

전자 상거래용 카탈로그 저장소를 위한 컴포넌트 기반 아키텍처

윤경환*⁰ 오재원* 홍영준† 박근덕* 이상구** 우치수*

* ** 서울대학교 컴퓨터공학부

† (주) JTEL

* {life, jwoh, dean, wuchisu}@selab.snu.ac.kr

yjhong@jtel.com

** sglee@mars.snu.ac.kr

A Component-based Architecture Suitable for Repositories of Online Catalogs for Electronic Commerce

Kyung-hwan Yoon*⁰ Jaewon Oh* Youngjoon Hong†
Geunduk Park* Sang-goo Lee* Chisu Wu*

* School of Computer Science and Engineering

† JTEL Co., Ltd

Seoul National University

요 약

상거래의 상품 정보를 기술하는 카탈로그는 전자 상거래의 상품 인지 단계에서도 중요한 역할을 하며 대부분의 인터넷 쇼핑 물에서 전자 카탈로그 저장소의 구축과 유지를 위해 많은 비용을 소모한다. 전자 상거래가 활발해지고 인터넷 쇼핑 물이 급격히 늘어남에 따라 구축과 유지에 많은 비용이 드는 전자 카탈로그 저장소를 공유할 필요가 생겨났으며, 그에 걸맞는 소프트웨어 아키텍처가 요구되고 있다. 이 논문에서는 서로 이질적인 전자 카탈로그 저장소들이 분산되어 있는 환경에서도 각각의 카탈로그들이 상호 연동할 수 있는 컴포넌트 기반 아키텍처를 제시한다. 이 아키텍처에서 컴포넌트들은 CORBA IDL을 통해 통신을 하며 이로써 이질적인 전자 카탈로그 저장소의 상호연동이 가능해지며 시스템의 신속성과 확장성 역시 높아지게 된다.

1. 서론

전자 상거래는 주문, 배송, 지불 등 여러 역할을 하는 상거래 주체들 사이의 많은 상호작용 속에서 이루어진다. 상거래 행위는 참여 주체들의 상호작용에 따라서 몇 가지 단계로 나누어 볼 수 있는데, 그 중 첫번째 단계는 상품의 구매자가 상품과 거래 상대에 대한 정보를 얻는 상품 인지 단계이다. 전자 카탈로그는 상품과 상품 제조업자에 대한 정보를 고객에게 제공하는 역할을 함으로써 상행위의 첫번째 과정인 인지 과정에서 중요한 역할을 하게 된다[1].

현재 전자 상거래의 영역에서 사용되고 있는 전자 카탈로그는 일반적으로 상품이나 서비스 등에 대한 설명을 비롯하여 구매 여부를 결정하는데 매우 중요한 가격 정보를 포함하는 모든 웹 페이지를 의미하고 있다[2].

그러나 순수한 웹 페이지를 직접 상품 카탈로그로 사용하는 데에는 많은 어려움이 있다. 웹을 이용할 경우 웹 문서가 가지고 있는 정보의 형식과 그 전달 방법의 표준이 정해져 있지 않기 때문에 해당 카탈로그를 공유의 정보로 사용하기에는 한계가 있고, 상품 종류의 변화나 상품 정보의 변화 등 상품 시장의 변동에 대비하기도 어렵다. 현재 각각의 상품 공급자는 각자의 카탈로그를 가지고 있으며, 이 카탈로그들의 형태는 표준화되지 않은 매우 이질적인 상태로서 전자 상거래 활성화의 가장 주된 문제점 중 하나인 것이다. 예를 들면, 상품 고객이 어떤 상품을 구매하고자 할 때 원하는 상품 공급자의 주소를 알아야 하며 관련된 모든 상품 사이트를 검색해야 한다. 하지만 이런 정보는 각 공급자 마다 다르게 저장되어 있기 때문에 고객이 원하는 상품에 대한 정보를 찾는 일 자체가 쉽지 않으며, 같은 의미를 가진 정보(예를 들어 상품의 가격 정보나 기능 설명)라도 카탈로그에 따라서 다른 형식으로 기록될 수 있기 때문에 같은 종류의 상품들을 비교하여 상품을 선택하고자 할 경우

에 이러한 웹 문서 형식의 상품 카탈로그를 이용하여서는 정보의 비교를 쉽게 할 수 없다.

