

분산환경에서 CORBA를 이용한 에이전트기반 초기설계 시스템

이동욱*, 이수홍**

An Agent-based Initial Design System in Distributed Environment using CORBA

Lee, D. W.* and Lee, S. H.**

ABSTRACT

This paper deals with the study of distributed design system and the implementation of JAVA based initial design system using CORBA^[1-3]. In the state of initial design, designer should consider other information related to a design part. In this case, designer has a difficult problem to search and calculate distributed data. We propose a special connector named 'Allocator' and implement an initial design system AIDS(Agent based Initial Design system). AIDS can help designer search and calculate the distributed information. In this study, JAVA and CORBA were chosen for handling network and distribute programming. AIDS focuses on the possibility of incremental concurrent design capability. AIDS is composed of Request-Agent, Calculator-Agent, View-Agent, Search-Agent, and 'Allocator'. The proposed system can be used for designing initial design of a gripper in the web regardless of operating system.

Key words : Allocator, CORBA, Agent, Distributed environment, KQML, Collaborative design

1. 서 론

현재 인터넷을 이용하여 분산된 컴퓨터의 자원을 이용하려는 노력이 많이 행해지고 있다. 이러한 노력은 기존의 웹을 이용하여 데이터를 검색하고 저장하는 단계에서 분산된 컴퓨터의 자원을 이용하는 단계로 발전하고 있다. 이를 위해서 현재 CORBA(Common Object Request Broker Architecture)를 이용하여 분산되어있는 자원을 이용하는 시스템들이 선보이고 있으며 기존의 시스템들도 이를 지원하는 시스템으로 변경되고 있는 상태이다. 또한 웹에서 주목 받는 언어로 JAVA를 들 수 있다. 이는 웹에서 프로그래밍을 쉽게 할 수 있다는 장점을 가지고 있어서 현재 많은 연구가 JAVA로 행해지고 있는 상태이다. 웹을 이용하여 초기 설계 시스템을 만드는 과정은 지금까지 여러 분야에서 개발되어왔다^[4]. 하지만 모든 시스템

이 초기에 설계되고 이를 통합하는 개발과정을 거치는 과정에서 지속적인 시스템의 변경과 설계자간의 협의가 필요하다. 이러한 것을 도와주는 툴로는 JATLite^[5]가 에이전트를 이용하여 통신을 지원해주는 툴로 많은 연구에서 응용되고 있다. 동시적인 설계 개발과 함께 하나의 시스템을 이루는 기능이 첨가되는 경우가 있는데 이러한 경우에 전체적인 시스템의 변경이 요구된다. 초기에 디자인 과정에서 보다 올바르게 디자인을 하도록 도와주는 툴로는 MIT에서 연구한 DOME(Distributed Object-based Modeling and Evaluation)^[6]와 AND^[7] 시스템이 있다. 이 시스템은 초기설계에서 모델링과 평가를 해 봄으로써 디자인의 신뢰성을 얻을 수 있다.

기존의 연구들이 분산되어 있는 시스템을 통합하여 분산되어 있는 자원을 이용하는 연구를 해왔다. 하지만 이러한 시스템들은 초기에 시스템의 구조와 데이터를 알고 있을 경우에 초기에 통합적인 시스템을 이루는 한계를 가지고 있다. 다시 말해서 초기 디자인을 위한 시스템을 만드는 경우에 초기 시스템에 대한 모든 정보를 가지고 있어야 한다는 한계가 있

*LG 생산기술원 지식기반센터
**정회원, 연세대학교 기계전자공학부
- 논문투고일: 2000. 3. 29
- 심사완료일: 2000. 9. 22

다. 하지만 현재 시스템의 흐름을 보면 분산 객체의 컴포넌트 기반의 프로그래밍 방법론을 쓰고 있다. 이는 다시 말해서 기존의 구조적인 순차적인 시스템의 개발에서 컴포넌트 중심의 객체 서비스를 이용하는 컴포넌트 방법론으로 개발되고 있다. 하나의 프로그램은 이제 단순히 하나의 프로그램으로서가 아니라 다른 시스템과 서로 통합 될 수 있는 시스템을 이루어야 한다.

