

분산 객체 지향 환경에서 UML을 이용한 Web 애플리케이션 개발에 관한 연구

박장한*, 김영선*, 장덕철*

*광운대학교 컴퓨터과학과

e-mail:parkch@explore.kwangwoon.ac.kr

A Study on Web Application Development based on UML in Distributed Object-Oriented Environment

Chang-Han Park*, Young-Sun Kim*, Duck-Chul Jang*

*Dept of Computer Science, KwangWoon University

요약

분산 객체 지향 환경에서 클라이언트/서버 시스템 구성을 위한 미들웨어는 복잡한 클라이언트/서버간의 통신을 단순화하기 위한 것으로부터 시작되어 최근에는 많은 사용자들의 다양한 요구를 해결하기 위해 보다 기능을 확장하고 있다. 이때 발생할 수 있는 문제점으로 네트워크 트래픽의 증가, 특정 웹 서버로의 집중에 따른 부하와 관리 부담의 증가 및 전체 시스템의 서비스 성능 저하를 초래할 수 있다. 이에 대한 해결 방안으로 본 논문에서는 객체 지향 분석 및 설계 방법론인 UML(Unified Modeling Language)을 이용하여 객체 지향 모델링 기법으로 Web상에서 관련된 여러 서버들간에 분산 저장된 정보를 일반 사용자에게 동적으로 빠른 서비스를 제공할 수 있고, 특정 서버에서 발생할 수 있는 오버헤드를 줄이며, 네트워크 상에서 발생할 수 있는 병목현상을 현저히 줄일 수 있도록 RDS(Remote Data Service)를 이용하는 Web 애플리케이션 설계 및 구현을 제안하였다.

1. 서론

최근 인터넷의 빠른 보급으로 인해 웹 상에서 정보의 제공자와 사용자가 급증하고 있다. 웹은 편리한 사용자 인터페이스와 멀티미디어 환경을 제공함으로써 점차 다양하고, 방대한 양의 정보와 사용자 중심의 동적인 서비스를 제공하고 있다. 이러한 변화에 유연하게 대처할 수 있기 위한 방안으로 웹과 클라이언트/서버 애플리케이션을 하나로 통합할 수 있는 미들웨어를 이용한 3-tier를 들 수 있다. 이러한 시스템의 장점은 모든 자원이나 제어 기능을 각 노드에 복제(replication), 또는 분산(distribution) 시킴으로써 전체 시스템의 성능 향상과 유용성(usability), 확장성(extensibility), 신뢰도(reliability) 등과 같은 면에서 중앙 집중 처리 시스템보다 장점을 얻게 된다[10,13]. 그러나 분산 처리의 성능은 다음과 같은 단점이 있다[2,16].

첫째, 분산된 중복 데이터에 대해 많은 수의 병행 갱신을 요청하는 작업을 처리시 오버헤드가 발생한다.

둘째, 분산 처리 시스템 설계 및 구현은 복잡도가 증가한다. 따라서 시스템 개발비용과 유지 보수비용이 증가한다.

셋째, 분산 데이터베이스 처리는 제어가 어렵다. 그러므로 장애가 발생하면 회복 수행이 훨씬 더 어려워진다.

