

# RMM과 XML의 결합을 통한 웹 애플리케이션 설계<sup>†</sup>

## (An Approach to Web Application Design by Integrating RMM with XML)

최 재화\*, 류 수인\*\*  
(Jae-Hwa Choi Su-In Ryu)

**요 약** 웹 기술의 발전은 웹 사이트와 사용자의 급속한 증가를 가져 왔으며, 많은 기업은 인터넷을 통한 비즈니스 거래를 지원하는 웹 애플리케이션을 기본 응용시스템으로 개발하고 있다. 이러한 웹 애플리케이션을 개발하기 위해서는 체계적인 개발방법이 필요하지만 많은 기업들이 체계적으로 웹 애플리케이션을 설계하고 있지 못하다. 본 논문은 웹을 기반으로 하는 전자상거래시스템의 효과적이고 체계적인 개발을 위한 설계방법의 하나를 제시한다. 구체적으로 하이퍼미디어 애플리케이션 개발 모델인 RMM (Relationship Management Methodology)을 이용하여 개념적 모델을 디자인 하고, 그에 대한 XML(eXtensible Markup Language) Schema를 디자인 하는 보다 체계적인 웹 애플리케이션 설계방법을 제시한다

**Abstract** Advancement of Web technology has increased the number of web site and user population. Businesses develop Web applications as its core applications to support business transaction through Internet. Although a systematic development methodology should be used to develop Web applications, most businesses do not follow a systematic approach. This study shows a systematic development approach for Web-based electronic commerce information systems. Specifically, it shows how to design a conceptual model following a hypermedia application development model RMM(Relationship Management Methodology) and develop a logical database represented in an XML(eXtensible Markup

### 1. 서 론

웹 기술의 발전은 웹 사이트와 사용자의 급속한 증가를 가져 왔으며, 많은 기업은 인터넷을 통한 비즈니스 거래를 지원하는 웹 애플리케이션을 기본 응용시스템으로 개발하고 있다. 웹 애플리케이션(Web Application)은 사용자 인터페이스가 웹 페이지로 구성되어 정보를 디스플레이 하는 것 뿐만이 아니라 사용자와 대화를 통하여 비즈니스 프로세스를 처리한다. 이러한 웹 애플리케이션을 개발하기 위해서는 체계적인 개발방법이 필요하지만 많은 기업들이 체계적으로 웹 애플리케이션을 설계하고 있지 못하다.

웹 애플리케이션의 복잡성이 더 해감에 따라 사용되는 개발방법과 장기적 품질과 완전성에 대한 고려가 점차 강조되고 있다[6,7,8,9]. 그러나 많은 경우 웹 애플리케이션이 즉흥적인 방법으로 개발되고 그 후에는 연속적인 보완 작업으로 운영되고 있다[20]. 이것은 과도한 개발비용 또는 예산초과, 허술한 개발 프로젝트 관리, 개발 비용 및 자원 예측의 어려움과 같은 개발관련 문제를 낳는다. 지금까지의 웹 애플리케이션 개발방법에는 시스템적 접근방법, 품질 관리 및 보증이 결여 되어 있다. 또 웹 애플리케이션 개발 기술을 가진 인력도 부족하다.

본 논문은 웹을 기반으로 하는 전자상거래시스템의 효과적이고 체계적인 개발을 위한 설계방법의 하나를 제시한다. 웹 사이트 또는 웹 애플리케이션은 기존의 하이퍼미디어 또는 하이퍼텍스트 개발방법과 전통적 시스템 개발방법을 결합하여 개발될 수 있다. 전자상거래에서 데이터의 교환을 돕기 위해, 하이퍼미디어 애플리케이션 개발 모델

\* 단국대학교 경상대학, Corresponding Author

\*\*한국전산원

† 이 연구는 2000학년도 단국대학교 대학 연구비의 지원으로 연구되었음

인 RMM[17]을 이용하여 개념적 모델을 디자인 하고, 그에 대한 XML Schema를 디자인하여 기업간에 보다 원활히 데이터를 교환할 수 있는 바탕을 제시한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 연구의 동기를 현실적 필요성과 기존의 연구를 통하여 제시하고, 제 3장에서는 RMM과 XML스키마에 대하여 간략히 소개한다. 제4장에서는 RMM을 이용하여 주문처리 애플리케이션의 개념적 설계를 하고 XML스키마를 만들어 기업간에 원활히 데이터를 교환할 수 있는 웹애플리케이션의 설계과정을 제시한다. 제 5장에서는 본 논문의 결론과 앞으로의 연구과제에 대하여 논의한다.

## 2. 관련연구 및 연구동기

### 2.1 웹 애플리케이션 개발방법론 연구

일반적으로 웹 애플리케이션은 점차 복잡해지고 있다. 결과적으로 웹 애플리케이션의 개발도 점진적으로 복잡해지고 있다. 그럼에도 불구하고 웹 애플리케이션 개발에 대한 연구는 미미하다.

체계적인 하이 퍼 미디어 또는 하이퍼텍스트 애플리케이션의 디자인 방법은 Garzotto, Paolini 그리고 Schwabe [17]의 HDM(Hypermedia Design Model)에서 처음으로 등장한다. HDM은 하이퍼미디어 또는 하이퍼텍스트 개발방법으로 멀티미디어 뉴스 서비스 웹 사이트, 문학 및 예술 프로그램 웹 사이트, 키오스크 웹 사이트 개발을 위해 제시된 개발 방법이다. 이 개발방법은 주로 웹 사이트의 사용자 인터페이스 설계에 초점을 맞추고 있다. 이 방법은 각 웹 페이지의 시각적 설계를 강조할 뿐 웹 사이트 관리에 대한 구체적 방법을 제공하지 못하고 있다.

