

# JAVA와 CORBA 기반의 주문 물품 모니터링 시스템 구현

정기완\*, 최정규\*, 변광준\*, 윤영태\*\*, 채승기\*\*, 서상일\*\*, 백종현\*\*  
\*아주대학교 \*\*대우정보시스템 기술연구소

## An Implementation of an Ordering Status Monitoring System Based on JAVA and CORBA

Ki-Wan Jung\*, Jeong-Gyu Choi\*, Kwang-Joon Byeon\*,  
Young-Tae Yoon\*\*, Seung-Gi Chai\*\*, Sang-Il Seo\*\*, Jong-Hyun Back\*\*  
\*AJOU University, \*\*Institute of Information Technology at DAEWOO  
Information Systems Co.

### 요 약

최근 기업의 경쟁력 확보를 위해 기업내의 분산된 이질적인 정보 시스템을 전사적차원에서 통합을 하려는 노력이 활발히 진행되고 있다. 그 중 많은 경우에 통합 문제를 향후의 IT 환경을 고려하여 객체지향 개념의 JAVA 및 CORBA를 기반으로 해결하려 하고 있다. 하지만, 실제 기업 환경에 적용된 경우는 아직은 드물다. 본 논문에서는 바이어가 전자회사에 주문한 물품의 생산 현황, 출하까지의 예상 시간, 배송일 등을 실시간으로 확인할 수 있는 시스템인 주문 물품 모니터링 시스템(DW-GOMS: DaeWoo - Global Ordering status Monitoring System)의 구현에 대해 설명한다. 이 시스템에 대한 사용자의 요구사항 분석은 Use Case를 이용했고, 분석 및 설계는 OMT 프로세스를 도입하였고 그 결과를 UML로 표기하는 등 객체지향 개발 방법론을 적용했으며, 설계된 결과를 바탕으로 JAVA 및 CORBA를 기반으로 구현하였다. 구현된 시스템은 CORBA를 적용하여 개발 속도를 현저히 줄일 수 있으며, 시스템 분석/설계에 객체지향 개발 방법론을 적용함으로써 향후 시스템 변경에 유연하게 대응할 수 있게 된다.

### 1. 서론

기업 정보 시스템 환경은 분산 컴퓨팅 환경 특히 클라이언트/서버 환경으로 빠르게 발전했다. 클라이언트/서버 환경은 서버 자원을 쉽게 공유할 수 있고 네트워크 기반으로 운용됨으로써 시스템 구성이 용이하다는 장점을 갖는다. 그러나 수많은 이기종 시스템으로 컴퓨팅 환경이 구성됨으로써 각 시스템에 종속적인 인터페이스를 작성해야 하며, 여러 플랫폼에서 상호 작동할 수 있도록 시스템을 통합해야만 하는 문제점이 발생한다.

OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture) 표준안은 클라이언트와 구현 객체간에 IDL(Interface Definition Language)을 이용하여 시스템 독립적인 인터페이스를 작성할 수 있도록 하며,[4,5,6] 실제 구현된 또는 구현될 프로그래밍 언어와 플랫폼에 독립적으로 시스템을 구축할 수 있는 시스템 통합 기반 기술을 제공함으로써 위의 문제점들을 해결할 수 있다.

본 논문에서 구현한 주문 물품 모니터링 시스템(DW-GOMS: DaeWoo - Global Ordering status

Monitoring System)은 국,내외 바이어의 주문에 대하여 주문 물품의 생산 현황과 출하까지의 예상 시간 및 배송일 등을 실시간으로 확인할 수 있는 분산 모니터링 시스템이다. DW-GOMS는 유즈케이스(Use Case)로 사용자 요구사항을 분석했으며, 시스템 분석과 설계는 OMT(Object Modeling Technique) 프로세스를 적용하고 그 결과를 UML(Unified Modeling Language)로 표기했다. 시스템 구현은 CORBA와 JAVA 언어를 이용했다.