이와 같은 문제점을 해결하기 위해 본 논문에서는 객체 지향 분석 및 설계 방법론인 UML을 이용하여 객체 지향

모델링 기법으로 RDS를 이용한 데이터 할당(data allocation) 방법을 논하고자 한다. 할당 방법은 데이터 분할 후 웹 상에서 파일에 대해 질의, 갱신 및 실행을 위해서 서로 다른 노드들에게 파일을 할당하는 것을 말한다. 이러한 방법으로는 데이터 복제(data replication) 방법과 데이터 이주(data migration) 방법이 있다[10,13]. 복제 방법은 모든 지역에 동일한 전체 데이터베이스를 유지하는 완전 복제(fully replicated)방법과 데이터베이스 일부분은 분할된 상태로 존재하고, 일부분은 여러 지역에 부분 중복 존재하는 부분 복제(partially replicated)방법이 있다. 완전 복제 방법은 모든 검색 요구를 지역적으로 처리할 수 있으므로 검색 비용이 매우 낮아진다. 그러나 동일한 데이터가 여러 지역에 존재하기 때문에 갱신 비용이 높을 뿐 아니라 각 데이터들의 일관성(consistency) 유지를 위한 방법이 아주 복잡해진다[5]. 이주 방법은 많이 참조하는 지역과 컴퓨터 중앙 처리장치의 오버헤드를 고려하여 해당 지역에 데이터를 이주시키는 방법으로 중복을 허용하는 복제 방법에 비해 기억 장소의 낭비가 없고, 여러 복제본 사이의 일관성 유지를 위한 오버헤드도 없다. 그러나 전체 파일 이주가 아닌 수직 분할(Vertical Fragmentation : VF) 및 수평 분할(Horizontal Fragmentation : HF) 이주시 많은 트랜잭션이 일어날 때 발생하는 통신비용과 최소의 응답 시간을

보장 받을 수 없다.

본 논문에서는 파일 복제와 이주시 관련된 테이블의 항목들을 RDS를 이용하여 클라이언트 컴퓨터에 레코드셋(RecordSet) 형태로 질의 결과를 전송한다. 이 방법은 원하는 데이터를 인터넷 익스플로러 안의 액티브X 데이터 컨트롤인 DBGrid 등을 위치시켜 놓은 추가, 삭제, 수정과 같은 일을 수행해 낸다. 즉 웹브라우저가 지원하는 스크립트 언어로 처리한 후 다시 웹서버를 거쳐 DB서버에 전송할 수 있다. 결국 디스크 I/O 시간과 지역간 통신비용을 줄여 데이터의 병행 속도를 빠르게 하는데 그 목적이 있다. 따라서 웹 환경에서 사용자로부터 최소로 응답시간을 보장받을 수 있도록 분산 트랜잭션 파일들을 어떻게 효율적으로 처리할 수 있는가에 대한 것으로 연구 범위는 아래와 같다.

첫째, 웹 데이터베이스를 UML을 이용하여 객체 지향 모델링 기법으로 설계 방법을 제시하며,

둘째, 설계 방법을 토대로 웹 환경에서 RDS 이용한 효율적인 분산 트랜잭션 처리방법을 제시하며,

셋째, 실 시스템을 적용 성능 평가 분석을 하여 우수성을 입증하고자 한다.

최근 CGI는 저 비용, 편리한 사용, 확장성, 폭넓은 사용자 기반 등의 장점을 갖는 웹에서 서비스 확장을 효과적으로 지원해 주기 때문에 많은 각광을 받고 있지만, CGI를 통한 서비스들은 웹서버의 많은 자원들을 필요로 하기 때문에 서비스를 요청하는 클라이언트의 수가 증가할수록 프로세스의 크기 및 관리 부담의 증가로 전체 시스템 성능의 저하를 초래할 수가 있다. 이런 단점을 보완하기 위해 본 논문에서는 UML을 이용하여 객체 지향 모델링 기법으로 설계하고, MS사의 RDS를 이용하여 웹 서버의 관리 부담을 줄이고, 부하 집중을 막을 수 있도록 클라이언트 계층에서 데이터를 동적으로 처리할 수 있는 웹 데이터베이스 관리 시스템을 설계를 제안하고 구현하고자 한다.

2. 관련 연구

2.1 UML(Unified Modeling Language)

객체 지향 시스템에서 개발되는 부품들의 명세, 문서화 및 시각화를 시키기 위하여 사용되는 하나의 모델링 언어가 UML이다. 이것은 Booch방법론, Rumbaugh의 OMT, Jacobson의 OOSE 등을 통합하여 만든 객체 지향 분석 및 설계 방법론으로 표준 지정을 목표로 제안되었다[1,2]. 따라서 모델링 단계에서 생성되는 각종 요소를 적절한 표기법과 절차로 도식화한다면 개발자는 시스템의 전체적인 모습을 정확하게 이해할 수 있는 통찰력을 얻을 수 있다. 또한 모델링을 통해 애플리케이션 개발자와 사용자간의 의사소통을 위한 연결 고리가 되기도 하며, 다이어그램 단위별로 모델을 기술할 때에 그 표기법의 이해를 높여 주고, 정

확한 표현이 되도록 도움을 준다. 본 논문에서는 UML을 이용하여 객체 지향 모델링 기법으로 설계하고자 한다.