Isakowitz, Stohr, 그리고 Balasubramanian [18]은 RMDM(Relationship Management Data Model) 데이터 모델을 사용하여 하이퍼미디어 애플리케이션을 설계하고 구축하는 RMM을 발표했고, Schwabe와 Rossi [22]와 Schwabe [23]는 객체지향 프레임워크의 추상과 합성을 사용하여 복잡한 정보항목의 간결한 기술과 복잡한 내비게이션 패턴과 인터페이스 전환의 규정을 결정하는 하이퍼미디어 애플리케이션 개발방법인 Object-Oriented Hypermedia Design Model(OOHDM)을 소개하였다.

웹의 발전에 따라 프로젝트 관리자의 관점에서 웹 사이트 또는 웹 애플리케이션 개발방법과 도구의 제시에 관심을 보이기 시작했다. Araneus[6] 설계방법은 모든 웹 사이트가 복잡한 시스템이지만 적어도 출판될 콘텐츠(파일 시스템 또는 데이터베이스에 저장), 하이퍼텍스트 구조(페

이지와 접근경로), 그리고 프레젠테이션(페이지의 그래픽 레이아웃)의 세 가지 요소를 가지고 있다는데 기초를 두고 있다. Takahashi와 Liang[27]의 WBIS (Web-Based Information System)개발방법과 De Troyer와 Leune[11]의 WSDM(Web Site Design Method)은 웹 정보시스템 개발을 위해 제시된 방법론으로 웹 사이트 개발방법을 제시한다. 웹기반 정보시스템(WBIS) 개발방법은 정적 분석과 동적 분석으로 나누어 지며 정적 분석에서는 ER(Entity-Relationship)모델을 사용하고 동적 분석 및 설계에서는 시나리오 방법을 사용한다. 데이터 집중적인 웹 애플리케이션 개발방법은 정보 또는 데이터 유지,관리에 초점을 맞추고 있다.

기업간의 비즈니스를 지원하기 위한 웹 애플리케이션은 정보가 계속 교환되고 업데이트 되는 등 정보의 변동성이 높고, 구조도 복잡하다. 특히 계속적 주문처리와 같은 애플리케이션을 중심으로 하는 웹 기반의 전자상거래에서는 데이터의 교환이 계속적으로 이루어지므로 위의 하이퍼미디어 개발방법 중 정보의 변동성이 높은 웹 애플리케이션에서 그 유용성이 높은 RMM의 활용이 적합하다.

### 2.2 전자상거래를 위한 웹 애플리케이션 개발

최근 전자상거래시스템의 개발이 활발하다. XML 표준의 제시로 XML기반의 웹 애플리케이션이 본격적으로 개발되고 있다[10, 12, 13, 19, 25, 26, 28]. 웹을 통한 전자상거래에서는 다용량의 데이터를 처리하는 애플리케이션이 주요 요소이다. 이러한 웹 기술과 데이터베이스 기술의 통합을 요구하는 애플리케이션의 개발이 수행되고 있으나 효과적인 개발방법도 없이 즉흥적으로 개발되고 있는 편이다.

XML을 사용하는 웹 애플리케이션 개발에 대한 연구가 점차 등장하고 있다. Shohoud[25]는 XML 스키마 언어를 이용해 데이터의 유효성을 검증하고 구매자와 공급자, 비즈니스 파트너들과 데이터 포맷을 공유할 수 있는 예를 제시하였다. 김동우[1], 구태희[2]는 웹 기반의 XML/EDI시스템 및 도구를 설계·구현하였고, 기존의 EDI 사용자의 데이터베이스와 연동하기 위한 콤포넌트를 소개하였다. 스키마 변화에 대한 기술적 연구[26]와 일관성 유지와 스키마 적용에 관한 연구[29]도 진행되고 있다.

인터넷을 통한 비즈니스를 지원하기 위한 웹 애플리케이션은 XML표준과 데이터베이스시스템의 다용량 데이터 처리 기술의 결합을 위한 기술 발전을 요구하고 있다. 특히 이러한 웹 애플리케이션의 개발은 전통적 다용

1) ACM SIGMOD Record의 2001년 9월호는 고급 XML데이터 처리를 주제로 다루고 있다. IEEE Computer Society의 Data Engineering Bulletin 2001년 6월 호는 XML데이터 관리에 대한 특집을 싣고 있다.

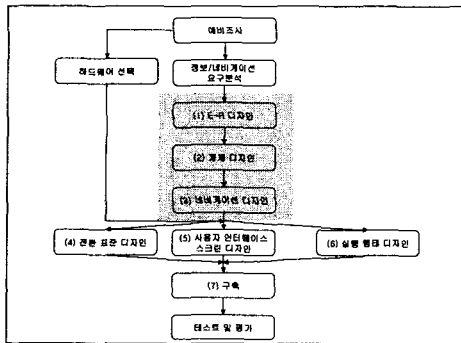
량 데이터베이스 애플리케이션 개발방법과 웹 사이트 개발 방법의 두 다른 개발방법 프레임워크(Framework)과 마인드셋(Mindset)의 통합을 요구한다. 웹 애플리케이션 개발을 위한 체계적 개발방법론이 필요하며 특히 전자상거래를 지원하는 웹 기반의 다용량 데이터베이스 애플리케이션의 경우 개념적 설계와 데이터베이스 설계가 중요하다.

### 3 RMM과 XML을 이용한 애플리케이션 설계

앞에서 언급한 것처럼 다른 영역에 대한 개발방법론을 살펴 보는 것에서 체계적인 웹 애플리케이션 개발방법을 찾아낼 수 있다. 물론 구체적 기법이나 방법론은 하이퍼미디어와 전통적 시스템 사이에는 큰 차이가 있을 수 있다. 그러나 각 개발방법론이 가지는 공통적인 요소가 있게 마련이므로 웹 애플리케이션 개발에 활용할 수 있는 것들에 대한 이해가 필요하다.