이러한 상호 연동성의 문제점을 극복하기 위해서 MEPC (Mediating Electronic Product Catalog)[3]와 smart catalogs and virtual catalogs[4] 등의 방법이 제안되었다. 또한 상호 연동성을 보장하는 카탈로그를 위해 CommerceNet의 Information Access Portfolio는 PIX (Product Information Exchange)를 제안하였다. PIX 프레임워크는 카탈로그의 상호 연동성을 이루기 위해 카탈로그를 콘텐츠(content), 커뮤니케이션(communication), 포맷(format), 프레젠테이션(presentation)의 네 부분으로 구성하고 있다[5].

또한 카탈로그 구축 비용의 측면을 고려할 때 일반적으로 전자 상점들은 상점을 구축할 때 드는 초기 비용 중 절반 이상을 제공하는 상품과 서비스에 대한 카탈로그를 만드는데 들고 있다[6]. 인터넷 상점의 입장에서는 상품 카탈로그의 유지 보수에 드는 비용이 크고, 상품 제조업자의 입장에서는 만든 상품에 대한 정보를 개별 상점에 직접 보내야 하는 비용과, 새로운 상품이 만들어 지거나 상품에 대한 수정 사항이 있을 때마다 그 정보를 각각의 상점에 보내야 하는 비용을 부담해야 한다[7]. 따라서 전자 상점이 늘어남에 따라 상품 카탈로그를 구축하거나 관리하는 데에 드는 비용을 줄이기 위해서 여러 상점들이 공동으로 사용할 수 있으며 상품 정보의 변화를 잘 반영할 수 있는 공유된 전자 카탈로그 저장소의 필요성이 점차로 커지고 있다.

전자 카탈로그 저장소가 공유되는 시스템을 전자 카탈로그 라이브러리(DCL : Digital Catalog Library [7])라고 할 때 이 논문에서는 전자 카탈로그 라이브러리를 위한 소프트웨어 아키텍처를 제시하였으며, 이 아키텍처는 카탈로그 저장소의 상호 연동과 시스템의 신속성과 확장성이 고려되어 CORBA에 바탕을 두었으며 해당 시스템

템은 컴포넌트로 구성된다.

2. 시스템 아키텍처

전자 카탈로그 라이브러리에서 사용된 컴포넌트들은 카탈로그의 상호 연동성(interoperability)을 위해 특정 응용프로그램이나 플랫폼에 관계없이 상품 제조업자들 사이에서 정보를 주고 받을 수 있어야 한다. 다시 말해 카탈로그 자체의 이질성과 전자 상거래에 참여하고 있는 많은 업체들의 이질성과 상행위 자체의 정보 집약적인 성격을 고려할 때, 인터넷 환경에서의 상거래에서는 문서, 메시지, 파일 형식 등 각 상거래 주체들 사이의 정보를 표현, 교환할 수 있는 표준 형식이 필요하다.

따라서 본 논문에서는 CORBA(Common Object Request Broker Architecture)와 XML기반의 시스템을 제안한다. 시스템에서 사용되는 모든 컴포넌트는 IDL (Interface Definition Language)로 정의된 인터페이스를 가지고 있으며 XML형식의 카탈로그 정보를 이 인터페이스를 통해서 주고받게 된다. 또한 XML문서들은 개별 상품군에 대해 이미 정의되어 있는 DTD를 준수한다.

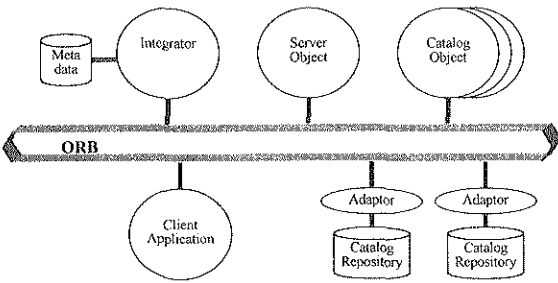


그림 1 시스템 아키텍처

그림 1에서 보는 바와 같이 제안된 시스템은 3-계층 클라이언트/서버(3-tier client/server) 시스템이다. 응용 프로그램(Client Application)과 서버 객체(Server Object) 및 인터그레이터(Integrator), 카탈로그 객체(Catalog Object), 카탈로그 저장소(Catalog Repository), 어댑터(Adapter) 등으로 구성된다. 미들웨어로 작동하는 ORB(Object Request Broker)를 이용하여 시스템을 구성하는 각 컴포넌트는 어떤 컴퓨터에 설치되어 있더라도 기능을 수행할 수 있다. 그리고 XML은 카탈로그 문서의 구조를 형식적으로 정의함으로써 컴포넌트 사이의 정보 공유와 재사용을 가능하게 한다.

