Journal of Korea Multimedia Society Vol. 7, No. 2, February 2004(pp. 272-282)

.NET 기반의 저부하형 웹 애플리케이션 설계 및 구현

최동우†, 안현식††

요 약

웹 서비스들이 보다 대형화 되면서 보다 많은 정보의 처리가 필요함에 따라 서버의 과부하를 방지하고 처리 속도를 향상시키기 위한 노력이 이루어지고 있다. 본 논문에서는 서버의 부하가 늘어나는 현상을 분석하고 최근에 등장한 .NET 기반의 저부하형 웹 애플리케이션 설계 및 구현 방법을 제안한다. 다중 접속시 서버의 부하를 최소화하기 위하여 관계형 데이터베이스를 설계하고 최소한의 모듈을 생성하였으며, SP를 이용하여 복잡한 SQL문을 단순화 하여 데이터베이스의 부담을 줄여 서버의 성능을 향상시켰다. 또한 User Control을 활용하여 페이지를 구성하여 페이지 처리 속도를 향상시켰으며, 스크립트를 활용하여 서버 언어를 이용한 작업을 최소한으로 줄였다. XML/EDI를 이용한 전자문서교환방식을 채택하여 관리비용을 줄일 수 있도록 하였다. 본 논문에서는 제안한 저부하형 시스템을 학사관리 시스템 상에서 구현하여 기존의 방법에 비해 보다 효율적 시스템임을 보인다.

Design and Implementation of Light Loaded Web Application Based on .NET

Dong-Woo Choi†, Hyunsik Ahn††

ABSTRACT

As Web services need to manage lots of information, it is indispensable to reduce server's load and speed up processing time. In this paper, we analyze the loading problems of server and suggest designing methods for light loaded Web applications based on .NET. The proposed methods include using a relative database to produce minimized modules and Stored Procedures to simplify SQL statements for reducing server's load. Web pages are organized with scripts replacing server language, which improve server's capability by reducing loads. The execution time is also speeded up by organizing Web pages with User Controls. And XML/EDI is used for managing the effective exchange of documents. The proposed methods are implemented as an education management system and shows its effectiveness.

Key words: .NET, Web Application(웹 애플리케이션), Education Management System(학사관리시스템), Database(데이터베이스)

1. 서 론

웹서비스의 대형화 고기능화 추세와 함께 보다 많은 정보들이 온라인상에서 송수신되고 있으며 보다 체계적인 관리 및 유지가 요구되고 있다. 웹 서비스의 보다 양질의 서비스를 위해서는 서버의 부하를 줄이고 호출 속도를 높이는 것이 중요하며 이를 위한 여러 가지 노력이 이루어지고 있다[1-3]. 특히 많은 사용자들이 동시에 접속하면 애플리케이션이 동시에 실행이 되므로 서버의 부하가 존재한다. 또한 웹

※ 교신저자(Corresponding Author) : 안현식, 주소 : 부산시 남구 용당동 535(608-711), 전화 : 051)610-8356, FAX : 051)610-8349, E-mail : hsahn@tmic.tit.ac.kr
접수일 : 2003년 6월 4일, 완료일 : 2003년 7월 25일
† 비회원, 동명정보대학교 대학원 로봇시스템공학과 석사과정 졸업 (E-mail : dwchoi@tmic.tit.ac.kr)
†† 정회원, 동명정보대학교 로봇시스템공학과 조교수
※ 이 논문은 2002학년도 동명정보대학교 학술연구비 지원에 의하여 이루어진 것임.

애플리케이션들은 다양한 기능을 지원하기 위해 지나치게 많은 모듈을 생성하면 한 모듈의 실행시에 파생하는 모듈이 연이어서 실행되게 되므로 역시 부하를 가중시키며 처리 속도가 저하되는 문제가 존재하게 된다. 이와 같은 문제를 해결하기 위해 J. Song 등은 서버의 부하를 줄이기 위한 목적으로 통신 스텍을 최적화 하고 캐시(cache) 데이터를 효율적으로 처리할 수 있는 웹 서버 가속기를 제안하였다[1]. B. Krishnamurthy 등은 사용자의 증가로 인한 웹서버의 부하를 분석하고 데이터 캐싱(caching)과 다중 서버의 데이터 분배로 통한 성능향상 방법을 제안하였다[2] 이러한 방법들은 주로 데이터 캐싱 방법을 이용한 서버의 성능을 향상시키는데 집중하고 있다.

본 논문에서는 웹 애플리케이션의 문제점을 서버와 클라이언트의 서비스 애플리케이션의 입장에서 분석하고, 최근에 등장한 .NET 기반으로 다중 접속시의 서버의 부하를 최소화하며 최소한의 모듈 생성을 위한 웹 애플리케이션의 설계 방법을 제안하며, 이를 학사관리 시스템 상에서 구현하여 효율적임을 보인다. 본 논문에서는 ASP.NET의 장점을 활용하는데 관계형 DB를 설계하고 저장 프로시저(SP: Stored Procedure)를 이용하여 복잡한 SQL 문을 단순화시킴으로써 관리를 보다 용이하게 함과 동시에 데이터베이스의 부담을 줄이며, 서버 언어의 사용을 줄이고 스크립트언어를 사용하여 서버의 부하를 줄인다. 또한 User Control을 이용하여 프레임과 같은 효과를 줌으로서 페이지 처리속도를 향상시켰다. 또한 XML/EDI을 이용하여 전자문서화 함으로서 기존의 EDI는 변환소프트웨어가 필요했던 단점을 개선하였다.