#### 3.1 RMM 설계

RMM 설계방법은 <그림 1>과 같이 소프트웨어 개발 사이클 전체를 보여준다[18]. 하이퍼미디어 개발에 직접적으로 관련되는 단계는 그림에서 보는 바와 같이 1) ER 디자인, 2) 개체 디자인, 3) 내비게이션 디자인, 4) 전환표준 디자인, 5) 사용자 인터페이스 스크린 디자인, 6) 실행행태 디자인, 7) 구축의 7단계가 된다.

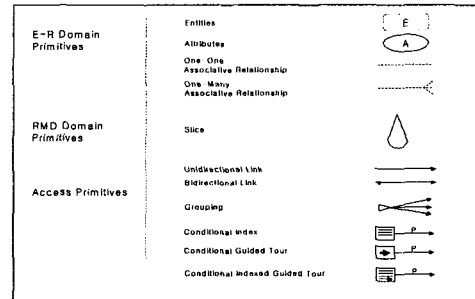


<그림 1> RMM 설계방법  
출처: Isakowitz, T., Stohr, E., & Balasubramanian, P. [18]

RMM은 RMDM이라 부르는 데이터 모델을 이용하여 하이퍼미디어 애플리케이션의 정보 객체와 내비게이션 메카니즘을 모델링한다. RMDM은 사용자 중심의 관점에서 사용자가 어떻게 하이퍼미디어 애플리케이션을 내비게

이트 할 것인가를 <그림 2>와 같이 ER 도메인 부(ER Domain Primitives), RMD 도메인 부(RMD Domain Primitives), 접근 부(Access Primitives)의 세 요소를 사용하여 RMDM 다이어그램으로 표현한다.

RMDM의 ER 도메인 부는 사용자 중심으로 분류된 정보를 어떻게 구성할 것인가를 모델링 하는데 사용되며, ER모델과 HDM에 기초하여 개체간의 1-1, 1-다, 그리고 다-다의 관계를 표현하는데 사용된다. 개체는 다른 특성을 가진 많은 속성을 가지므로 한 개체의 모든 속성을 한 스크린에 나타내는 것은 적절하거나 바람직하지 않다. 그러므로 속성은 슬라이스(Slice)로 나누어진다. RMD 도메인 부는 어떻게 정보가 표현되는가를 모델링 하는데 사용된다. RMDM에서 항해는 여섯 개의 접근 부를 사용하여 내비게이션 모델링을 한다. 단방향(Unidirectional link) 링크와 양방향(Bidirectional link) 링크는 한 개체의 슬라이스 사이의 접근을 표시한다. Grouping은 하이퍼미디어 문서의 다른 부분의 접근점을 제공한다. 예를 들어, 많은 애플리케이션의 첫 스크린은 다른 기능 또는 정보 부류를 접근하게 하는 버튼이나 메뉴를 포함한다.

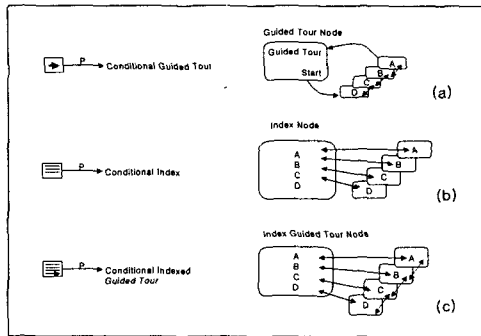


<그림 2> RMDM Primitives  
출처: Isakowitz, T., Stohr, E., & Balasubramanian, P. [18]

RMDM의 가장 중요한 접근 구조는 <그림 3>과 같이 Index, Guided tour, 그리고 Indexed guided tour이다.

<그림 3>에서 보듯이 Conditional Guided Tour는 개체 집단에서의 선형적인 통로로만 연결되어 있으며, 통로를 통해서만 항해의 시작과 끝이 가능하다. 즉 항목의 집합을 통해 선형적 접근을 가능하게 하여 사용자가 항목 간을 전방향 또는 후방향으로 움직일 수 있게 한다. Conditional Index는 인덱스 구조를 통해 접근하는 방법으로, 각각의 배당된 노드(Node)로부터 Index로의 연결이 가능하다. Conditional Indexed Guided Tour는 위의 두 방법을 그룹화 해 놓은 방법으로 다양한 접근 구조를 제공한

또 한국 정보처리학회의 정보처리 학회지 2001년 5월호에서도 XML 활용을 특집으로 내었다..



<그림 3> 세 가지 RMDM 향해 요소  
출처: Isakowitz, T., Stohr, E., & Balasubramanian, P. [18]

다. 이 세가지 접근 구조는 접근되는 항목을 지정하는 선택문 역할을 하는 논리적 조건을 활용한다.

### 3.2 XML 스키마 설계

웹 애플리케이션의 개념적 모델이 설계되면 이를 기반으로 데이터베이스 스키마를 개발할 수 있다. 스키마(Schema)란 정보의 구조를 설명하기 위한 모델을 말하며 XML에서는 document의 class를 위한 모델이다. XML 스키마 언어는 데이터베이스의 데이터 정의 언어(DDL; Data Definition Language)와 비슷하다. 특정한 XML 스키마는 데이터베이스의 특정한 테이블 정의와 비슷하다. 또한 XML 스키마 언어는 객체지향 언어의 클래스를 정의하는 언어와도 비슷하다. <그림 4>는 관계형 데이터베이스와 객체지향 언어, XML을 비교한 것이다.