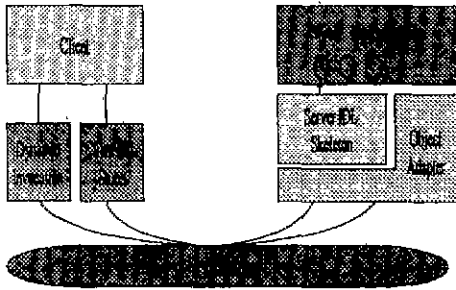
본 논문은 DW-GOMS 구현에서 핵심 기술인 CORBA와 JAVA에 대해서 2장에서 설명한다. 3장에서는 DW-GOMS에 대한 개요와 객체지향 개발 방법론을 이용한 분석 및 설계에 대한 설명과 그 결과를 기술한다. JAVA와 CORBA를 이용한 시스템 구현과 시스템 동작에 대해서는 4장에서 설명하고 향후 연구 과제와 결론은 5장에서 제시한다.

### 2. CORBA와 JAVA

CORBA는 OMG에 의해 작성된 분산 객체 판리를 위한 표준안이며, 객체가 실행되는 방법이나 객체가 실행되는 곳을

물라도 다른 객체와 상호 동작할 수 있도록 프로그래밍이 가능하게 할 수 있는 객체지향 미들웨어이다.[5,6] CORBA 표준안의 핵심은 ORB(Object Request Broker)와 IDL이다.

ORB는 클라이언트와 구현 객체간의 중계자 역할을 제공하는 여러 가지 컴포넌트와 인터페이스로 구성된다.[5] ORB는 분산 객체를 클라이언트/서버 관제로 만들어 주는 객체 미들웨어이며, 클라이언트 객체의 호출을 적절한 구현 객체에 대한 호출로 연결하고 구현 객체가 실행한 결과를 클라이언트 객체의 호출 결과로 전달하는 역할을 한다. [그림 1]은 OMG의 ORB 구조이다.



[그림 1] ORB 구조

IDL은 CORBA 클라이언트 객체와 구현 객체간의 인터페이스를 정의하는 언어이다.[5,6] CORBA는 정적 IDL과 동적 인터페이스를 지원한다. 정적 IDL은 컴파일할 때 스태브와 스펬리튼을 이용하여 클라이언트 객체와 구현 객체간의 호출이 결정되는 방식이며 컴파일 시간에 미리 호출되는 객체가 정해질 경우에 사용한다. 동적 인터페이스는 프로그램이 실행 중에 호출하고자 하는 구현 객체를 찾아서 호출할 수 있는 방식으로 실행시간에 새로운 객체를 호출할 필요가 있는 경우에는 동적 인터페이스를 이용한다.

JAVA는 플랫폼에 독립적인 프로그래밍 언어이며, 네트워크를 통해서 코드를 이동할 수 있는 특성을 갖는다.[3] 또한 JAVA는 애플릿이라고 하는 웹 브라우저에 의해서 실행될 수 있는 프로그램을 작성할 수 있다.

CORBA는 이런 모빌 코드 기능과 웹 브라우저에 플러그인 할 수 있는 JAVA 클라이언트를 이용하여 인터넷과 인트라넷으로 보편화된 웹 어플리케이션 환경에 적합한 해결책을 제공한다. 이 기술을 ObjectWeb이라고 한다.[5,9] 본 논문에서는 ObjectWeb 기술을 이용하여 DW-GOMS를 구현한다.

### 3. 시스템 분석 및 설계

#### 3.1 시스템 개요

DW-GOMS에서 바이어는 바이어 등록을 요청하고 제공되는 제품 리스트를 보고 주문 물품을 선택한다. 관리자는 바이어 등록 요청을 승인 또는 거부할 수 있으며, 등록이 허가된 바이어만이 주문을 요청할 수 있다. 또한 관리자는 주문된 내용을 승인, 취소할 수 있으며, 승인된 주문 물품은 생산 계획이 수립되고 공장에서 생산된다. 필요한 경우 바이어는 개략적인 주문 물품의 생산 및 배송 현황을 확인할 수 있으며, 관

리자는 상세한 주문 물품의 생산 공정을 확인할 수 있다. [8]

#### 3.2 유즈케이스를 이용한 사용자 요구사항 분석

시스템 분석의 첫 단계인 객체 식별을 위해서는 사용자 요구사항 분석이 선행되어야 한다. 유즈케이스 기법은 외부에서 시스템을 보는 관점을 제공하며, 이는 시스템에 대한 요구 사항을 표현하는데 유용하므로 우리는 이것을 이용하여 사용자 요구 사항을 분석한다.[2,7] DW-GOMS에 존재하는 액터는 바이어, 관리자 그리고 공장 시스템이다. 바이어는 관리자에 의해 시스템에 등록이 허가된 사용자로서 생산하는 제품을 주문하려는 사용자이다. 관리자는 바이어 정보의 관리, 제품 주문에 대한 승인 등을 담당하는 사용자이고, 공장 시스템은 현재 구축되어 있는 시스템으로서 제품의 상태 정보를 제공하는 역할을 한다.