본 논문의 구성은 2장에서는 ASP.NET, ADO.NET 및 XML/EDI에 대한 관련연구를 언급하였으며, 3장에서는 본 논문에서 제시한 웹 애플리케이션 시스템의 설계를 자세히 서술하였다. 4장에서는 제안한 시스템을 학사관리 시스템에 구현한 결과를 보여준 후 이에 대한 고찰을 언급하고, 5장에는 본 논문의 결론을 맺는다.

2. 관련연구

본 장에서는 기존의 ASP와 비교하여 .NET 기반의 요소기술들의 장점과 XML/EDI의 특징에 대해 간략히 언급한다.

2.1 ASP.NET

기존의 ASP방식은 HTML과 혼합되어 사용되었으므로 코드를 ASP코드들을 별도로 분리 할 수 없었다. 그러나 ASP.NET은 코드 블록을 따로 정의 하여 코드부분과 사용자 인터페이스 부분을 별도로 분리하였으므로 코드의 재사용성과 보안성을 높였다. 또한 기존의 ASP방식은 스크립트 언어를 사용하여 해석하기 때문에, 실행 속도가 저하시키고 서버 성능을 저하 시킬 수 있었으나, ASP.NET에서는 분리된 구현 코드를 컴파일 하여 저장해두고, 호출시에 재 전송하므로 해석 시간을 줄일 수 있으므로 응답속도가 개선되었다. 또한 기존의 ASP에서는 스크립트 언어이기 때문에 라이브러리의 수가 제한되어 있으므로 다른 라이브러리가 필요할 경우 ActiveX 컨트롤을 제작하여 서버 컴포넌트로 만들어 사용하였다. 그러나 ASP.NET은 .NET 플랫폼을 기반으로 웹 애플리케이션 실행하기 때문에, .NET에서 제공하는 풍부한 클래스 라이브러리를 사용할 수 있으며 필요에 따라 별도로 클래스 라이브러리를 구축해서 쓸 수 있는 장점이 있다. ASP.NET페이지에서 사용할 수 있는 언어로는 C#, VB.NET, JSCRIPT.NET을 사용할 수 있으며 하나의 웹 애플리케이션에서 여러 언어를 혼용해서 쓸 수 도 있다[5, 9]. 그림 1은 ASP.NET의 WebForm과 Service의 관계를 나타내는 그림으로서 ASP.NET에서 System.Web 네임스페이스는 여러 개의 하부 네임스페이스를 사용할 수 있다[5].

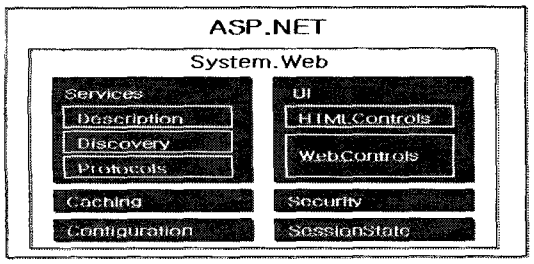


그림 1. WebForm과 Service

2.2 ADO.NET

기존의 ASP에서는 ADO(ActiveX Data Object)를 이용해서 데이터베이스와 연동하였으나, ASP.NET에서는 ADO.NET이라는 새로운 형태의 데이터베이스 연동기술을 사용하고 있다. ADO.NET은 DataSet이라는 새로운 개체를 도입하였는데 XML

을 지원하고 데이터베이스와 연결이 끊어진 상태(disconnect)에서 데이터에 액세스할 수 있어서 서버 측 부담을 줄인 장점이 있다. ADO.NET의 데이터 연결(connecting) 방법에는 두 가지로 나누어진다. 첫 번째는 데이터베이스와의 연결을 유지한 상태에서 ADO의 역할을 수행하는 .NET Data Provider이고 하는 연결하지 않은 상태에서 ADO의 역할을 수행하는 비연결 지향적인 DataSet이다. 이 구조를 간단하게 도식적으로 표현하면 그림 2와 같다[5].

.NET Data Provider는 데이터베이스와의 연결을 유지하면서 모든 일을 하는 것을 말한다. 따라서 데이터베이스의 종류에 따라 SQL 서버용과 OLEDB 데이터용, 그리고 ODBC를 이용할 수 있는 네임스페이스를 제공하고 있다. 이 네임스페이스(name space)들은 System.Data 네임스페이스에 속해 있고, 이 네임스페이스들을 사용하려면 System.Data 하부의 각각의 네임스페이스를 사용해야 한다. DataSet은 비연결 지향적인 데이터 클래스이며 데이터베이스에서 레코드를 가져온다는 점은 기존의 ADO의 RecordSet과 비슷하나 레코드를 가져온 후에 데이터베이스와의 연결을 종료한다는 점에서 차이가 있다. 즉 DataAdapter 클래스가 레코드를 가져와서 DataSet에 레코드를 전달하고 종료한다. 또한 이렇게 가져온 레코드는 XML문서 형식으로 클라이언트의 메모리에 저장하여 다시 호출될 수 있다. 또 다른 ADO.NET의 장점은 XML과 잘 부합되도록 설계되어 있다는 점이다. ADO.NET으로 응용 프로그램을 작성하면 XML의 유연성과 폭넓은 수용성을 활용할 수 있다. 즉 XML은 네트워크를 통해 데이터 집합을