XML의 Document Type Definitions(DTD) 또한 스키마로 볼 수 있다. 그러나 DTD는 하나의 문서에 대한 스

	Language used to define data type	Data type definitions	Data type instance
Relational database	DDL	Database table (Customers)	Table row (a specific customer)
OO Language	OO Language used to define classes	Class definition (CCustomer)	Object instantiated from class (myCustomer)
XML	XML Schema Language	An XML schema for customer	An XML document containing customer data

<그림 4> 관계형 데이터베이스, 객체지향언어, XML 비교  
출처: Shohoud, Y. [25]

키마로서는 아주 적절하다고 할 수 있지만, 문서에 담긴 데이터에 대한 스키마로서는 확장성이 없고, 데이터형을 다루는 기능이 부족하다. 기존의 XML 문서에 새로운 섹션을 추가하려면 그 DTD 전체를 다시 작성하고 새로 만들어진 DTD가 기존의 문서들과 제대로 일치하는 지 확인해야 한다. 그리고 DTD는 네임스페이스(Namespace)를 지원하지 않고, 하나의 DTD로부터 다른 DTD로의 상속의 개념이 없는 등 여러 제한점이 있다.

애플리케이션 개발자의 시각에서 볼 때 스키마란 하나의 XML 문서 안에 어떤 것들을 담을 수 있는지에 대한 규칙들의 집합이다[5]. 스키마가 있기 때문에 프로그래머들은 어떠한 애매함이나 혼란 없이 문서 안에 특정한 방식으로 배치된 특정 정보를 뽑아서 처리하는 소프트웨어를 만들 수 있는 것이다. 또 문서의 유효성 역시 스키마에 근거해서 판단하게 된다. 스키마에 근거해서 유효한 문서라면 그 스키마에 근거해서 만든 애플리케이션도 유효한 것이다.

## 4 RMM과 XML을 이용한 주문처리 웹 애플리케이션 설계

웹 애플리케이션을 체계적으로 설계하는 과정을 보이기 위해 RMM의 개발 사이클에 맞추어서 주문처리 웹 애플리케이션의 개념적 모델을 설계하기로 한다. 우선 주문처리 웹 애플리케이션의 정보 및 향해 요구분석을 하고, 분석된 정보를 바탕으로 ER모델을 설계하고, 개체, 향해를 순차적으로 설계한다. 향해 설계가 끝나면 데이터베이스 스키마를 작성하고 이를 XML 스키마로 전환한다.

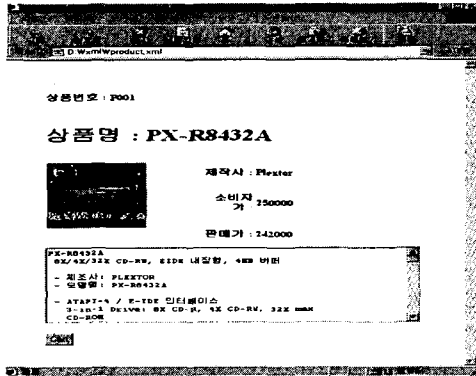
### 4.1 개념적 설계

#### 4.1.1 요구분석

본 논문에서 예시되고 있는 주문처리 웹 애플리케이션은 인터넷 상에서 컴퓨터 하드웨어, 소프트웨어를 판매하는 인터넷 쇼핑몰을 위한 전자상거래 시스템에서 기본적으로 필요로 하는 프로세스(Process)와 데이터의 흐름을 포함하고 있으며, 주요 구성요소로는 상품 디스플레이 화면, 장바구니(Cart), 주문접수 화면을 포함한다.

<그림 5>와 같이 상품 디스플레이 화면에 상품 그림을 포함한 상품 정보가 나타나게 되고, 사용자가 원하는 상품을 선택하여 장바구니에 담으면, 상품명, 가격 등에 대한 정보가 장바구니 화면에 전송되어 전송된 상품 정보와 수량, 금액을 디스플레이한다. 그러면 사용자는 필요에 따라 수정을 거쳐 주문을 하게 된다. 사용자가 주문을 하게

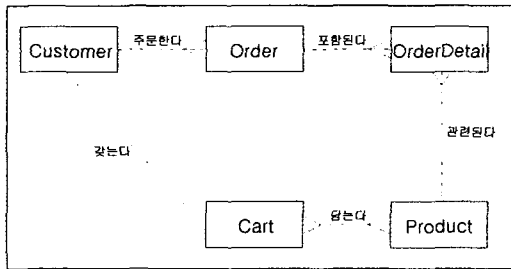
되면 주문자의 정보와 배달 및 결제에 관련된 정보를 입력 받게 된다.



<그림 5> 상품 디스플레이 화면

#### 4.1.2 ER 설계

위와 같이 선택한 상품을 장바구니에 담고 수정을 거쳐 주문하는 과정에 따른 ER 모델 설계를 보면 <그림 6>과 같다.

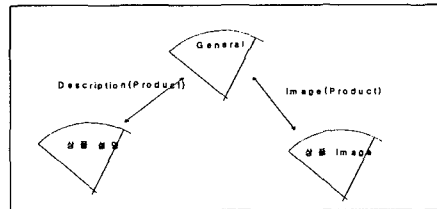


<그림 6> ER 모델

개체(Entity) 사이의 관계를 나타내는 관계성(Relationship)에서 "Customer"와 "Cart"의 관계를 나타내는 "갖는다" 관계성은 "1 대 1"이다. "Cart"와 "Product"의 관계를 나타내는 "담는다" 관계성은 "다 대 다"이다. "Cart"는 고객이 "Product" 데이터를 장바구니에 담아 놓으면 그 데이터를 저장하고 있다가 주문을 하게 되면 "Order"와 "OrderDetail"에 저장하고 "Cart"의 정보는 삭제하게 된다. 즉, "Cart"에는 정보를 임시로 저장하게 된다. "Order," "OrderDetail"과 "Product" 사이의 관계는 "1 대 다"의 "포함된다" 관계성과 "다 대 1"의 "관련된다" 관계성으로 "Order"와 "Product" 사이의 "다 대 다" 관계를 세분화하여 설계하였다.