2.2 클라이언트/서버 모델

대부분의 클라이언트/서버 시스템들은 분할된 작은 시스템들로 구성된 2-tier 구조로 구성되어 있다. 이는 개발 생산성이나 비용 면에서 3-tier보다 다소 유리하고, 운영시 시스템 확장성을 고려하지 않아도 된다. 그러나 웹에서 데이터베이스 관리 시스템을 구축하려면, 중간 계층에 웹 서버 애플리케이션 서버가 존재하게 됨으로 자연스럽게 3-tier 구조를 형성하게 된다.

과거의 클라이언트/서버 모델에서 RPC(Remote Procedure Call)를 기반으로 차세대 웹 환경을 지원하게 될 분산 객체 기반의 모델로는 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture)와 MicroSoft사의 DCOM(Distributed Component Object Model)이 있으며 이들은 분산 환경을 위한 통합 개발 환경에 적합하다.

2.3 원격 데이터 서비스(RDS)

OLE 데이터베이스를 기반으로 한 ADO(ActiveX Data Object)는 데이터베이스 관련 인터페이스를 제공하는 ActiveX 컨트롤로서 인터넷이나 클라이언트/서버 환경에서 동적인 웹 데이터베이스의 구축이 가능하다. 그러나 ADO는 모든 데이터에 대한 처리가 서버에서만 가능하지만, RDS(Remote Data Service)는 각종 데이터를 클라이언트 내에서 스크립트 처리를 할 수 있다.

또한 RDS는 원격 데이터베이스 서버에 저장된 데이터를 ActiveX 데이터 컨트롤에 바인딩시킬 수 있고, 대량 질의 결과를 이진형태로 변환해 주며, 클라이언트에 데이터를 저장할 수 있어 서버와의 불필요한 접속을 피할 수는 장점을 가지고 있다.

특히 애플리케이션이 비즈니스 로직을 객체에 결합할 수 있도록 도와주는데, 이런 middle-tier객체들이 클라이언트와 응용프로그램으로부터 받은 요구를 ODBC를 통해 MS-SQL서버와 같은 데이터베이스 컴포넌트와의 통신을 가능하게 한다.

2.4 웹과 데이터베이스 연동

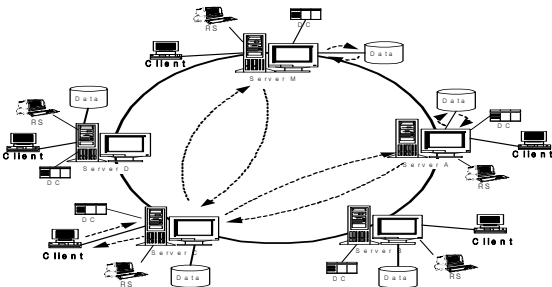
웹 기반의 CGI는 데이터베이스와 연계되어 클라이언트의 질의를 표준 SQL로 해석하여 웹 서버 및 데이터베이스 서버와 상호 연동시킬 수 있다. 그러나 메시지 전송시 CGI 프로그램과 웹 서버간의 오버헤드가 발생하고, 데이터베이스 처리는 기본적인 검색 질의만 가능하며, 임의의 트랜잭션 발생시 데이터의 일관성을 갖지 못하는 단점을 가지고 있다.

반면에 ASP(Active Server Page)는 서버에 있는 객체를 사용하여 객체의 인스턴스를 생성할 수 있고, 생성된 객체의 메소드와 이벤트를 직접 사용할 수 있게 된다. 따라서 일반적으로 스크립트 언어가 갖는 한계를 극복할 수 있으며, 동적인 웹 페이지를 만들기 위해 PWS(Personal Web Server), IIS(Internet Information Server)로부터 제공되는 확장 엔진으로, 스크립트 형태로 서버에서 실행하게 되며, 스크립트 안에서 ActiveX 서버 컴포넌트를 호출하여 실행할 수 있다.