본 논문에서는 시스템의 통합 한계를 극복하기 위하여 CORBA를 이용하여 하나하나의 기능을 구현하는 에이전트를 하나의 컴포넌트로 보고 이러한 에이전트의 기능을 컴포넌트로 해석하여 에이전트의 독립적인 의미를 시스템 개발 방법론으로서 이용하고자 한다. 에이전트는 스스로 처리하는 자율기능과 독립적인 기능 등 여러 가지를 가지고 있어야 한다. 이러한 기능들과 분산객체의 CORBA를 이용하여 분산 되어 있는 에이전트들이 각각의 기능을 구현함으로써 각각의 에이전트들간에 좀더 유연한 시스템을 만드는 것이 본 논문의 목적이다. 이러한 기능을 실제로 구현해 봄으로써 현재까지 에이전트를 관리하는 측면에 부족했던 것을 가시화 한다. 또한 실제로 초기 설계 시나리오에 적용함으로써 실제로 이러한 구현 시스템의 필요성을 검증해 본다. 실제로 초기 설계를 지원하는 시스템은 설계의 변경에 따라서 시스템 자체도 변경을 해 주어야 하는 요구가 있다. 이러한 과정이 생기는 경우에 실제로 이를 해결할 수 있는 시스템을 구현함으로써 초기 설계 과정에서 에이전트를 이용한 분산 시스템의 타당성을 증명한다.

2. 에이전트 및 분산 시스템

2.1 에이전트

에이전트는 정확한 정의가 내려진 것이 없다. 하지만 일반적으로 에이전트는 일반적으로 다음과 같은 기능들을 갖는다. 첫째, 대리인으로써 사용자의 작업을 자율적으로 실행해주는 자율성이다. 이는 다른 에이전트를 쓰는 기본적인 이유로서 사람이 하는 반복적인 일이나 논리적인 일을 에이전트를 이용하여 자동으로 일을 처리한다. 이러한 예로서 메일을 이용한 비서 에이전트 같은 것이 있을 수 있다. 메일이 날라오면 자동으로 이를 관련자에게 통보를 해주고 자동으로 답신 메일을 보내며 받은 메일을 체계적으로 데이터 베이스화 하여 나중에 쉬운 검색을 가능하게 한다. 둘째, 지식 베이스를 기반으로 지능을 갖추어 기존의 지식을 가지고 새로운 것을 추론

할 수 있는 특징이 있다. 다시 말해서 기존의 지식들, 즉 데이터 베이스를 이용하여 이러한 데이터베이스를 이용하여 입력된 값을 인식하여 이를 분석하여 새로운 결과값을 도출하게 된다. 이러한 지능적인 기능을 보강하기 위해서 인공 지능의 여러 가지 방법이 이용되기도 한다. 셋째, 에이전트 이동성을 가질 수 있다. 즉 한 군데에서 실행되는 응용프로그램의 차원이 아니라 요즘 논의되고 있는 mobile agent의 기능으로 서버에서 단순히 계산을 해서 결과값을 반환하는 것이 아니라 자바의 애플릿을 이용하여 서버에서 클라이언트로 이동되고 거기서 기능을 실행하여 새로운 결과 값을 도출하게 된다. 이러한 이동성은 기존의 클라이언트/서버와의 관계와는 다른 것으로 클라이언트가 필요로 하는 작업을 위해 에이전트를 서버로 보내어 수행을 시킨다. 넷째, 에이전트는 통신능력으로 단순한 기능의 단독 수행이 아니라 여러 에이전트가 복합적으로 결합하여 새로운 기능을 수행하게 된다. 이러한 에이전트의 통신기능은 에이전트가 통신 언어를 이용한 지식의 전달과 지식의 축적, 에이전트간의 의사 전달을 가능하게 한다. 위의 이러한 기능뿐만 아니라 다른 측면에서 보면 에이전트의 기능 별로 나누어 위의 기능별로 에이전트를 구분하는 방법을 쓰고 있다. 다시 말해서 Collaborative 에이전트, interface 에이전트, mobile 에이전트, information/internet 에이전트, reactive 소프트웨어 에이전트, hybrid 에이전트, heterogeneous 에이전트 시스템으로 나누었다. 이러한 에이전트의 구분은 여러 가지 정의가 있을 수 있겠으나 에이전트는 위의 내용들을 포함하는 기능을 구현하며 특히, 자율성과 통신 능력을 중요하게 여기고 있다.