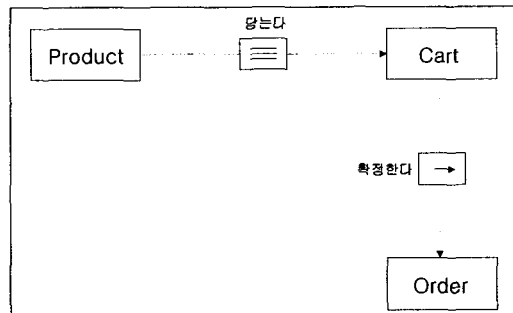
#### 4.1.3 개체 설계

이 단계에서는 선택된 개체가 어떻게 정보를 최종 사용자에게 보여주고, 어떻게 사용자가 접근할 수 있을 것인가를 정의하게 된다. 구체적으로 개체는 의미 있는 슬라이스로 나누어지고 이것은 하이퍼텍스트 네트워크로 조직되어 진다. 간단한 형태로는 한 개체의 모든 정보가 하나의 창에 보여지는 것이다. 그러나 개체는 많은 서로 다른 속성들로 구성되어 있기 때문에, 한 개체의 모든 속성을 한번에 모두 나타내는 것은 불가능하고, 바람직하지도 않다. 그래서 대안으로 정보를 서로 연관되어 있지만 나누어져 보여질 수 있는 의미 있는 단위로 나누어 슬라이스로 그룹화 할 수 있다.



<그림 7> Slice diagram for Product

<그림 7>은 상품개체의 정보를 세 슬라이스(일반적인 상품정보, 상품설명, 상품 Image)로 다시 구분한 개체 다이어그램이다. 각 슬라이스는 하나 이상의 속성을 가지게 된다. 그림의 "일반적 상품정보 (General)" 슬라이스는 상품명, 제작회사, 소비자가격, 판매가격 등의 속성을 가지게 되고, "상품 설명" 슬라이스는 상품의 상세 설명을, "상품 Image" 슬라이스는 상품 그림을 포함한다. 또한 개체 다이어그램에서 각 슬라이스는 단방향 또는 양방향으로 링크되어 있다. 예를 들어 "General"에서 "상품 Image"로는 "Image"로 링크되어 있고, 반대 방향으로는 "Product"로 링크되어 있다.



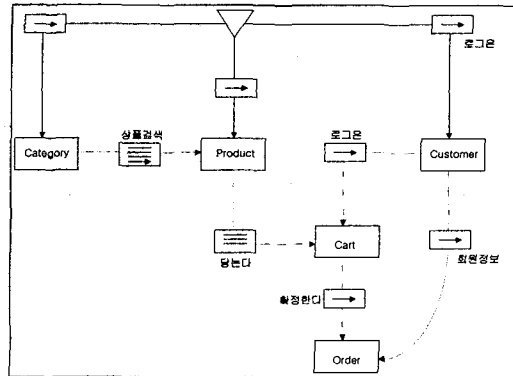
<그림 8> 주문과정 향해 다이어그램

#### 4.1.4 항해 설계

이 단계에서는 가능한 하이퍼텍스트 내비게이션을 디자인 한다. 각각의 관계성이 ER 다이어그램에서 분석되어져 나타났기 때문에 요구분석에 따라서 각 관계성에 대한 내비게이션을 연결시킬 수 있다. 각 관계성과 개체의 특성에 따라 명확하게 연결하기 위해서는 세가지 RMDM 항해요소가 필요하다. 이 세가지 요소는 앞에서 본 Conditional Indices, Conditional Guided Tours, Conditional Indexed Guided Tours이다. 세 가지 RMDM 항해요소를 기반으로 상품을 장바구니에 담고 주문하는 과정의 항해를 다이어그램으로 표시하면 <그림 8>과 같다.

개체간의 항해 설계는 관계성에 기반을 둔다. <그림 8>에서 "Product"와 "Cart"사이의 관계성인 "담는다"는 "Cart"에 있는 상품들이 "Product"에서 선택되어지는 것을 나타내기 위해 Conditional Index를 사용했다. "Cart"로부터 "Order"로의 관계성인 "확정한다"는 ER모델에 포함되지 않았으나 주문거래처리를 위해 필요한 항해 설계이며, Conditional Guided Tour를 사용했다.

주문처리 애플리케이션 전체의 항해 다이어그램은 <그림 9>와 같다.



<그림 9> 항해 다이어그램

위의 데이터베이스 스키마 중 상품화면의 XML Document "Product.xml"은 <그림 10>과 같다.

<그림 10>의 XML document는 하나 이상의 상품(Product)을 포함하고 있고 각 상품은 ProductNumber, ProductName, ProductImg, Company, UnitPrice, SellPrice, Description을 포함하고 있다. 위의 "Product.xml"에 대한 유효성 검증과 일관성을 지키기 위해서 XML 스키마를 정의할 수 있다. 즉, <그림 10>의 XML Document에 대한 XML 스키마는 <그림 11>과 같다.