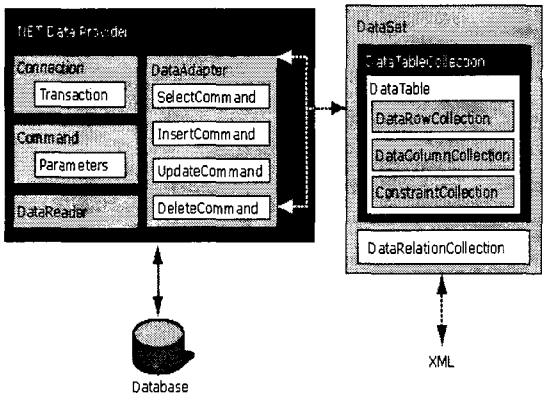


그림 2. ADO.NET의 Data Provider와 DataSet

전송하기 위한 형식이므로 XML형식을 읽을 수 있는 모든 파일에서 데이터를 처리 할 수 있다. 전송 구성요소는 수신 구성요소가 구현된 방식과 상관없이 단순히 데이터 집합을 대상으로 전송할 수 있기 때문에 수신 구성요소는 ADO.NET의 구성요소가 아니더라도 관계가 없다. 여기서 대상 구성요소는 Visual Studio로 작성된 Windows 형태의 도구를 사용하여 구현된 기타 모든 응용 프로그램이 될 수 있다. 유일한 요구 사항은 수신 구성요소에서 XML을 읽을 수 있도록 구성되어야 한다는 점이다.

2.3 XML/EDI

현재 기업간의 거래를 하면서 문서를 주고받는 방법에는 VAN(Value Added Network) 을 이용한 전통적인 EDI(Electronic Data Interchange)나, 웹 서버를 구축하여 인터넷 기반으로 거래를 하면서 문서를 보여줄 때는 HTML tag를 사용하고 있다. VAN을 이용한 EDI는 가입 회사에 특정한 클라이언트 프로그램이 필요하다는 점에서 폐쇄적인 방식이며, 유지보수에 상당한 노력이 필요하고, Internet EDI의 경우 개방형이지만 HTML tag로 보여주는 문서를 화면으로 보기만 할 뿐, 내부 시스템과 연동하여 데이터로 활용하려면 별도의 plug-in이 필요하다. XML/EDI는 주고받는 문서를 XML로 만들어서 상대방과 송수신할 수 있도록 하였다[4,10]. 이것은 Internet을 이용하여 문서를 송수신하는 것은 마찬가지이지만, XML로 주고받은 문서는 내부 시스템과 쉽게 연동이 가능하여 자체 데이터로 활용이 가능하게 되었다. 따라서 유지 보수의 비용을 줄일 수 있으며 다른 EDI와의 연동을 자유롭게 할 수 있다.

3. 저부하형 웹애플리케이션의 설계 방법

3.1 관계형 데이터베이스 설계

본 논문에서는 데이터베이스 설계를 관계형 데이터베이스로 설계하였다. 관계형 데이터베이스 설계는 정규화를 거친 후 비정규화를 통해 데이터 모델링을 실시하는 방법으로서 테이블의 컬럼을 수정하고자 할 경우 빠른 대처를 할 수 있으며 재사용성을 높일 수 있다. 그림 3은 학사관리 시스템을 예로 들어 웹 애플리케이션의 데이터베이스의 개념설계인 ER

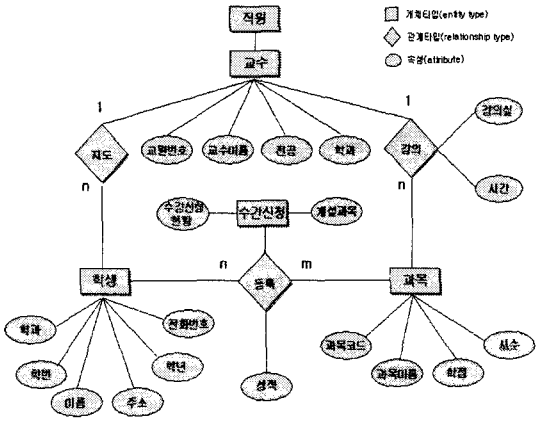
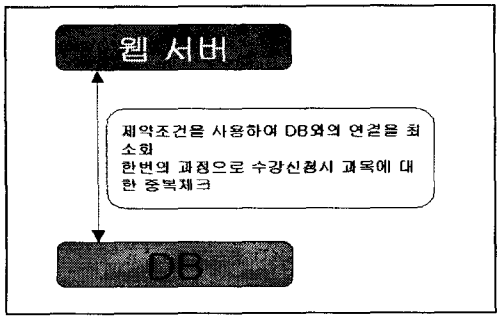


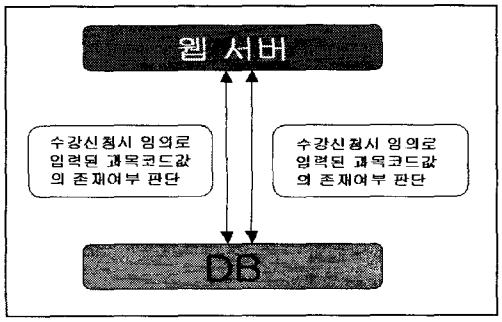
그림 3. 관계형 데이터베이스 개체 모델 구성

(Entity Relationship) 모델링을 실시 한 결과를 보여주고 있다.