구성 컴포넌트들

이 논문에서 제안하는 아키텍처는 Model/View/Controller (MVC) 패러다임[9]의 한 변형이다. MVC는 Smalltalk의 인터페이스를 만드는 데 사용하는 패턴으로서 대부분의 GUI(Graphic User Interface) 클래스 라이브러리에서 쓰이고 있다. MVC 패러다임에서 모델(Model)은 응용 프로그램과 이의 캡슐화된 데이터를 모델링 한 것이다. 뷰(View)는 데이터를 화면상에 보여주는 컴포넌트를 의미하고, 컨트롤러(Controller)는 유저 인터페이스가 사용자의 입력과 GUI의 이벤트에 어떻게 반응하는가를 정의한다. 이 논문에서 제시되는 아키텍처는 위의 세가지 객체 모델을 3계층 아키텍처의 각 계층에 적용하여 구성된다. 다만 여기서는 컨트롤러를 서버 객체, 카탈로그 객체, 인터그레이터의 세가지 컴포넌트로 분해하였다.

서버 객체는 시스템에서 서버의 작업 내용을 담고 있다. 이 객체는 상품 카탈로그나 상품 카탈로그의 정보 자체와는 별개로 순수하게 작업 수행을 위한 객체이다. 작업 분담(load balancing)이나 인증, 카탈로그 객체의 생성과 같은 일을 담당한다.

카탈로그 객체는 상품 정보를 표현하는데 쓰인다. 카탈로그에서 상품 정보는 몇 개의 범주로 분류되어 있으며, 카탈로그는 다른 객체가 접근할 수 있는 인터페이스를 제공한다. 카탈로그 객체가 가지고 있는 카테고리들은 사용자를 위한 뷰를 나타낸다. 카탈로그 객

체는 상품 정보를 직접 가지고 있지 않고, 대신 사용자의 요구에 알맞게 인터그레이터에서 현재 유효한 상품 정보를 동적으로 가져온다. 사용자들은 클라이언트 응용프로그램을 이용하여 카탈로그 객체를 조작한다. 시스템에 성공적으로 연결하면 응용 프로그램은 모든 상품 카탈로그와 연결되는 기본 카탈로그 객체를 얻게 된다. 좀더 상세한 뷰를 얻고 싶을 때는 카탈로그 객체에 뷰를 특화시키는 요청을 보내고, 이것은 다시 서버의 인증과정을 거쳐 인터그레이터를 통하여 새로운 뷰를 가진 카탈로그 객체가 생성된다.

카탈로그 저장소는 상품 정보를 직접 저장하는 저장공간으로써, 네트워크의 어느 곳이나 위치할 수 있다. 상품 정보로 저장 될 수 있는 것들은 카탈로그의 구조적인 정보, 일반 텍스트, 그림, 음성 정보 등 멀티미디어를 비롯한 모든 종류의 정보를 저장할 수 있다. 어댑터는 SQL과 같은 질의 언어를 이용하여 저장소의 카탈로그 정보를 사용할 수 있다. 어댑터는 인터그레이터가 저장소에 저장된 상품 정보에 접근 할 수 있도록 하는 표준 인터페이스를 제공한다. 어댑터와 인터그레이터는 CORBA IDL로 정의된 인터페이스를 가지고 있다. 어댑터가 일반적인 상품 정보 검색 요청에 대해서 진행되는 작업은 세가지 부분으로 나누어 볼 수 있는데, 첫번째는 자신이 담당하는 카탈로그 저장소가 가지고 있는 상품을 인터그레이터에 알린다. 두번째는 인터그레이터로부터의 검색 요청을 각 카탈로그 저장소에 적합한 언어로 바꾼다. 마지막으로 카탈로그 저장소로부터의 검색 결과를 시스템에서 정의한 규약(protocol)에 맞도록 변경하여 인터그레이터에게 보내 준다.

인터그레이터는 여러 개의 카탈로그 저장소에 저장되어 있는 전체 카탈로그의 집합에 대한 포괄적인 뷰를 제공한다. 이를 위해서 인터그레이터는 카탈로그 저장소의 어댑터를 이용하게 된다.