3. 효율적인 Web 애플리케이션 설계 및 UML 적용

3.1 분산 네트워크 구성

본 논문에서는 분산 환경에서 주 서버에 발생하는 부하 집중을 줄이고, 클라이언트에서의 필요로 하는 동적인 서비스 요청을 효과적으로 대응하기 위해 “학점 은행제”에 관한 데이터베이스 관리 시스템을 <그림1>과 같이 구성하고자 한다.



<그림1> 분산 네트워크 구성도

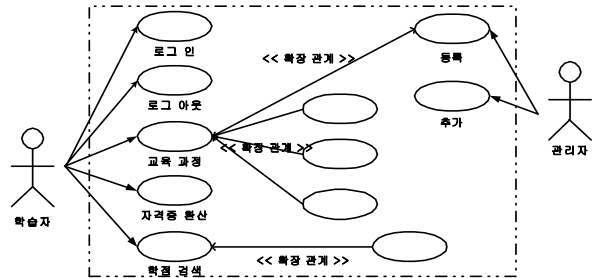
<그림1>에서와 같이 사용자가 서버C를 통해 웹 서버M에 접속하여 질의한 내용에 대한 인덱스만을 받아 관련된 서비스를 분산된 각 서버들을 통해 받아들이고, 이를 동적으로 재구성하여 사용자에게 제공할 수 있도록 한다. 또한 라우팅 서버(Routing Server)를 두어 메시지를 잠시 보관하고 목적지까지의 경로를 알려주는 메시지 전달만을 책임지는 서버와 임시 테이블을 저장하고 있는 비독립적 클라이언트(Dependent Client)를 둔다. 이는 모든 서비스를 서버M을 통해서만 서비스하는 방법에 비해 주 웹서버의 부하 집중을 줄이고, 네트워크에서의 병목 현상을 현저히 줄일 수 있다.

3.2 객체 지향 모델링

사용자 중심의 요구사항을 모델링하기 위하여 UML 기반 요구사항 등 다음과 같이 제안한다. 우선 첫번째 단계로 UML의 의미와 다이어그램 표기법을 사용하고, 두 번째 단계에서는 요구사항의 분석을 통하여 검토와 토론을 통해 모델을 정제시키고, 마지막으로 최종 요구사항을 결정한다.

3.2.1 Use Case 다이어그램

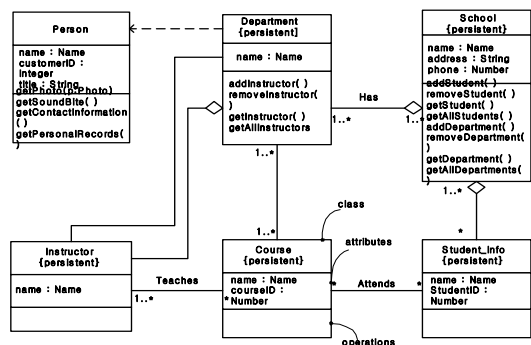
Use Case 다이어그램은 개발할 시스템의 기능적인 요구사항을 Use case, Actor 및 그들 사이의 관계를 함께 나타낸 것으로, 사용자 요구에 따른 시스템의 행위 또는 의미를 나타낸다[1,2]. 다음 <그림2>은 “학점 은행제”에서 필요로 하는 사항을 학습자와 관리자 및 시스템의 행위의 Use Case 다이어그램이다.



<그림2> Use Case 다이어그램

3.2.2 클래스 다이어그램

클래스(Class) 다이어그램은 객체 지향 분석 및 설계의 가장 중요한 요소로서 시스템 내부 클래스들간의 정적인 구조를 나타낸다. 클래스들은 서로간의 관계를 가질 수 있는데, 이러한 클래스들간의 관계는 속성이나 연산과 같은 클래스의 내부구조와 함께 정적 구조인 클래스 다이어그램에 나타나게 된다[1,2]. 또한 모든 클래스는 하나의 스테레오타입(stereotype)을 가질 수 있으며, Boundary, Entity, Control, Exception, Metaclass, Parameterized, Utility 클래스 등이고, 클래스의 이름 표기 부분에 “<<>>” 사이로 표시한다. <그림3>은 학점 은행제 운영에 필요한 전반적인 학사 관리를 효율적으로 처리하기 위한 개인정보, 학습과정, 학위정보 등을 나타낸 것이다.