#### 3.3 시스템 분석 및 설계

DW-GOMS의 분석과 설계는 현재 국내에서 많이 사용되고 있는 객체지향 개발 방법론의 하나인 OMT를 이용하며 결과를 UML로 표기한다. OMT는 객체 모델링, 동적 모델링 그리고 기능 모델링에 의해 시스템을 분석하고 설계한다.[1]

DW-GOMS의 분석에 의해서 여러 시퀀스(Sequence) 다이어그램과 객체 다이어그램을 생성한다. 예를 들면 바이어가 주문한 물품에 대한 현재 상태를 확인하기 위한 과정을 주문 추적 시퀀스 다이어그램을 통해서 나타낸다.

시스템 분석 단계에서 작성한 객체 다이어그램은 실제 시스템 구현에 적용될 수 있도록 상세한 정보를 포함하여 클래스 다이어그램을 작성하며, 이것을 이용하여 패키지 다이어그램을 작성한다.

### 4. 구현

#### 4.1 IDL 작성

3장의 시스템 분석 및 설계의 산출물은 케이스(CASE) 툴인 패러다임 플러스(Paradigm Plus)를 이용하여 도식화 한 후 CORBA 어플리케이션을 위해 OMG IDL을 작성한다. IDL 인터페이스는 다음과 같이 시스템 사용자의 관점에서 Buyer와 Manager 그리고 Control 부분으로 크게 나눌 수 있다.

##### ● Buyer

```
interface C_Buyer // 바이어 객체를 생성해 준다
interface C_Buyer_Register_Form//신규 바이어의 정보 등록.
interface C_Buyer_Service_Form//바이어에게 사용 가능한 서비스를 출력한다.
interface C_Change_Buyer_Info_Form//바이어 정보 변경 폼
```

##### ● Manager

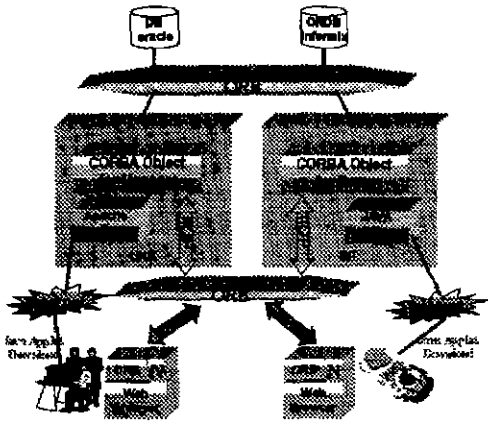
```
interface C_Login_Form//시스템을 사용하고자 하는 접근자의 인증 과정
interface C_Manager//관리자 정보를 얻는다.
interface C_Manager_Service_Form//관리자가 접속했을 경우 사용 가능한 서비스를 보여준다.
interface C_New_Buyer//신규 바이어
interface C_New_Buyer_Detail_Form//신규 바이어 상세 폼
interface C_New_Buyer_List_Form//신규 바이어 리스트 폼
interface C_Product//제품 - 제품 정보에 대한 생성 및 관리
```

● Control

```
interface C_Order//주문 클래스
interface C_Order_Info_List_Form//주문 정보 리스트 폼
interface C_Order_Product_Detail_Form//주문 제품 상세 폼
interface status//제품 현황
```

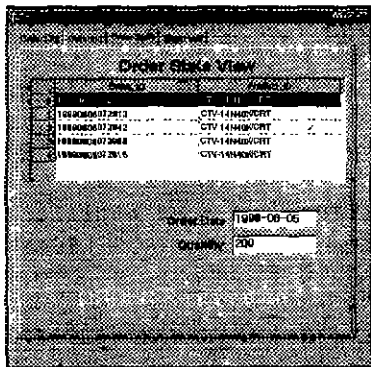
4.2 시스템 동작

DW-GOMS는 관리자를 위한 관리자 구현 객체를 위해 SUN Solaris 워크스테이션을 사용하며, 바이어에게 제공되는 서비스들을 제공하기 위한 바이어 구현 객체는 MS Windows NT 4.0에서 동작한다. 바이어와 주문 정보를 관리하기 위해서 오라클 7.3.3 RDBMS를 이용하였고, 제품 정보 관리는 인포믹스 9.1 ORDBMS를 이용했다. CORBA ORB는 Inprise 사의 Visibroker for java 3.2를 이용하였고 Borland JBuilder C/S 1.0.1을 이용하여 인터페이스를 개발했다. [그림 2]는 DW-GOMS의 구조이다.