2.1.1 에이전트 통신언어

에이전트 통신언어(ACL, Agent Communication Language)는 에이전트간에 통신을 위해 쓰이는 언어이다. DARPA(Defense Advanced Research Project Agency)의 KSE(Knowledge Sharing Effort)그룹에서 시스템간 지식의 공유를 위하여 많은 연구가 진행되고 있는데 지식 표현을 위한 공통 언어으로써 통신 규약과 통신언어으로써 KQML을 쓰고 있으며, 내용을 표현에는 KIF가 쓰인다. 에이전트간의 통신을 위해서는 통신규약, 통신 언어, 전송 규약이 관련된다.

2.2 분산시스템

지난 20년간 컴퓨터 관련 하드웨어의 개발은 빠르게 발전 한 반면에 소프트웨어의 개발은 이를 따라잡지 못하고 있다. 이는 소프트웨어 개발자들이 하

드웨어 개발자 만큼 개발을 해내지 못한 것도 있지만 소프트웨어는 계속적인 유지 보수를 위한 시간과 노력이 많이 들기 때문이다. 현재 프로젝트 개발비용을 하드웨어 개발 비용보다 이를 이용한 소프트웨어 개발 비용이 더 많은 것이 현실이다. 이러한 것을 해결하기 위하여 이기종의 운영체제와 프로그래밍 언어, 데이터 베이스를 네트워크를 통하여 개발하려는 것이 컴포넌트를 기반으로 한 분산 객체 프레임 워크이다. 또한 현재 개발되고 있는 많은 소프트웨어가 서로 다른 운영체제와 네트워크 환경에서 개발되고 있기 때문에 이들이 다른 시스템과도 쉽게 통합이 가능하기 위해서 제시된 것이 분산 객체 프레임워크이다. 이러한 분산 객체 프레임워크는 다음과 같은 특징을 가진다. 분산 시스템은 다층의 클라이언트/서버 구조를 가진다^[6].

3. 초기 설계 프로세스

그리퍼 전문 업체는 제품 주문을 맡는 수주부서와 그리퍼를 설계하는 설계부서로 나뉜다. 제품의 설계만을 담당하는 전문회사로 설계분야에서 1인이지만 제품을 직접 만들지는 않는다. 그리퍼 전문업체는 기존의 parallel 그리퍼를 만드는 파트A와 angular 그리퍼를 만드는 파트B, 그리고 이번년도에 새로 진출한 Double Acting 그리퍼(DAG)를 만드는 파트C로 나뉘어 있다. 기존의 parallel 그리퍼의 데이터와 angular 그리퍼 데이터, DAG는 각각 NT 서버와 워크스테이션, Linux에 저장되어 있다. 데이터베이스를 하나로 합쳐서 하나의 시스템으로 만들려고 했으나 기존의 데이터 양이 너무 많고 또한 데이터베이스 시스템이 MS-SQL과 MySQL로 서로 상이한 관계로 하나로 합치기에는 너무나 많은 인력낭비가 필요했다. 이를 해결하기 위하여 CORBA와 Allocator를 이