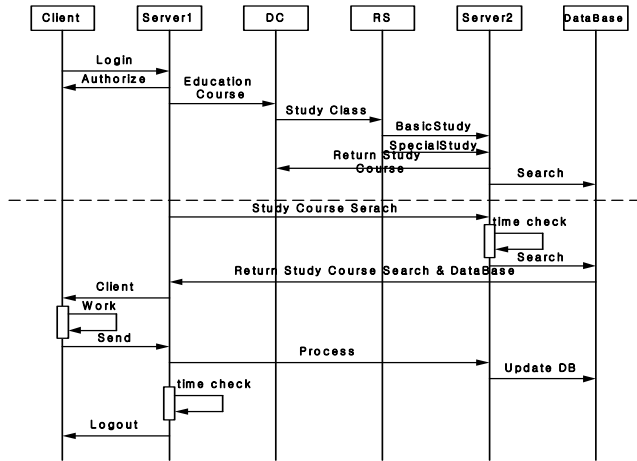


<그림3> Class 다이어그램

3.2.3 순차 다이어그램

Interaction 다이어그램에서 순차(sequence) 다이어그램은 여러 개의 객체들 사이에 동적인 협력사항으로 Use Case에서 분기된 객체 사이에 메시지를 보내는 순서를 나타낸다. 또한 수직선상의 여러 개의 오브젝트(object)들로 구성되며, 시간은 아래 방향으로 진행을 하고 순서나 함수로 시간이 지나가는 것에 따라서 오브젝트들 사이에 메시

지를 교환한다[1,2]. <그림4>은 클라이언트에서 학습과목과 전공에 관련된 학위 정보, 교육과정과 그 결과를 클라이언트에서 작업하는 과정이다.



<그림4> Sequence 다이어그램

3.3 웹 서버의 RDS 설계

RDS는 3-tier에서 클라이언트/서버 환경을 완벽하게 지원할 수 있는 객체이다. 클라이언트 계층은 웹 서버에서 데이터를 받은 후 데이터를 처리할 수 있는 능력을 가지고 있다. 클라이언트 계층에 있는 비즈니스 객체 프록시와 RDS 데이터 스페이스 객체는 중간 계층에 위치한 비즈니스 객체와 통신하기 위해 비즈니스 객체 프록시를 사용한다. 프록시는 HTTP나 DCOM을 통해 클라이언트의 데이터를 전송할 수 있게 해주고, 중간 계층의 비즈니스 객체가 클라이언트 호출에 의해 생성될 때마다 자동으로 만들어진다. 클라이언트 계층에서 RDS 데이터베이스 스페이스를 선언하고, 서버측 비즈니스 객체인 RDS 서버 데이터 팩토리로 통신시 필요한 프록시를 생성한다. 다음 질의를 넘겨주면 비즈니스 객체는 해당 결과에 대한 레코드 셋을 보내주고, 이것을 다시 RDS 데이터 컨트롤 객체인 ADC01에 결합시켰다.

RDS 서버 데이터 팩토리 객체는 사용자 질의를 받아 데이터 소스에 전달하고, 질의 결과를 인터넷을 통해 사용자에게 서비스하는 내장 비즈니스 객체이다. 이에 대한 VBScript 코딩은 <그림5>과 같다.

```
<HTML>
<BODY>

<! RDS.DataControl
<OBJECT CLASSID= BD96C556-65A3-11D0-983A-00C04FC29E33"
ID=ADC01></OBJECT>
<! RDS.DataSpace
<OBJECT CLASSID= BD96C556-65A3-11D0-983A-00C04FC29E36"
ID=ADS01></OBJECT>
...
<SCRIPT LANGUAGE=VBScript>
Sub Window_onLoad()
Dim ADF, myRs
Set ADF = ADS01.CreateObject(RDSServer.DataFactory, _
http://<%=Request.ServerVariables(SERVER_NAME)%>
Ser myRs = ADF.Query(UDI=sa;PWD=pwd;DSN=dsn; _
Select * from tb001
ADC.SourceRecordset = myRs
End Sub
...