응용 프로그램은 디지털 카탈로그를 사용자에게 보여준다. 사용자들은 이 응용프로그램을 사용하여 각자가 원하는 상품을 찾거나, 새 상품을 등록하거나, 상품 카탈로그에 대한 수정 및 삭제할 수 있다. 응용프로그램 역시 CORBA를 기반으로 하여 서버 객체의 위치에 관계 없이 작동할 수 있다.

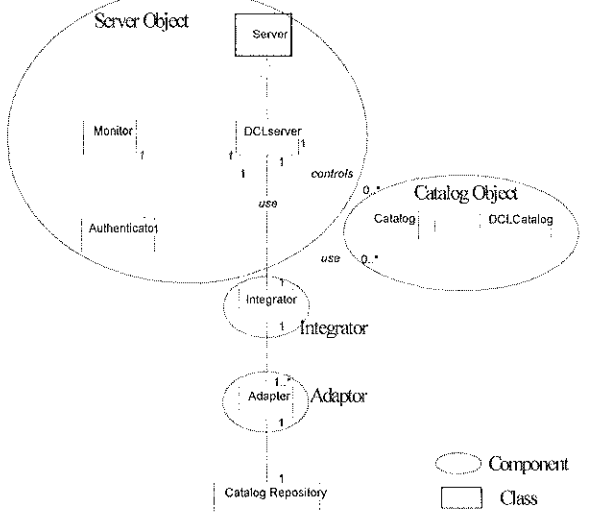


그림 2 시스템의 정적 구조

정적 구조

그림 2는 서버 객체, 카탈로그 객체, 인터그레이터, 어댑터 등 시스템을 구성하는 컴포넌트를 보다 더 세분하게 클래스 다이어그램으로 나타낸 것이다. 서버 객체는 DCL.server, Monitor, Authenticator 클래스를 포함한다. 서버 객체는 로그인이나 로그아웃과 같은 사용자를 위한 인터페이스를 가지고 있다.

DCLserver 클래스는 클라이언트/서버 아키텍처에서 공통적으로 쓰이는 기능들을 가지고 있는 Server 클래스에서 파생된 클래스로서 인증 관련 작업을 위한 Authenticator 클래스와 작업량을 관찰하기 위한 Monitor 클래스를 가지고 있다. 카탈로그 객체는 DCI.catalog 클래스의 객체이다. 카탈로그 객체는 사용자의 요구에 따라 생성되며 인터페이스로부터 동적으로 상품정보를 얻어낸다. 즉, 카탈로그 객체는 상품 정보를 물리적으로 저장하지 않으며, 사용자를 위해 상품정보에 접근하고 그 뷰를 보여줄 수 있는 인터페이스를 제공한다. 각 어댑터는 인터페이스와의 표준 인터페이스를 가지며, 인터페이스로부터의 질의 요청을 저장소에 전하는 역할을 한다.

응용 프로그램은 특화된 카탈로그의 참조값을 받아서 다시 위의 2번 작업부터 반복할 수 있다. 특화된 카탈로그 객체를 삭제하기 위해서 응용 프로그램은 DeleteView 메시지를 해당 카탈로그 객체를 생성한 카탈로그 객체에 요청하면 된다.

3. 맺음말

이 논문에서는 전자 카탈로그 라이브러리를 위한 컴포넌트 기반의 아키텍처를 시술했다. 전자 카탈로그 라이브러리를 이용하여 상품 제조업자는 상품에 대한 기본적인 정보를 가지고 있는 상품 카탈로그를 등록하고 인터넷 상점들은 상품 카탈로그를 받아와서 나름대로 특화된 상품 카탈로그를 만들 수 있다.

이 논문에서 제시된 아키텍처에서 주목할 부분은 신축성과 확장성이다. 이는 컴포넌트 기반의 CORBA 테크닉으로부터 가능하였다.