용하여 기존의 데이터를 손쉽게 가져올 수 있었다. 하지만 한가지 문제가 더 생겼다. 부서 A에서 사용하는 프로그램과 부서 B, C에서 사용하는 프로그램 간에 원활한 인터페이스를 위한 수단이 없었다. 또한 설계부서의 파트 A, B, C에서 작성한 각각의 프로그램이 서로 간 기존의 지식을 교환하기 힘들었다. 초창기에는 A부서에서 필요한 간단한 계산만을 이용하였으나 점차 설계가 복잡해 감에 따라서 부서 B에서는 부서 B에서 필요한 새로운 계산 프로그램을 쓰고 있었다. 그런데 새로운 그리퍼 설계에서는 부서 C에서의 새로운 프로그램을 이용하는 것 뿐 아니라 기존의 부서 A, B에서 이용한 계산 프로그램이 필요했다. 이를 해결하기 위하여 KQML을 이용한 에이전트기반의 시스템을 구축하였다. 수주부서는 주문 받은 수주자료를 바탕으로 검색에이전트를 이용하여 부서별로 분산된 회사의 데이터베이스를 검색하게 된다. 분산된 데이터 베이스의 상이한 입출력 규약에 따른 정보 교환의 문제점을 해결하기 위하여 풍부한 지식 표현이 가능한 KQML을 이용하였다.

4. AIDS (Agent-based Initial Design System)

4.1 AIDS의 구조

AIDS는 Agent-based Initial Design System의 약자로서 에이전트 기반으로 초기 설계를 할 수 있도록 해주는 시스템이다. AIDS는 웹 기반의 시스템을 구현하기 위하여 플랫폼에 독립적이고 웹상의 프로그램이 가능한 자바 언어를 사용하였다. 또한 에이전트 기반 시스템을 구현하기 위하여 JATLite를 이용하여 에이전트 간의 통신을 가능하게 하였다. 그리고 이기종 언어에서의 에이전트 확장을 증명하기 위하여 C++관의 연동을 CORBA를 통하여 구현하였

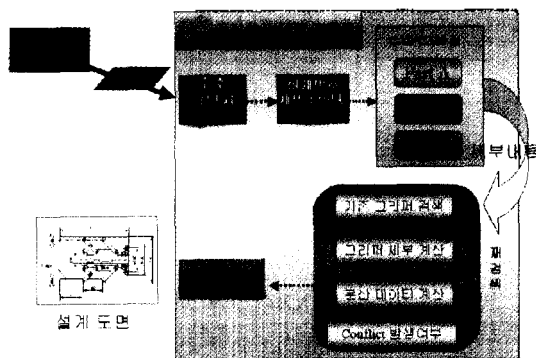


Fig. 1. Scenario of gripper's Initial Design.

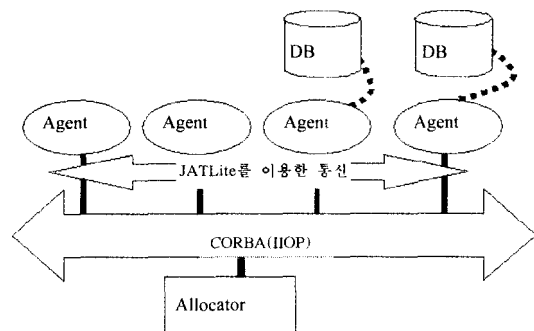


Fig. 2. Architecture of AIDS. (Agent based Initial Design System)

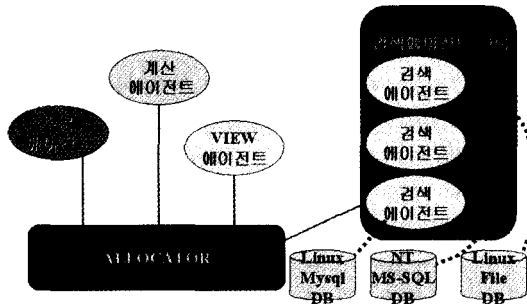


Fig. 3. AIDS's Agents and Allocator.