```

<그림5> 사용자 질의에 대한 RDS 스크립트 코드

클라이언트 계층에서 RDS 데이터 스페이스를 선언하고, 서버측 비즈니스 객체인 RDS 서버 데이터 팩토리로 통신시 필요한 프록시를 생성한다. 그런 다음 질의어를 넘겨주면 비즈니스 객체는 해당 결과에 대한 레코드셋을 보내주고, 이를 다시 RDS 데이터 컨트롤 객체인 'ADC01'에 결합시켰다.

RDS를 이용하여 레코드셋을 얻을 수 있는 방법은 사용자가 직접 만든 비즈니스 객체를 이용하려면 먼저 중간층에 사용자 비즈니스 객체를 생성해 놓아야 한다. 이 객체는 비주얼 C++, 비주얼 베이직, 델파이 등을 이용해 쉽게 생성할 수 있으며, 컴포넌트화하면 웹 환경뿐만 아니라 일반 클라이언트/서버 환경에서도 DCOM을 이용해 동시에 사용할 수 있다. 이에 대한 VBScript 코딩은 <그림6>과 같으며, 단지 비즈니스 객체를 사용했다는 것을 제외하고는 RDS 서버 데이터 팩토리 객체를 이용하는 것과 거의 동일하다고 파악할 수 있다.

```
<! RDS.DataControl
<OBJECT CLASSID= BD96C556-65A3-11D0-983A-00C04FC29E33"
ID=ADC01></OBJECT>
<! RDS.DataSpace
<OBJECT CLASSID= BD96C556-65A3-11D0-983A-00C04FC29E36"
ID=ADS01></OBJECT>
...
<SCRIPT LANGUAGE=VBScript>
Sub Window_onLoad()
Dim myObject, myRs
Set myObject = ADS01.CreateObject
(MyCustomBusinessObject.DataFactory,
http://<%=Request.ServerVariables(SERVER_NAME)%>
Ser myRs = myObject.GetRecordset(' Select * from tb001 )
ADC01.SourceRecordset = myRs
End Sub
...

```

<그림6> Custom Business Object 스크립트 코드

<그림7>에서 사용자 비즈니스 객체의 메쏘드인 GetRecordset()는 한 개의 매개 변수를 갖고 있으며, 이를 통해 SQL 문장을 넘겨받게 된다. 함수 내부에는 질의 결과인 레코드셋을 저장할 변수와 데이터베이스 접속에 필요한 정보를 저장할 변수를 각각 하나씩 선언한다. 레코드셋의 오픈(Open) 메쏘드를 데이터베이스에 접속해 SQL문의 실행 결과를 넘겨받은 후에는 이를 호출한 클라이언트에 보내질 수 있도록 함수명을 치환한다.

```
Public Function GetRecordset( strSql As String) AS Variant
Dim RS As New ADODB.Recordset
Dim connStr As String

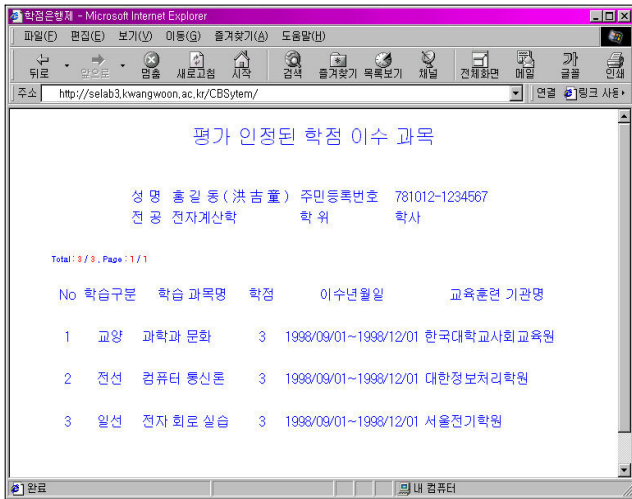
ConnStr = DSN = user: UID = user; PWD = user
RS.CursorLocation = adUseClientBatch
RS.Open Ssql, connStr
Set GetRecordset = RS
End Function