[그림 2] DW-GOMS 구조

[그림 2]와 같이 DW-GOMS는 시간과 공간에 관계없이 웹에 접근할 수 있는 환경에서 실시간으로 제품 정보를 보고 주문하고 주문 물품에 대한 정보를 모니터링할 수 있다. [그림 3]은 제품을 주문한 바이어가 현재 주문 물품에 대한 상태를 확인하고 있는 모습이다. 이 상태를 보고 바이어는 현재 몇 %의 공정이 진행 중이며, 주문 물품이 언제 출하될 것인지 실시간으로 확인한다.



[그림 3] 바이어의 주문 물품 상태 리스트

5. 결론 및 향후 계획

본 논문에서는 CORBA와 JAVA를 이용하여 분산 컴퓨팅 환경에서 바이어와 관리자를 위한 DW-GOMS를 구현한다. DW-GOMS는 이기종 분산 환경에서 시스템을 통합하고 분산된 객체들을 관리하는 CORBA 기술을 이용하여 이기종 데이터베이스 접근과 네트워크 프로토콜 제어 등의 분산 프로그래밍의 복잡함을 쉽게 함으로써 시스템의 개발 속도를 향상시켰다.[4,6] 또한 시스템 분석과 설계에 도입된 객체지향 개발 방법론은 충분한 분석과 설계가 수행될 경우 재사용 가능한 시스템 모델이 구축됨으로써 계속되는 업무 환경 변화에 빠르게 대처할 수 있다.[1,2,7]

DW-GOMS에서 관리자는 생산한 제품에 대한 정보를 홍보와 판매 목적으로 입력시킬 수 있으며 고객보다 더 상세한 주문 물품에 대한 정보를 확인할 수 있다. 바이어는 웹에 접속할 수 있는 환경에서 언제 어디서나 제품 주문과 주문 물품 모니터링을 할 수 있다. 결과적으로 DW-GOMS는 기존 시스템의 주 사용자인 관리자 뿐 아니라 바이어까지도 만족시킬 수 있다.

본 논문에서 구현한 DW-GOMS는 이기종 분산 정보를 처리하기 위해 시스템 통합 기술인 CORBA 기술을 사용함으로써 글로벌하게 발전하는 새로운 기업 환경에 대한 해결책이 될 것이다. 또한, 계속적으로 업무 요구사항이 변화하는 기업의 업무 시스템에서 점점 그 중요성이 커지는 소프트웨어의 재사용성을 증진시키기 위해 객체지향 개발 방법론을 적용함으로써 재사용성이 높은 기업의 업무 소프트웨어를 개발하려는 개발자들에게 좋은 사례가 될 것이다. 차후에 소프트웨어의 개발에 지속적으로 객체지향 개발 방법론을 적용함으로써 소프트웨어 특성에 적합한 절차를 정립할 것이며, 분산 환경에서의 트랜잭션 관리를 위한 기능을 구현할 것이다.

참고 문헌

- [1] James Rumbaugh, et al., Object-Oriented Modeling and Design, Prentice Hall, 1991.
- [2] Jacobson I, Chrisserson M., Jonsson P, and Overgaard G., Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, 1992.
- [3] Ken Arnold, James Gosling, "The Java Programming Language 2nd Edition", Addison Wesley, 1997
- [4] Robert Orfali, Dan Harkey, "Client/Server programming with JAVA and CORBA", 65-115, Wiley Computer, 1997
- [5] Robert Orfali, Dan Harkey, Jeri Edwards, "Instant CORBA", Wiley Computer, 1998
- [6] Seigel, Jon, "CORBA Fundamentals and Programming", Wiley Computer, 1996
- [7] 대림정보통신㈜ 컨설팅센터, "객체란 무엇인가", 전자신문사, 1997.
- [8] 대우전자㈜, "국내생산공장 현황 분석서(TV공장)", 대우정보시스템 CALS 추진 1부, 1997. 12.
- [9] 박재현, "코아 코바", 영한 출판사, 1998