다. AIDS 시스템 구조는 Fig. 3과 같다.

위의 그림을 보면 에이전트간의 간단한 통신을 위해서는 JATLite를 사용하였고 분산 에이전트의 함수를 호출하기 위해서는 CORBA(IIOP)를 이용함으로써 단순한 통신과 객체의 메소드 호출을 분리하였다. 이렇게 함으로써 지식 전달의 KQML을 복잡한 CORBA(IIOP)의 IDL로 정의하여 통신 언어의 추가로 인하여 시스템 레벨의 하위 구조까지 다시 작성하여야 하는 번거로움을 없앴다. 각각의 에이전트는 독립된 기능을 하며 각각의 에이전트는 전문화된 기능을 하면서 서로간의 협동 작업을 위하여 통신을 통하여 항상 서로간의 대화의 창은 열려있게 된다. 그리고 분산된 에이전트들에서 각각의 에이전트에 대한 특수한 기능을 이용하기 위해서는 Allocator에게 요청을 하게 되고 Allocator는 기능에 맞게 각각의 에이전트에게 전달하게 된다. 따라서 에이전트는 에이전트의 능력만을 알고 있으면 된다. 에이전트는 각각의 에이전트에게 JATLite를 이용하여 협동 설계 시의 문제나 기타 예외 상황이 발생 할 경우에 신속히 대처할 수 있게 된다. JATLite는 KMQL을 통하여 각각의 에이전트의 요구 의도를 파악할 수 있으며 설계 도중에 필요한 정보들을 손쉽게 교환할 수 있다.

4.2 AIDS의 구성 에이전트

AIDS의 각각의 에이전트는 다음과 같이 구성된다. 초기에 웹상으로 주문을 받는 주문 에이전트, 주문된 정보를 바탕으로 검색을 하는 검색 에이전트, 세부 설계를 위한 계산을 수행하는 계산 에이전트, 설계에서 만들어진 그리퍼의 프로토타입을 보여주는 View 에이전트, 마지막으로 AIDS의 연결자 역할을 하는 Allocator가 있다. AIDS의 구성을 간단하게 나타내면 아래의 Fig. 4와 같다.

검색 에이전트와 각각의 에이전트는 인터넷을 통하여 각각의 정보를 교환할 수 있다. 또한 JATLite를

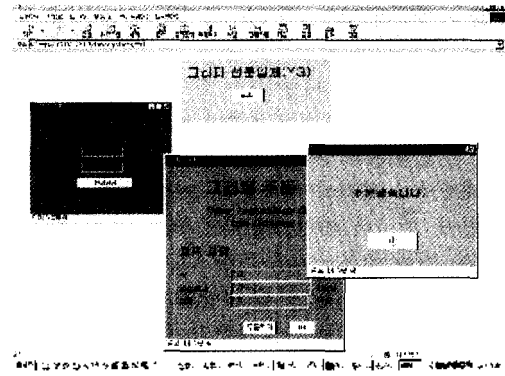


Fig. 4. Request Agent.

이용하여 에이전트 상호간의 간단한 통신을 할 수 있으며 분산된 데이터를 검색 에이전트를 통하여 손쉽게 검색을 할 수 있다.

4.2.1 주문 에이전트

주문 에이전트는 주문이라는 기능 상 많은 주문을 받고 또한 접근 편리성을 고려하여 웹상으로 누구나 접근 가능한 환경하에서 기능을 수행한다. 현재 웹 브라우저는 컴퓨터의 일부분 일 만큼 널리 퍼져있으며, 브라우저를 이용한 웹 접근의 수요 또한 기하 급수적으로 늘어나고 있는 상황이다. 이러한 환경을 고려하여 주문 에이전트는 웹상에서 주문을 받게 된다. 주문 에이전트는 웹상에서 접근이 가능하며 그리퍼의 그래픽을 입력하면 그리퍼 에이전트는 이를 데이터 베이스에 저장하여 제품의 설계를 위한 중요 스펙으로 쓰게 된다. 이는 주문자의 스펙에 대한 자료를 데이터 베이스에 저장함으로써 그리퍼 설계시에 참조 자료로도 사용될 수 있다.