## 4.2 스키마 전환

### 4.2.1 데이터베이스 스키마

주문처리 웹 애플리케이션에서 관리되는 정보는 상품정보, 고객이 상품을 장바구니에 담았을 경우 주문내용을 담고 있는 장바구니 정보, 그리고 주문했을 경우 배달 주소와 결제에 대한 주문정보로 이루어져 있다. 각각의 정보를 담고 있는 데이터베이스 테이블은 아래와 같다.

```
Product(ProductNumber, ProductName, ProductImg,
Company, UnitPrice, SellPrice, Description)
Cart(CartID, ProductName, SellPrice, Quantity,
TotalPrice, CustomerID)
CartProduct(CartID, ProductNumber, Date)
Order(OrderID, CustomerID, OrderDate, Bank, Memo)
OrderDetail(OrderID, ProductNumber, Quantity,
TotalPrice)
Customer(CustomerID, Passwd, CustomerName,
PublicID1, PublicID2, Sex, RegiDate, Region, City,
PostalCode, Phone, Cellular, eMail)
```

### 4.2.2 XML 스키마

```
<?xml version="1.0" encoding="EUC-KR"?>
<?xml:stylesheet type="text/xsl" href="product.xsl"?>
<products>
  <Product Number="P001"/>
  <ProductName>PX-R8432A</ProductName>
  <ProductImg src="d:\xml\plextor.jpg"></ProductImg>
  <Company>Plextor</Company>
  <UnitPrice currency="₩">250000</UnitPrice>
  <SellPrice currency="₩">242000</SellPrice>
  <Description>
    PX-R8432A
    8X/4X/32X CD-RW, EIDE 내장형, 4MB 버퍼

    - 제조사: PLEXTOR
    - 모델명: PX-R8432A

    - ATAPI-4 / E-IDE 인터페이스
    - 3-in-1 Drive: 8X CD-R, 4X CD-RW, 32X max
      CD-ROM
    .....
  </Description>
</products>
```

<그림 10> Product.xml

<그림 11>의 스키마는 XML 스키마 네임스페이스

(Namespace)를 디폴트(Default) 네임스페이스로 선언하는 것으로 시작한다. 여러 다른 XML 애플리케이션에서 태그를 혼합하고 대조할 때 두 개의 다른 대상을 지칭하는 데 사용되는 동일한 태그가 있을 수 있다. 네임스페이스는 URI를 각각의 태그셋(Tag Set)에 연결시키고, 각 엘리먼트(Element)가 어느 태그셋에 속하는지를 나타내 주는 접두어를 붙여서 이런 애매한 점을 명확하게 한다.

기본적인 XML 스키마는 크게 두 가지 요소로 나눌 수 있다. 첫째는 선언 요소(Declaration Component)이고 두번째는 타입 정의 요소(Type Definition Component)이다. 선언 부분은 다시 엘리먼트 선언과 속성 선언으로 구분되어진다. 위의 예에서 “<element name=“Products” type=“ProductType”>”는 “ProductType”이라는 타입을 갖는 “Products”라는 엘리먼트가 있다는 것을 선언하는 부분이고 “<attribute name=“currency” type=“string”/>”은 “currency”라는 속성을 선언하는 부분이다.

```

<schema xmlns="http://www.w3.org/1999/XMLSchema">
  <element name="Products" type="ProductType">
    <complexType name="ProductType">
      <complexType name="ProductNumberType">
        <attribute name="Number" type="string"/>
      </complexType>
      <element name="ProductName" type="string"/>
      <element name="ProductImg" type="string"/>
      <element name="Company" type="string"/>
      <element name="UnitPrice" type="PriceType">
        <complexType name="PriceType"
          base="nonNegativeInteger" derivedBy="extension">
          <attribute name="currency" type="string"/>
        </complexType>
      </element>
      <element name="SellPrice" type="PriceType"/>
      <element name="Description" type="string"/>
    </complexType>
  </element>
</unique>
  <selector>Products/Product</selector>
  <field>@Number</field>
</unique>
</schema>

```

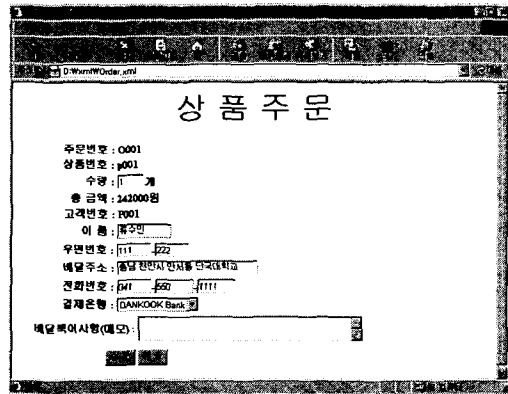
<그림 11> ProductSchema.xml

XML 스키마는 선언된 엘리먼트에 대해서 자신만의 타입을 정의할 수 있는데, 위의 예에서 타입이 정의된 부분을 보면 “<complexType name=“ProductType”>”은 선언된 엘리먼트에서 사용되는 새로운 타입을 만들어 주는 부분이다. 타입은 단순타입(SimpleType)과 복합타입(Complex Type)으로 구분된다. SimpleType은 속성을 포함하지 않고 텍스트만 포함하는 엘리먼트이고 ComplexType은 모든 엘리먼트에 사용된다. 즉, 속성이나 서브 엘리먼트(Subelement)를 포함하는 엘리먼트는 ComplexType이고, 속성은 다른 엘리먼트를 포함할 수 없기 때문에 SimpleType이다.