먼저 정규화 과정을 통해 그림 4와 같이 불필요한 테이블의 생성을 줄이고 중복성이 없는 컬럼으로 구성된 테이블을 설계하였다. 중복성이 필요한 컬럼에는 PK(Primary Key)를 지정하였으며, 연관성 있는 테이블 간의 컬럼들의 관계를 만들기 위해서 FK (Foreign Key)를 생성하였다. 예를 들어 '과목'에서는 PK와 FK를 동시에 설정하고, 수강에서는 FK로 설정하였다. 만약 수강에서 임의의 과목코드를 입력시 제약조건에 의해서 입력된 과목코드가 과목의 과목코드에 존재하는지를 체크하고 그 과목이 존재하지 않으면 True를 반환됨으로 그림 5(a)와 같이 웹서버와 데이터베이스연결이 하나의 과정으로 종료된다. 그러나 이 설정이 없을 경우 임의로 입력된 과목코드가 존재하는지 여부를 확인하기 위한 과정이 별도로 필요하게 되므로 그림 5(b)와 같이 연결과정이 늘어나게 된다. 따라서 데이터베이스의 규모가 줄게 되어 데이터 처리속도가 향상되게 된다.



(a) 제약조건으로 데이터베이스 구성시



(b) 제약조건 없이 데이터베이스 구성시

그림 5. 정규화 효과

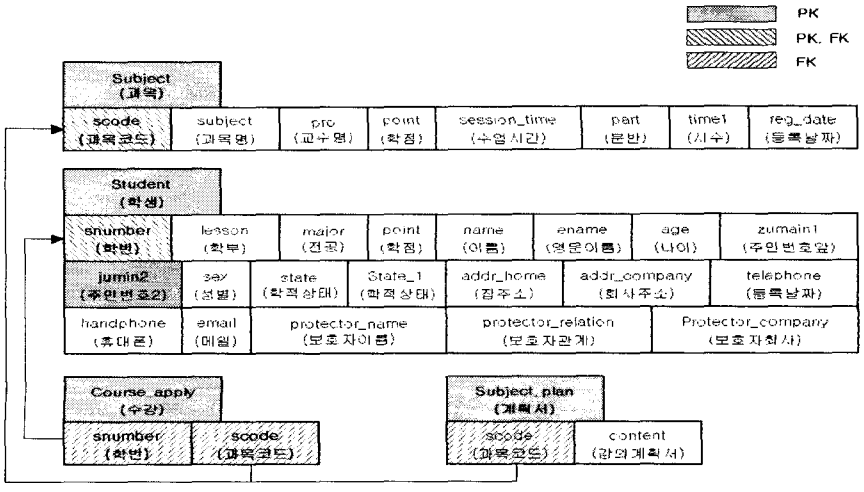


그림 4. 정규화 과정

3.2 저장프로시저 구성

본 논문에서는 저장프로시저를 이용하여 복잡한 SQL문을 단순화함으로써 네트워크 트래픽을 줄인다. 그림 6은 저장프로시저를 이용한 예로서 그림 6(a)는 저장프로시저 생성구문이며 그림 6(b)는 SP를 생성하여 데이터베이스에 저장하는 부분으로 학번별 수강신청리스트의 경우이다. 학번컬럼(snumber)을 변수 @snumber로 정의하고 학번의 값을 받아서 쿼리(query)문에서 실행된 결과값 만을 반환한다. 그림 6(c)는 데이터베이스에 저장된 SP를 ASP.NET에서 호출하는 부분이다. 메소드에서 string형으로 받은 snumber값을 SP의 snumber변수에 넘겨주며 SP에서 실행된 쿼리문의 값을 반환하여 사용한다. 여기서는 하나의 쿼리문이 여러 부분에 사용되므로 특정 쿼리문을 수정시 사용되어진 모든 부분을 각각 수정해야 한다. 그러나 SP를 사용하게 되면 SP로 저장된 부분만을 수정하므로 사용된 부분을 재사용할 수 있다. 또한 SP를 이용함으로써 기존의 컴파일 된 결과를 다시 호출하게 되므로 빠른 속도를 얻을 수 있다.

3.3 저부하형 프레임처리

본 시스템은 저부하형 프레임처리를 위하여 User Control을 사용하였다. 각 페이지에 로딩될 때 프레임으로 나누어져 있던 것을 User Control로 포함 시키면 프레임으로 구성되어 각각의 프레임을 로딩하

```
CREATE PROC[EDURE] procedure_name [;number]
    [
        {@parameter data_type} [VARYING] [= default] [OUTPUT]
    ]
    [,...n]
[WITH
    {
        RECOMPILE
        | ENCRYPTION
        | RECOMPILE, ENCRYPTION
    }
]
[FOR REPLICATION]
AS
    sql_statement [...n]
```

(a) 저장프로시저 구문

```
Create proc tbl_sugang
        @snumber        int

As
        select distinct * from course_apply c, subject s where
        sunmber = @snumber and c.scode = s.scode
go
```