4.2.2 검색 에이전트

검색 에이전트는 주문된 자료를 바탕으로 주문 에이전트가 저장 해 놓은 정보를 바탕으로 기존의 제품을 검색 하게 된다. 주문자의 제품 스펙을 건딜 수 있는 기존의 제품을 찾아서 기존의 제품 설계를 참조하게 된다. 제품의 실제에서 기존의 제품의 설계를 검색하는 것은 단순한 것이 아니다. 본 논문에서는 서로 분산되어 있는 그리퍼 정보를 검색하기 위해서 CORBA를 이용한 검색 에이전트를 이용하였다. 검색 에이전트는 이기종의 환경에 분산되어 있는 제품을 검색하기 위해서 각각의 분산 서버에 하나의 검색 에이전트를 구현하고 이들을 검색 에이전트 그룹으로 분류하여 각각의 분산된 자료의 특성을 부여하였다. 다시 말해서 검색 에이전트 1은 그리퍼의 part A에 관한 자료를 검색하기 위한 에이전트로

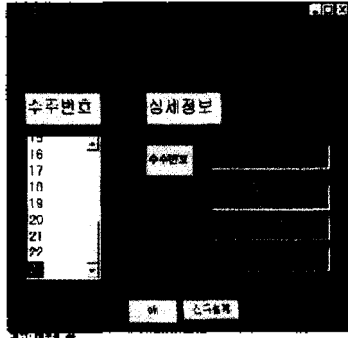


Fig. 5. Request Information by Client.

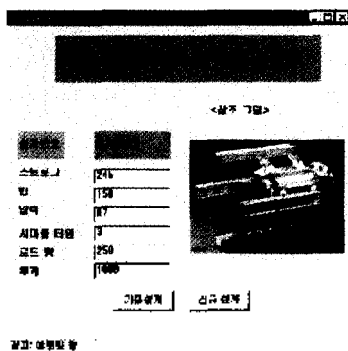


Fig. 6. Information searched by Search Agent.

구현하였고, 검색 에이전트 2는 그리퍼 part B에 대한 자료, 검색 에이전트 3는 part C에 대한 자료를 검색하게 구현하였다.

따라서 각각의 부품에 대한 것에 대한 검색을 위해서는 Allocator가 자동으로 검색하여 줌으로써 나중에 part A만 필요하거나 또 다른 part가 첨가되었을 때 각각에 대한 검색을 독립적으로 분리 할 수 있다. 이는 시스템의 변경이 일어났을 때 좀더 효율적으로 시스템을 구현할 수 있는 측면이 있다. 검색 에이전트는 시스템의 규모에 따라서 계속적으로 변경이 일어나고 또한 제품의 수정에 따른 변경이 많이 일어나기 때문에 구현상의 독립성이 보다 중요하게 여겨진다. 본 논문에서의 시나리오에서도 그리퍼 회사의 제품의 기술적인 발전으로 인하여 제품의 부품 개수가 점차 늘어나는 상황 때문에 지속적인 그리퍼 데이터의 증가와 함께 기존의 분산 된 데이터 들 효율적으로 관리하기 위하여 검색 에이전트를 구현하였다.

4.2.3 계산 에이전트

계산 에이전트는 설계 상에서 필요한 제품에 대한 물리적인 계산을 해주는 에이전트로 계산 에이전트

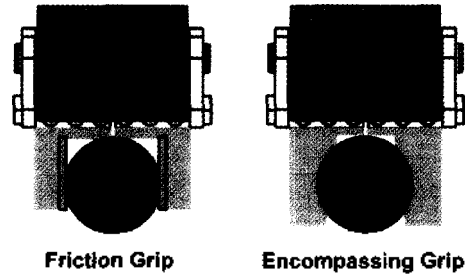


Fig. 7. Gripper Type.