“Product.xml”문서의 기본키(Primary Key)는

“Product” 엘리먼트로 “Number=P001”이라는 속성을 갖는다. “Product”엘리먼트를 기본키로 지정해 주는 방법은 “key”또는 “keyref” 엘리먼트를 사용하는 방법도 있지만, 여기서는 “Unique” 엘리먼트를 이용하였다. 이 경우, “selector” 엘리먼트는 “XPath”의 표현이 포함되어 있다. “Products/Product”는 위의 예에서 모든 “Product” 엘리먼트 리스트를 만들고 “field” 엘리먼트의 “@Number” 속성은 엘리먼트의 “Number” 속성이 유일해야 한다는 것을 표현한다. 제품가격 엘리먼트(UnitPrice)의 데이터 타입인 “PriceType”은 “nonNegativeInteger”타입을 기본으로 확장된 타입으로 “currency” 속성을 갖는다. “nonNegativeInteger” 타입은 “0”을 포함하고 “PositiveInteger”는 “0”을 포함하지 않는 차이점이 있다.

선택된 상품을 장바구니에 담아서 주문을 하게 되면 <그림 12>와 같은 주문화면이 뜨고 주문정보를 입력하게 된다.



<그림 12> 주문화면

<그림12>를 위한 주문정보에 대한 문서 “Order.xml”은 상품과 상품수량, 총 금액과 고객정보, 배송정보, 결제은행, 배달 특이사항 등을 포함한다. “Order.xml”에 대한 스키마는 <그림 13>과 같다.

XML Schema 언어에서는 네임스페이스를 지원함으로써 다른 XML문서에서 사용하였던 데이터 타입 등을 다시 정의 하지 않고 스키마를 import함으로써 그대로 사용할 수 있다. 위의 예에서 보면 “Product” 문서의 스키마를 import 함으로써 “TotalPrice” 엘리먼트와 “Product” 엘리먼트의 데이터 타입을 다시 정의하지 않고 “Product” 문서의 데이터 타입을 이용할 수 있다.

또한 수량을 나타내는 “Quantity” 엘리먼트의 속성인 “positiveInteger”는 1이상의 수를 말한다. XML 스키마 언어는 일련의facet을 제공한다. facet은 길이, 정확성 (precision), scale, 패턴을 포함한다. 예로, 위의 스키마가

가지고 있는 “ZipType”은 우편번호인 “OrderPostalCode” 엘리먼트의 데이터 타입으로 facet의 패턴을 사용하여서 우편번호가 세개의 숫자, 대시 기호, 세개의 숫자로 이루어졌음을 말해준다. 즉 333-333과 같이 되었음을 말한다. “OrderPhone” 엘리먼트의 데이터 타입인 “PhoneType”도 마찬가지이다. 은행명인 “Bank” 엘리먼트의 데이터 타입인 “BankType”은 “enumeration” 속성을 이용하여 계좌를 가지고 있는 은행 이름의 목록을 보여준다.

XML 스키마 언어를 사용한 위의 예에서는 “key” 엘리먼트를 사용해서 “Product” 엘리먼트를 외래키로 설정해 줌으로써 관계성을 표현하였다.

```

<schema targetNamespace="file:///d:/xml"
  xmlns="http://www.w3.org/1999/XMLSchema"
  xmlns:Pro="file:///d:/xml/ProductSchema.xml"
  xmlns:Cus="file:///d:/xml/CustomerSchema.xml">
  <import namespace="file:///d:/xml/ProductSchema.xml"/>
  <import namespace="file:///d:/xml/CustomerSchema.xml"/>
  <element name="Orders" type="OrdersType"/>
  <complexType name="OrdersType">
    <element name="Order" type="OrderType" maxOccurs="*" />
  </complexType>
  <complexType name="OrderType">
    <element name="Product"
      type="Pro:ProductNumberType"/>
    <element name="Quantity" type="positiveInteger"/>
    <element name="TotalPrice" type="Pro:PriceType"/>
    <element name="Customer" type="Cus:CustomerType"/>
    <element name="OrderName" type="string"/>
    <element name="OrderPostalCode" type="ZipType"/>
    <element name="OrderAddress" type="string"/>
    <element name="OrderPhone" type="PhoneType"/>
    <element name="Bank" type="BankType"/>
    <element name="Memo" type="string"/>
    <attribute name="ID" type="string"/>
  </complexType>
  </element>
  <unique>
    <selector>Orders/Order</selector>
    <field>@ID</field>
  </unique>
  <key name="foreignKey">
    <selector>Orders/Product</selector>
    <field>@number</field>
  </key>
  <simpleType name="ZipType" base="string">
    <pattern value="/d(3)-d(3)"/>
  </simpleType>
  <simpleType name="PhoneType" base="string">
    <pattern value="/d(3)-d(3)-d(4)"/>
  </simpleType>
  <simpleType name="BankType" base="string">
    <enumeration value="DANKOOK Bank"/>
    <enumeration value="DAEILIAN Bank"/>
  </simpleType>
</schema>

```

<그림 13> OrderSchema.xml

## 5 결 론

목표한 웹 애플리케이션을 개발하기 위해서는 체계

적인 개발방법이 필요하지만 많은 기업들이 체계적인 개발 방법을 찾지 못하고 시행착오를 겪고 있다. 본 논문에서는 웹 애플리케이션을 설계하기 위한 방법으로 하이퍼미디어 개발방법론인 RMM을 이용해 주문처리 애플리케이션의 개념적인 모델을 설계하여 웹을 기반으로 하는 전자상거래 시스템의 체계적인 개발방법을 제시하였으며, 설계된 개념적 모델을 기반으로 데이터베이스 스키마를 XML을 중심으로 한 스키마를 설계하는 방법을 제시하였다.

웹 애플리케이션 개발이 체계적이고 효과적이기 위해서는 효과적인 방법론의 개발은 물론 개발과정을 돕는 도구의 개발도 필요하다. 또 XML스키마는 정적이 아니라 변화하는 애플리케이션의 요구사항에 의해 진화한다. 스키마 변화에 대한 기술적 연구와 일관성 유지와 스키마 적용에 관한 연구가 필요하다.