```

<그림7> 사용자 비즈니스 객체의 GetRecordset() 메쏘드

4. 적용 사례

<그림8>은 학점은행제를 위한 웹 데이터베이스 관리 시스템에서 인증된 사용자 ID와 패스워드를 입력한 후, 현재 이수한 과목에 대한 사용자 검색 질의에 대한 결과를 보여주는 실행 화면이다.



<그림8> 검색 질의에 대한 실행 예

5. 결론

본 논문에서는 객체 지향 방법론인 UML을 이용하여 웹 기반의 분산환경에서 학점은행제에 대한 사례를 적용하여 구현하였다. 따라서 개발자에게 프로그램의 이해와 유지보수를 효율적으로 수행할 수 있도록 하였으며, 서비스의 속도를 향상시키기 위해 서버에 인덱스 테이블을 구축하여 데이터를 복제와 이주하여 검색을 용이하게 할 수 있도록 하였고, MS사의 RDS를 이용하여 웹 서버의 관리 부담을 줄이고, 부하 집중을 막을 수 있도록 클라이언트 계층에서 데이터를 동적으로 처리할 수 있는 웹 데이터베이스 관리 시스템을 설계하고 구현하였다.

향후 연구과제로는 실세계를 보다 정확하게 표현하는 소프트웨어를 개발할 수 있도록 객체 지향 방법론인 UML의 “디자인 패턴”을 적용하여 소프트웨어 컴포넌트를 만들어 다량의 데이터를 동시에 액세스할 있는 다중처리 방법과 이에 대한 설계가 필요하다.

참고문헌

- [1] Booch, G., Rumbaugh, J., Jacobson, I., “The Unified Modeling Language User Guide”, Addison Wesley Longman, Inc, 1999.
- [2] Rumbaugh, J., Jacobson, I., Booch, G., “The Unified Modeling Language Reference Manual”, Addison Wesley Longman, Inc, 1999.
- [3] Alex Homer & david Sussman. ‘Professional MTS MSMQ with VB and ASP’, WROX Press, 1998.
- [4] Scot Hillier & Daniel Mezick. ‘Inside Programming Active Server Pages’, Microsoft Press, 1998.
- [5] Microsoft. ‘Inside Client/Server Programming with Visual Basic & SQL Server’, Microsoft Press, 1998.
- [6] Robert Orfali & Dan Harkey. ‘Client/Server

Programming with Java and CORBA 2ed Ed’, John Wiley & Song, Inc, 1998.

- [7] Frank E. Redmond III. ‘DCOM: Microsoft Distributed Component Object Model’, IDG Books, 1997.
- [8] 김영걸 역. "분산컴퓨팅과 클라이언트-서버 시스템", 도서출판 북플러스, 1996.
- [9] 박영수. "C/S모델의 분산 환경에서 CORBA명세를 이용한 객체 복제의 설계와 구현", 광운대학교 전산대학원, 1996.
- [10] 김중근외 4인. "웹과 DB연동시 CGI모델과 자바모델의 성능평가", 정보처리학회 봄 학술발표논문집 Vol.25, 1997.
- [11] 황선명의 2인. "분산 객체 환경을 위한 통합 개발 환경의 설계", 한국정보과학회 봄 학술발표논문집 Vol.25, 1998
- [12] 박장한의 2인, “RDS를 이용한 Web 데이터베이스 관리 설계 및 구현”, '99 춘계 학술발표논문집 1999, 04 v.6, n.1, pp.384-387
- [13] <http://www.microsoft.com/data/ado/rds/custhand.htm>
- [14] <http://support.microsoft.com/support/kb/articles/q190/7/73.asp>
- [15] <http://www.cayennesoft.com/products/wpaper/rdbms.htm>
Developing OO applications for Relational DBMS
- [16] <http://msdn.microsoft.com/library/psdk/dasdk/usin1x68.htm>