는 그리퍼의 토크에 대한 계산을 하게 된다.

본 논문에서는 마찰(Friction) 그리퍼와 Encompassing 그리퍼의 두 가지 종류의 계산을 위해서 그리퍼의 설계를 위한 계산을 하게 된다. 기본적으로 그리퍼가 필요한 힘은 다음과 같은 물리적인 식으로 구할 수 있다.

4.2.4 Allocator

AIDS 시스템을 이루는 각각의 에이전트는 에이전트의 특성상 각각의 자율적인 기능을 가지고 있다.주문 에이전트, 검색 에이전트, 계산 에이전트, View 에이전트등은 각각 주문, 검색, 계산, View의 기능을 구현하도록 구현되었다. 하지만 이러한 각각의 에이전트가 하나의 통합된 시스템(AIDS)를 구현하기 위해서는 각각의 에이전트간에 의사소통을 할 수 있고 또한 어떠한 세의 상황이 발생 했을 때 이를 수정, 보완하여 시스템의 견고성을 지켜야 한다. AIDS에서는 에이전트의 통신 수단으로 JATLite를 이용하여 서로간의 의사 소통을 가능하게 하며, 서로간의 지식 전달을 위해서 KQML을 이용하고 있다. 여러 에이전트를 관리하고 에이전트 자체를 검색하는 개념으로는 Facilitator가 있었다. 하지만 기존의 facilitator는 에이전트의 단순한 관리에 그치고 있다. 기존의 로컬 환경에서만 에이전트의 등록이 가능한 한계가 있었다. 하지만 분산 환경에서 각각의 에이전트에 대한 연결 투명성을 제시해주지는 못했다. 연결 투명성을 제시해 주지 못 할 경우에 에이전트의 협동 설계가 한계를 갖게 된다. 에이전트의 자율성과 독립성을 최대한 보장하기 위해서 각각의 에이전트는 확장될 수 있어야 한다. 초기의 시스템 구축과정에서 구현된 에이전트 시스템은 기존의 시스템에 새로운 에이전트의 추가나 변경이 생길 경우에 많은 변경 없이 좀더 큰 시스템으로 확장 가능해야 한다. 이를 위하여 본 논문에서는 분산 에이전트를 관리할 수 있고 또한 상호간의 브로커 역할을 할 수 있는 Allocator 를 제안하였고 이를

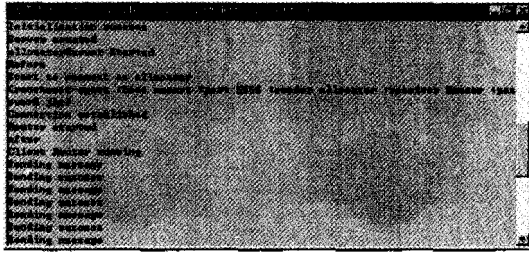


Fig. 8. Message sending by Allocator.

구현하였다.

4.3 Allocator의 기능

본 논문에서 제시한 Allocator는 기존의 에이전트의 단순한 관리가 아니라 에이전트에 대한 identity를 CORBA의 IOR과 바인딩 시켜서 분산된 환경에서 자유롭게 에이전트의 고유성을 지킬 수 있다. 또한 에이전트간의 상호 협동을 위하여 Allocator가 다른 에이전트에 대한 교량 역할을 해 준다.

