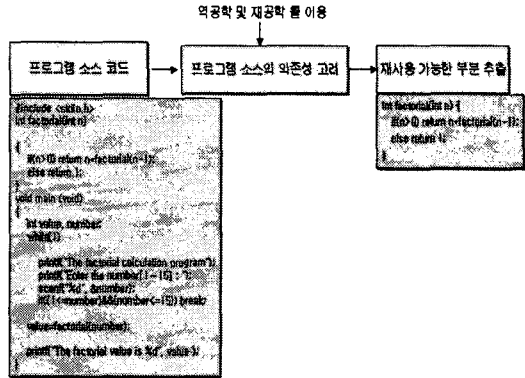
3) 소스 코드 분석

기존 애플리케이션이나 시스템으로부터 재사용 가능한 서비스를 추출하기 위해 소스 코드를 분석하고, 실제로 서비스를 제공하는 부분이 어느 위치에 있는지를 파악한다. 기존 소스 코드의 분석은 다소 개발자 능력에 따라 차이가 있을 수 있으며, 역공학 툴이나 재공학 툴을 사용하여 분석 작업을 실행할 때에는 이에 상응하는 지식이 요구된다. 또한 일반적으로 대부분의 소스는 각 모듈간의 관계가 서로 독립적이지 못하고 종속적인 관계로 되어 있는 경우가 많기 때문에 이러한 모듈간의 관계를 쉽게 파악할 수가 없다. 그러므로 모듈간의 관계가 복잡해질수록 구현하려는 서비스를 추출하기가 어려워진다. 본 논문에서는 프로그램 소스를 분석하기 위해서 아래와 같은 방법을 사용하였다.

- 프로그램 소스를 분석해서 원하는 기능을 찾고 모듈화 할 수 있는 부분이 있는지 파악한다.

- 프로그램에 사용된 변수나 기타 모듈간의 독립성을 고려한다.

[그림 5]는 프로그램 소스중의 일부로 재사용 가능한 부분을 분석을 통해 추출하는 과정을 보여주고 있으며, factorial 함수가 구현되어 있는 부분을 재사용 가능한 부분으로 추출한다.



[그림5] 소스 코드 분석

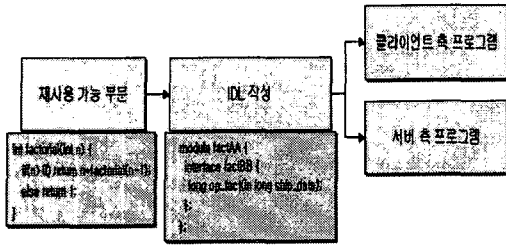
4) 기존 코드의 재모듈화

사용 가능한 서비스를 코드 분석 과정에서 확인한 뒤 이 모듈을 사용 가능하도록 하기 위해서 기존 코드를 재모듈화한다. 재모듈화 과정은 기존 어플리케이션을 수정해야 하므로 모듈간의 관계가 매우 복잡해질 경우 재모듈화가 어렵게 되어 오히려 재모듈화하는데 시간과 노력이 더 많이 들어가는 문제점이 있다.

5) 사용 가능한 모듈의 호출 인터페이스 설계

재모듈화한 서비스부분을 다른 시스템에서 호출할 수 있도록 IDL을 이용하여 호출 인터페이스를 설계하고 클라이언트/서버에서 사용할 어플리케이션을 개발한다.

[그림 6]은 재사용 가능한 부분을 이용하기 위해서 IDL을 작성하고 컴파일하여 클라이언트와 서버가 서로 상호 운용 가능하도록 어플리케이션을 개발하는 것을 보여준다. IDL의 factorial 함수를 호출하는 부분인 "long op_fact(in long ship_data);"가 클라이언트와 서버 사이의 인터페이스 역할을 한다.

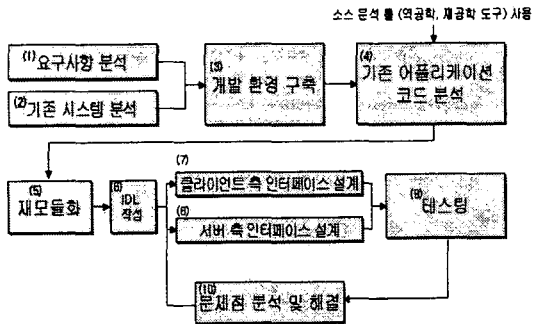


[그림 6] 인터페이스 설계

6) 통합 테스트

분산 이질 시스템 내에서 설계한 프로그램들이 제대로 동작하는지에 대한 통합 테스트 작업을 실시한다. 테스트 작업에 의해서 문제점이 발생되면 문제점을 해결한 뒤 다시 테스트를 한다. 테스트 작업이 성공적으로 끝나게 되면 문서화 작업을 실시한다.

본 논문에서 제시한 상호 운용 절차의 전체적인 흐름은 [그림 7]과 같다.



[그림 7] 상호 운용 절차

5. 결론 및 향후 연구 방향

본 연구에서는 분산 객체 미들웨어 기술을 이용하여 이질적인 군 정보 시스템을 상호 운용 가능한 형태로 전환할 수 있는 절차를 연구하였다.

체계 전환된 시스템 구조에서는 CORBA를 통해서 구현된 객체에 접근할 수 있는 인터페이스만 알고 있으면, 해당 인터페이스만 호출함으로써 서비스를 이용할 수 있다. 그러므로 이질적인 군 정보 시스템에서는 상호운용이 가능하도록 체계 전환을 하기 위해서는 상호 운용하려는 서비스들을 확보하고, 다른 시스템에서 사용하도록 하기 위해 구현객체를 정의하고, IDL을 이용하여 구현객체와 클라이언트사이의 인터페이스를 정의한다. 본 연구에서는 상호

운용 가능한 서비스를 확보하기 위한 방안과 체계전환을 위한 기법을 연구하였다.

향후 연구되어야 할 과제로는 본 논문에서 설계한 시스템 구조를 바탕으로 실제 시스템에서 제시한 상호 운용 기법을 실제로 구현하는 과정이 필요하다.

참고문헌

[1] F. Manola, S. Heiler, D. Georgakopoulos, M. Hornick, M. Brodie "Distributed Object management", GTE Laboratories Incorporated Waltham
 [2] Distributed Object Management Integration System (DOMIS) FY 94 Final Report, MITRE
 [3] M. T. Roth, P. Schwarz, "A Wrapper Architecture for Legacy Data Source", IBM Almaden Research Center
 [4] P.E. Chung, Y. Huang, et. al., "DCOM and CORBA Side by Side, Step by Step, and Layer by Layer"
 [5] T. J. Brando, "Comparing DCE and CORBA", MITRE
 [6] S. Vinoski, "CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments", IONA Technologies. Inc
 [7] Distributed Object Management Integration System(DOMIS) FY96 Final Report, MITRE
 [8] Orfali, R., Harkey, D., and Edwards, J., The Essential CORBA: System Integration Using Distributed Objects, Wiley, 1996