4. 참고문헌

- [1] Beat F. Schmid, Markus A. Lindemann, "Elements of a Reference Model for Electronic Markets," *Proceedings of the 31st Annual Hawaii International Conference on Systems Science HICCS'98*, Vol IV, Hawaii, January 6-9 1998.
- [2] Segev A., Wan D., Beam C., "Designing Electronic Catalogs for Business Value: Results the CommerceNet Pilot", CITM Technical White Paper, <http://www.haas.berkeley.edu/citm/research/>.
- [3] David-Michael Lincke, Beat F. Schmid, "Architecture and Business Potential of Mediating Electronic Product Catalogs," *Proceedings of the Association for Information Systems 1997 Americas Conference (AIS'97)*, Indianapolis, August 1997.
- [4] Arthur M. Keller, "Smart Catalogs and Virtual Catalogs", *Reading in Electronic Commerce*, Ravi Kalakota and Andrew Whinston. eds., Addison-Wesley, 1997.
- [5] Catalog interoperability study. CommerceNet research report 27-Feb-1998, http://www.commerce.net/research/pw/bulletin/98_05_r/98_05_r.pdf.
- [6] Daehong Communications Inc, Establishing and Sharing the standards of digital catalogs for electronic commerce. Technical Report, Dept. of CS, Seoul National Univ., 1999(<http://evgnus.snu.ac.kr/snude1/>).
- [7] Sang-goo Lee, Chisu Wu, Kapsoo Kim, Dongkyu Kim, Woochang Shin, "Digital Catalog Library: A Shared Repository of Online Catalogs for Electronic Commerce", WECWIS 98, 1998.
- [8] V.S. Alagar, K.Periyasamy, *Specification of Software Systems*. Springer, 1998.
- [9] Gamma, E. Helm, R., Johnson, R., and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, Mass., 1995.
- [10] Larman, Craig, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*, Prentice Hall, 1997.
- [11] Open Market, "LiveCommerce Technical Architecture", Open Market Technical White Paper. available via <http://www.openmarket.com>.

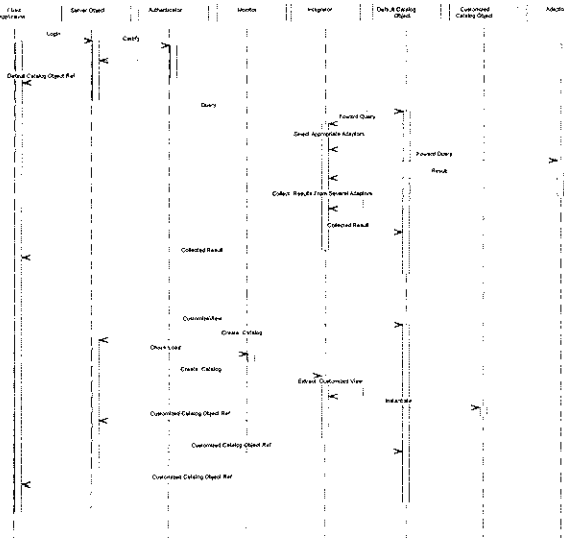


그림 3 시스템의 동적 구조

동적 구조

그림 3은 이 논문에서 제시된 시스템의 동적 구조를 sequence diagram을 이용해 설명해 주고 있으며 [10], 그 내용은 다음과 같다.

사용자가 시스템에 연결하면 서버 객체는 기본 카탈로그 객체를 찾아서 그 참조값(reference)을 응용 프로그램으로 전송한다. 사용자는 기본 카탈로그 객체의 분류 구조 속에서 각자가 원하는 정보를 찾는다.

사용자는 구매하고자 하는 상품의 정보를 얻고자 기본 카탈로그 객체에 질의를 보낼 수 있다. 기본 카탈로그 객체는 각 저장소에 연결된 어댑터와 인터페이스를 이용하여 사용자의 질의를 처리한다.

사용자는 기본 카탈로그 객체가 가지고 있는 전체 검색 범위를 줄일 수 있다. 사용자가 카탈로그 객체에 특화된 뷰를 요구하면 현재 카탈로그 객체에 의해 제공되는 뷰를 사용자의 요구에 맞게 재단한 뷰를 얻을 수 있다. 예를 들어 어떤 사용자는 자동차에, 또 다른 사용자는 컴퓨터에 관심이 있다고 할 때, 본 시스템은 각자 사용자의 관심 영역에 따라 사용자 별로 다른 정보를 쉽게 제공할 수 있다.

새로운 뷰에 대한 요구를 받으면 카탈로그 객체는 카탈로그 객체를 생성하는 역할을 맡는 서버 객체로 요구사항을 전송한다. 시스템 부하를 측정과 같은 부가적인 작업이 끝나면 서버 객체는 카탈로그를 생성하는 CreateCatalog 메시지를 인터페이스로 보내고, 인터페이스는 주어진 조건에 맞는 카탈로그 객체를 사용자가 가지고 있는 현재 카탈로그 객체로부터 만들어 내게 된다. 새롭게 만들어진 카탈로그 객체의 참조값은 사용자의 응용 프로그램에 전달된다.