(b) 학번별수강신청리스트 tbl_sugang SP생성

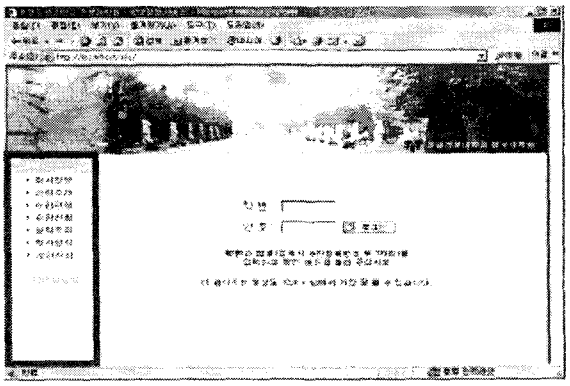
```
    public string Login(string snumber, string scode) {
    SqlConnection myConnection =
        new SqlConnection(ConfigurationSettings.AppSettings["ConnectionString"]);
    SqlCommand myCommand = new SqlCommand("tbl_sugang", myConnection);
    myCommand.CommandType = CommandType.StoredProcedure
    .....
```

(c) tbl_sugang SP를 호출해서 쓰는 부분

그림 6. 저장프로시저

는데 필요한 시간을 제거할 수 있다. 그림 7은 User Control을 사용하고 있는 예로서 로그인 페이지의 좌편 하단의 굵은 선으로 표시된 부분의 서브메뉴를 menu.ascx로 구성한 예이다. 메인페이지에서 header부분에 User Control을 정의하고, 서브메뉴가 사용되어지는 위치에 삽입하여 사용하였다. 기존의 방법대로 하나의 프레임으로 구성하면 프레임에 속한 모든 페이지들을 전부 로딩하게 된다. 그러나 User Control을 사용하면 프레임으로 구성된 페이지와 같이 각각의 페이지를 전부 로딩하지 않고 나누어진 일부만을 로딩하게 되므로 로딩량이 줄게 되어 속도가 향상된다.

또한 각 페이지 마다 페이지에 로딩되는 메뉴의 연결 정보나 동적 데이터들을 XML에 저장하여 마우스의 이벤트가 발생시에만 XML에서 불러오게 하였다. XML은 페이지 리로드(reload)가 없기 때문에 데



(a)

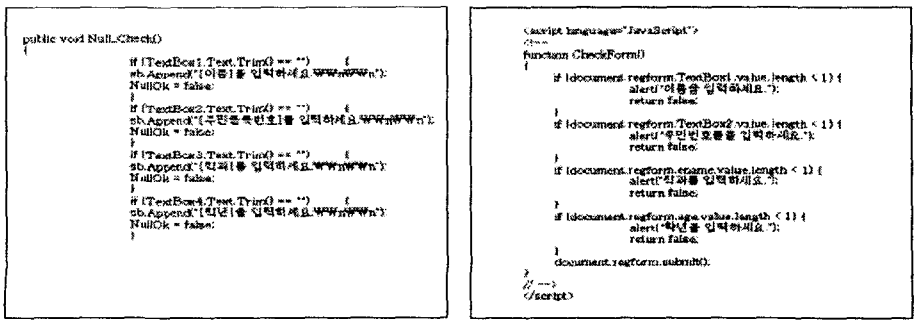
```
<%@ Register TagPrefix="Sub_menu" Tagname="menu" Src="Menu.ascx" %>
                                        // head 부분에 User Control 정의

                    ..................

<sub_menu:menu id="menu" Runat="Server" />
                                        // 메뉴가 들어갈 부분에 삽입
```

(b)

그림 7. User Control 사용 예, (a) 실행결과와 (b)코드 부분



(a) (b)

그림 8. 스크립트 사용 예 (a) 서버언어 aspx.cs에서 빈칸 확인 과정, (b) 스크립트를 이용한 빈칸 확인 과정

이터를 읽을 때 생기는 화면의 리로드가 필요 없으며, 페이지의 전체의 내용을 다 로딩 하지 않고 마우스 이벤트 발생시만 데이터를 읽어 오므로 페이지 로딩속도가 향상 될 수 있다. 그림 8과 같이 서버의 작업량을 줄여서 속도의 향상을 위해서 서버 언어의 사용을 최대한으로 줄이고, 스크립트를 많이 사용하였다. 스크립트는 클라이언트에서 컴파일된 상태로 저장되어 있다가 실행되므로, 서버작업량이 줄어들게 되며 처리속도가 향상되게 된다.

4. 저부하형 학사관리 시스템의 구현

본 논문에서는 .NET 기반 저부하형 웹 애플리케이션을 학사관리시스템에 적용하여 구현하였다. 구현한 시스템은 Windows 2000 Sever기반이며 웹서비스를 하기 위해서는 IIS5.0을 사용하였다. 데이터베이스는 Windows 2000 Server의 주된 플랫폼인 MS-SQL 2000을 이용하였다.