새로운 에이전트의 첨가나 변경을 쉽게 고칠 수 있고 분산되어 있는 환경에서 시스템의 변경이 용이하다. Allocator의 실행순서는 다음과 같다. Allocator는 각각의 에이전트를 IOR로 관리하여 인터넷을 통하여 분산된 에이전트의 요청을 각각의 에이전트에게 전달한다. 기존의 facilitator는 서버 하나의 에이전트에 대해서만 협동설계를 가능하게 하였으나 AIDS에서는 CORBA라는 기술을 이용하여 하나의 서버에 국한되지 않고 모든 CORBA객체인 에이전트와 협동설계를 가능하게 한다.

5. 결론 및 향후과제

본 연구에서는 CORBA의 통신 투명성과 에이전트의 특징을 이용하여 초기 설계를 가능하게 하는 시스템을 만들었다. 기존의 초기 설계 시스템은 단순한 하나의 모듈을 개발 하여 이들을 초기에 통합하여 기존 시스템의 확장을 어렵게 하였다. 하지만 본 논문에서 구현한 AIDS 시스템은 기존 시스템의

통합과정에서 가장 어려운 에이전트 간의 통신 부분을 CORBA를 이용한 Allocator를 통하여 해결하였다. 기존의 자바 언어를 통하여 이기종간의 개발 환경이 가능하였지만 시스템의 안정을 위한 보안으로 인하여 자바만의 이기종 개발에는 많은 한계를 가지고 있었다. 또한 기존의 CORBA관련 개발 시스템에서는 한 회사 제품을 사용하여 vendor에게 구속되는 개발 한계가 있었다. 하지만 본 논문에서는 이러한 한계를 Allocator를 통하여 극복하고 진정한 의미에서의 개발 투명성을 구현하였다. 또한 에이전트의 개발 방법론을 이용하여 시스템의 기능적인 확장을 가능하게 한 시스템을 구현하였다. 초기 설계에서 필요한 제품의 수주와 기존 제품의 검색, 상세 설계에서 필요한 계산, 설계를 검토해 볼 수 있는 각각의 에이전트를 이용하여 가상의 시나리오를 통하여 에이전트간의 협동 설계를 검증하였다.

참고문헌

1. "Common Object Request Broker Architecture and Specification Revision 2.3", Oxford University Press, Object Management Group, 1996.
2. Jungfer, K. and Patricia, R. T., "Maplet: a CORBA-based Genome map viewer", proceedings of ASME, Vol. 14, No. 8, pp. 734-738, 1998.
3. "CORBA", <http://www.omg.com>.
4. 백종구, 박종태, "WWW와 CORBA를 이용한 원격의료진단시스템의 설계 및 구현", 경북대학교 대학원위원회, pp. 7-9, 1996.
5. "JATlite", <http://java.stanford.edu/>.
6. Pahng, F., Senin, N. and Wallace, D.R., "Distributed Modeling and Evaluation of Product Design Problems", *Computer-Aided Design*, 1998.
7. Mohammad, R.D. and Yan, J., "AND: An Agent-based Decision Network For Concurrent Design And Manufacturing", *Proceedings of ASME*, 1999.
8. Oobert, O. and Dan, H., "Client/Server Programming with JAVA and CORBA", John Wiley & Sons, pp. 61-64, 1998.
9. 왕창중, 이세훈, "CORBA3 프로그래밍", 대림출판사, pp. 145-206, 1999.



이 동 옥

1998년 고려대 기계공학과 학사
2000년 연세대 기계공학과 석사
2000년-현재 LG생산기술원 지식기반센터
연구원
관심분야: 분산협동설계, 지식기반시스템,
인공지능, 데이터 마이닝



이 수 현

1981년 서울대 기계공학과 학사
1983년 서울대 기계공학과 석사
1991년 Stanford대학 Design Division Con-
current Engineering 전공, 박사
1991년~1992년 Lockheed Missile and Space
Co. Cable Harness Design System개
발 Post-Doc.
1983년~1994년 KJMM CAD/CAM실, 선임
연구원
1994년-현재 연세대학교 기계공학과, 부교수
관심분야: 동시공학설계, 지식기반시스템설계,
DFM