## 참 고 문 헌

- [1] 김동우, 고규준, 박정선, “XML/EDI 도구 프로토타입개발에 관한연구,” 한국전자상거래학회지, 제5권, 제1호, 2000.6
- [2] 구태희, 변광준, 황인준, “웹 기반 XML/EDI 시스템 구현,” 한국전자상거래학회지, 제6권, 제1호, 2001.4
- [3] 이강찬, 이규철, “XML기반의 인터넷 정보 중재자의 설계 및 구현, 한국전자상거래학회지, 제6권, 제2호, 2001. 8
- [4] 지식진, “자바기반의 XML 스키마 편집기의 설계 및 구현,” 한국정보처리학회 추계 학술발표논문집, 제 7권 2호, pp-55-58, 2000. 10
- [5] 황병연, 김연혜, “XML 스키마 발전 동향”, 한국정보처리학회지, 제8권, 제3호, 2001. 5
- [6] P. Atzeni, “Web Sites Need Models and Schemes,” in Conceptual modeling - ER '98 : Proceedings of 17th International Conference on Conceptual Modeling, (Ed. Tok Wang Ling , Sudha Ram , Mong Li Lee. ), November 1998
- [7] V. Balasubramanian, B. M. Ma, and J. Yoo, “A Systematic Approach to Designing a WWW Application,” Communication of the ACM, 58(8):



47-48, August 1995

[8] S. Ceri, P. Fraternali, S. Paraboschi, "Design Principles for Data-Intensive Web Sites," SIGMOD Record, 28(1), March 1999

[9] Diaz, A., Isakowitz, T., Maiorana, V., & Gilabert, G., "RMC: A Tool To Design WWW Applications"

[10] Chawathe, S., "Describing and Manipulating XML Data", IEEE Computer Society Technical Committee on Data Engineering, 1999

[11] O. M. F. De Troyer, C. J. Leune, "WSDM: A User-Centered Design Method for Web Sites," Proceedings of WWW7 Conference, Brisbane, April 1997

[12] Eric van der Vlist, "Using W3C XML Schema," October 17, 2001, <http://www.xml.com/pub/a/2000/11/29/schemas/part1.html>

[13] Fallside, D., "XML Schema Part 0: Primer", <http://www.w3.org/TR/xmlschema-0/>, 2001

[14] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. "Strudel: A Web-site Management System," Proceedings of ACM SIGMOD, May 1997

[15] Fitzgerald, B., "Systems Development Methodologies: Time to Advance Clock," Systems Development Methods for the Next Century, ed. Wojtkowski et al., Plenum Press, New York, 1997

[16] P. Fraternali and P. Paolini, "A Conceptual Model and a Tool Environment for Developing More Scalable and Dynamic Web Applications," Proceedings of EDBT, March 1998

[17] F. Garzotto, P. Paolini and D. Schwabe, "HDM - a model-based approach to hypertext application design," ACM Transactions on

Information Systems, 11(1): 1 - 26, 1993

[18] T. Isakowitz, E. A. Stohr, and P. Balasubramanian, "RMM: A methodology for structured hypermedia design," Communication of the ACM, 58(8), August 1995

[19] Kotok, A., "XML and EDI Lessons Learned and Baggage to Leave Behind", <http://www.xml.com/pub/a/1999/08/edi/index.html>

[20] Manola, F., "Towards a Web Object Model", <http://www.objs.com/OSA/wom.htm>, Object Service and Consulting, Inc., feb 1998

[21] Murugesan, S., "Web Engineering," ACM SIGWEB Newsletter, Vol.8, No.3, October 1999

[22] D. Schwabe and G. Rossi, "The Object-Oriented Hypermedia Design Model," Communication of the ACM, 58(8), August 1995

[23] Schwabe, D., Rita de Almedia Pontes, & Moura, I., "OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the WWW", ACM SIGWEB newsletter 8(2), June 1999

[24] Shanmugasundaram, J., Tufte, K., He, G., Zhang, C., DeWitt, D., & Naughton, J., "Relational Databases for Querying XML Document: Limitations and Opportunities", VLDB Conference, 25, Edinburgh, Scotland, 1999

[25] Shohoud, Y., "XML's Grand Schema", XML Magazine, Summer 2000, <http://www.devx.com/premier/mgz/narch/xml/2000/03sum00/ys0300/ys0300.asp>

[26] Su, H., Kramer, D., Chen, L., Claypool, K., & Rundensteiner, E., "XEM: Managing the Evolution of XML Documents," RIDE-DM 2001; 103-110, 2001

[27] K. Takahashi, E. Liang, "Analysis and Design of Web-based Information Systems," Proceedings of WWW7 Conference, Brisbane,

April 1997

[28] Thompson, H., Beech, D., Maloney, M., & Mendelsohn, N., "XML Schema Part 1: Structures", <http://www.w3.org/TR/xmlschema-1/>, 2001

[29] Wedemeijer, L., "Engineering for Conceptual Schema Flexibility," RIDE-DM 2001; 85-92, 2001



**최재화** (Choi, Jae Hwa)

1980.2 연세대학교 상경대학  
경제학 학사(BA)

1984.6 미국 Drexel  
University 경영대학  
경영학 석사(MBA)

1991.8 미국 University of Maryland

경영대학

정보시스템학 박사 (Ph.D)

1995.3 현재 단국대학교 경상대학 경영학부  
부교수

관심분야: 데이터베이스시스템, 전자상거래



**류수인** (Ryu Su In)

1999.2 단국대학교 경영학 학사

2001.2 단국대학교 대학원 경영  
정보학 석사

2001.3 - 현재 한국전산원 전자  
거래연구부

관심분야: 전자상거래