4.1 학사관리시스템의 데이터베이스

본 시스템 데이터베이스를 관계형으로 구현하기 위해 요구분석을 통하여 필요한 기능들만을 구성하여 서버의 부하를 줄였다. 이를 위해 그림 3의 모델을 근거로 하여 학사관리 시스템의 일부분인 학적관리, 수강신청관리, 성적조회관리, 공지사항관리, 교원관리로만 구성하였으며, 학생과 교원, 직원의 컨텐츠간의 상호 작용성을 중시하는 학사관리시스템으로 설계하였다. 본 시스템의 관계형 데이터베이스는 그림 9의 ER 다이어그램과 같이 여러개의 테이블로 구성하였다.

student 테이블에는 학생들의 신상과 수강 신청시 필요한 데이터를 포함 하고 있다. course_apply 테이블에는 수강 신청시 수강한 학번과 수강한 코드를 저장한다. subject 테이블은 subject_all 테이블에 과목코드로 당해 학과의 개설과목을 생성하여 과목코드별 내용을 저장한다. record와 record_term 테이블은 성적조회를 위한 테이블이며, notice 테이블은 공지사항, pds는 자료실, free_board는 게시판을 위한 테이블들이다. student 테이블의 학번 컬럼과 course_apply 테이블에 있는 학번 컬럼에 제약 조건을 주었다. 수강신청시 student테이블에 없는 학번을 입력시 참조 무결성을 위배 하므로 수강신청이 이루어질 수 없게 된다. 이를 해결하기 위해 course_apply 테이블과 subject 테이블은 과목코드로 제약조건을 주었다. subject 테이블에 없는 과목코드를 입력할 경우에도 동일한 방법으로 참조 무결성을 위배하지 않도록 처리하였다. 또한 SP를 이용해서 데이터의 축적으로 인해서 생기는 속도저하를 줄였다.

4.2 수강신청 서비스

본 절에서는 저부하형으로 구현한 수강신청 부분을 살펴본다. 먼저 각 페이지는 User Control을 이용해 프레임을 나누지 않고 부분적인 메뉴 부분을 로드하는 방식으로 프레임을 구성하여 속도를 향상시켰다. 또한 수강신청시 그림 10과 같이 필요한 다양한

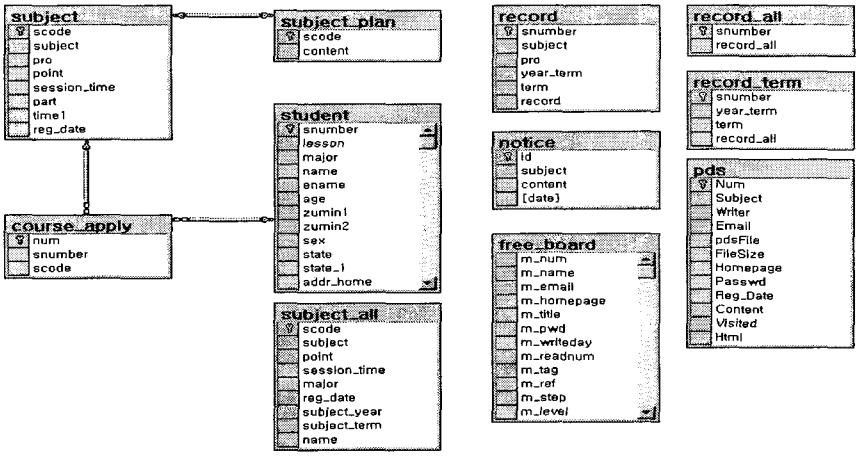


그림 9. ER 다이어그램

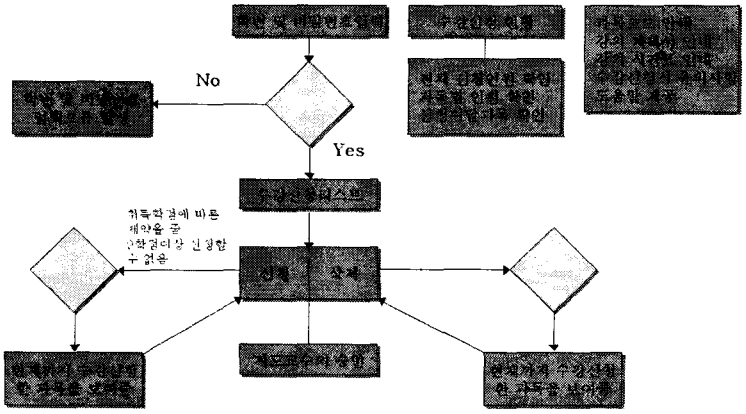


그림 10. 수강신청 절차

모듈들을 최소화하여 서버의 부하를 줄여 속도를 향상시켰다. 또한 테이블에서 필요한 값들을 얻기 위해서 서버언어에만 의존하면 서버부하로 인해 데이터 처리속도가 떨어지게 된다. 이를 개선하기 위하여 그림 4의 테이블디자인을 이용하였는데, 테이블에 필요한 값을 얻어오기 위해 사용되는 서버 언어의 작업량을 줄이기 위하여, SQL 쿼리문에서 중복성 체크나 참조 무결성 체크를 통해서 필요한 데이터를 가져오게 하였다.

4.3 XML/EDI에 의한 전자문서

XML/EDI시스템은 컴퓨터 간에 서로 합의된 메시지 표준형식을 이용하여 구조화된 데이터를 전자적인 수단을 통하여 정보를 전송하는 시스템이다. XML/EDI를 본 학사관리시스템에 도입함으로써 기존의 학교의 문서교환방식인 파일을 전송하고 전송된 파일을 수정한 후 다시 전송하는 번거로움을 줄일 뿐 아니라 문서표준을 만들어 데이터 재입력으로 인해 생기는 오류발생 과 처리에 드는 비용을 절감할 수 있다. 이러한 장점들을 활용하여 근로학생신청서, 장학금 신청서, 학점포기 신청서등 각종 신청서를 웹상에서 작성하게 하였다. 그림 11은 웹상에서 작성하게 되는 신청서를 보여주고 있으며, 그림 12는 작성된 신청서 폼의 정보들을 XML문서로 작성하는 부분이다. 그림 13은 작성된 XML문서를 보여주고 있다. 이렇게 작성된 신청서는 XML로 전송되어 신청의 종

근 로 장 학 생 신 청

성 명	최홍우	주민등록번호	******-******* ('-' 기호를 넣으세요)		
소 속	로봇시스템 학과	4 학년	학번	96111038	주야구분	주
주 소	부산광역시 수영구 남천 2동 남천비치 아파트 205동 211호				
연락처	☎ (051) 621-0001				
	휴대폰 018-571-0001				
은행명	주택				
계좌번호	1111-1111-1111				
장학종류	근로장학 (A, B, 특별 중 택일) A				
희망부서	정보공학부				

2002 년 11월 20 일

교 학 처 장 귀 하

그림 11. EDI 전자문서 작성폼

```
string filename, filename2;
filename = Server.MapPath("XmlData/giveup_"+ DataTim.Now.ToString("yyyMMdd") + "_" +
Convert.ToString(Session["roll_num]) + ".xml");
filename2 = "giveup_" + DataTime.Now.ToString("yyyyMM dd") + "_" +
Convert.ToString(Session["roll_num"]) + "xml";
XmlTextWriter Writer1;
Writer1 = new XmlTextWriter(filename, Encoding.Default);
Writer1.formatting = Formatting.Indented;
Writer1.WriteStartDocument();
Writer1.WriteComment("학점포기 신청서");
Writer1.WriteStartElement("giveup");
Writer1.WriteAttributeString("uidx", Convert.ToString(Session["idx"]));
Writer1.WriteElementString("year", TextBox1.Text);
Writer1.WriteElementString("term", TextBox2.Text);
Writer1.WriteElementString("faculty",TextBox3.Text);
Writer1.WriteEndElement();
Writer1.Flush();
Writer1.Close();
DataSet ds = new DataSet();
ds.ReadXml(filename);
string name;
localhost1.tit_Auth_ws ws = new localhost1.tit_Auth_ws();
name = ws.ot_getUseinfo((Convert.ToString(Session["Roll_n um"])), "name", "[User]",
"roll_num","");
localhost2.ws aws = new localhost2.ws();
aws.Receive_XML(filename2, name, "Giveup", Convert.ToString(Session["roll_num"]), ds);
Response.Redirect("haksa/ok.html");
```

그림 12. XML을 이용한 문서 전송

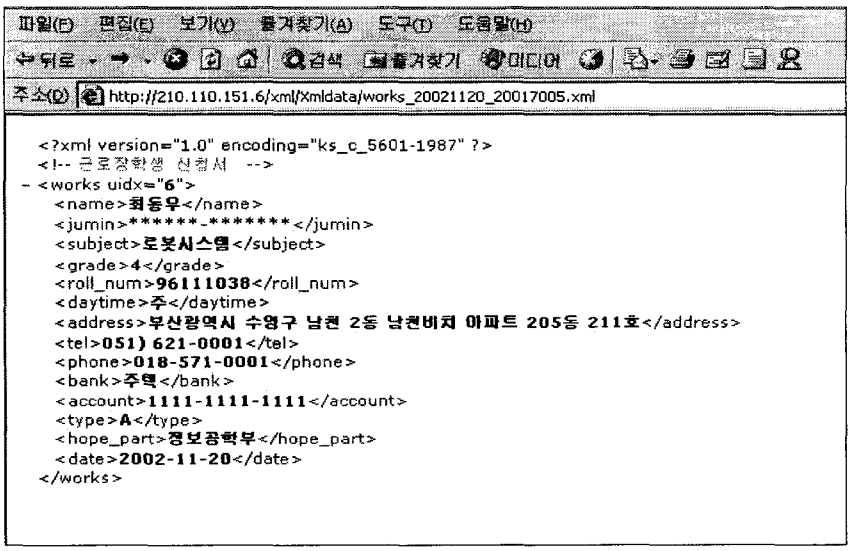


그림 13. XML/EDI 변환된 XML문서

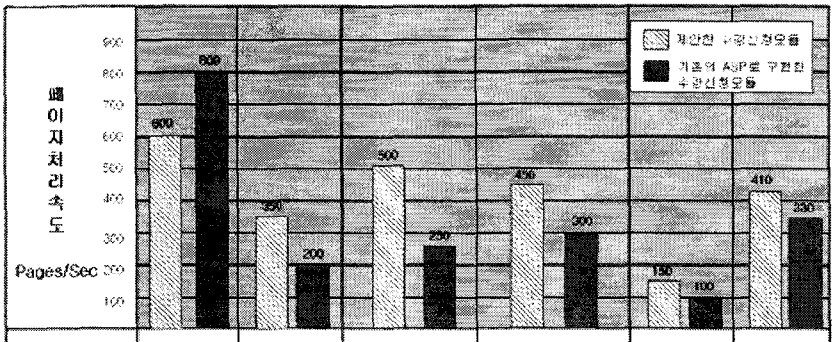
류와 날짜를 구분하여 지정된 폴더에 XML파일로 저장되게 된다. 저장된 XML문서는 관리자가 그 문서를 확인하게 되면 자동으로 백업을 위한 폴더로 이동하게 되어 있다. 문서를 백업하기 위한 폴더들은 파일 스트림을 이용해 문서의 종류별로 날짜별로 구분하여 분류하였다.

5. 성능 평가

본 논문에서는 제안한 시스템의 성능을 평가하기 위해 Window 2000 server에 내장되어 있는 성능 모니터링 콘솔을 사용하였다. 이 툴은 클라이언트의 로드, 연결 수, 쿠키형식, HTTP 응답 헤더 등을 모니터링 할 수 있다. 성능 평가를 위해 ASP 기반의 수강신청 시스템을 별도로 구성하였고, 제안한 ASP.NET 기반의 수강신청 시스템과 비교하였다. 성능 모니터링 콘솔은 기존의 ASP의 경우 서버 컴퓨터에 대한 성능 카운터는 전체적으로 평가되지만 APS.NET에서는 응용프로그램별로 평가가 가능하다. 본 논문에서는 동일한 조건하에서 평가하기 위하여 ASP.NET이 전체적으로 평가가 될 수 있도록 ASP에 들어 있는 로드, 연결 수, 쿠키형식, HTTP응답 헤더 등의 기능을 추가하였다. 두 경우에 대해 동일한 서버에서 수강신청 서비스를 실행하면서 페이지 항목별 페이지 처리 속도를 비교하였다. 그림 14는 비교한 결과를 도표로 보여주고 있다. 로그인의 경우 첫 페이지를 로딩 하는 시간은 ASP로 구성된 시스템에 비해서 본 논문에서 제안한 방식이 최초 컴파일하는 시간 때문에 처리 시간이 더 많이 소요되는 것을 볼 수 있다. 그러나 수강신청, 과목코드 조회, 수강신청, 내용 삭제 등 접속 후 페이지별 이동에 대한 페이지처리 속도는 기존의 방법을 이용한 경우보다 성능이 향상된 것을 볼 수 있다. 페이지 처리 속도가 향상된 원인은 User Control을 활용함으로써 기존의 프레임으로 구성되어 있던 ASP기반 시스템에 비해 페이지 처리속도가 향상된 점을 지적할 수 있다. 또한 서버의 작업을 줄이기 위해 서버언어의 작업량을 줄이고, 스크립트를 활용하므로 ASP기반 시스템에 비해 페이지 처리속도가 향상됨을 알 수 있다. 특히 데이터베이스에서 많은 데이터를 가져와야하는 과목코드 조회시에는 처리속도가 2배 이상 향상 된 것을 볼 수 있다. 이것은 제안한 관계형 데이터베이스 설계와 SP를 활용한 결과로 판단할 수 있다.

6. 결 론

본 논문에서는 온라인상의 웹 애플리케이션의 경우 다중접속의 부하에 대한 대처와 보다 빠른 실행을 위해 .NET을 기반으로 한 저부하형 웹 애플리케이션 설계 방법을 제안하였으며, 이를 학사관리 시스템에 적용하여 구현하였다. 제안한 시스템은 관계형 데이터베이스를 설계하여 재사용이 가능하고, 모듈을 최소화 하도록 생성하였으며, SP를 이용하여 복잡한 SQL문을 단순화하여 처리속도가 향상되도록 하였다. 또한 ASP.NET에 포함되어 있는 User Control을 활용하여 페이지를 구성하여 프레임으로 나누어져 있던 페이지에 비해 페이지 처리 속도를 향상시켰고 스크립트를 활용하여 서버언어의 작업을 최소화함으로써 서버의 부하를 줄일 수 있었다. 또한 XML을 활용하여 이벤트가 발생하지 않을 경우 페이지의 동적 데이터를 로딩하지 않도록 하여 페이지의 로딩



항목	로그인	수강신청	과목코드조회	수강신청 내역	삭제	평균
1인당 평균 Page View	2	2	3	1	2	10
View 빈도수	20%	20%	30%	10%	20%	100

그림 14. 페이지항목별 성능비교

속도를 향상시켰다. XML/EDI를 활용하여 기존의 EDI에 비해 문서전송에 있어서의 효율성을 높였다.

본 논문에서 제안한 시스템은 현재의 그룹웨어나 학사관리시스템과 같이 대형 웹 애플리케이션을 설계시 서버의 부하를 줄이고 처리 속도를 향상시키기 위한 설계방법으로 활용될 수 있다. 본 논문은 현재 사용되고 있는 .NET을 이용한 서비스의 효율성을 극대화 하기 위한 웹애플리케이션에 대해 논의를 하였으나 새로운 웹 개발환경의 개발시에는 부하 문제를 개선하기위한 보다 근본적인 방안이 강구되어야 할 것이다.

참 고 문 헌

[1] J. Song, A. Iyengar, E. Levy-Abegnoli and D. Dias, "Architecture of a Web Server Accelerator," *Computer Networks*, Vol.38, No.1, pp. 75-97, Jan. 2002.

[2] B. Krishnamurthy and C. E. Wills, "Analyzing Factors that Influence End-to-End Web Performance," *Computer Networks*, Vol.33, No.1, pp. 17-32, Jun. 2000.

[3] 문진용, 구용완, "인터넷 상에서 PHP를 이용한 학사관리 시스템의 설계 및 구현," 한국정보처리학회 논문지, 제7권, 제10호, pp. 3148-3154, 2000.

[4] 고민정, 채기준, "교육망에 XML/EDI를 도입한 문서 관리 시스템," 한국컴퓨터교육학회 논문지, 제3권, 제1호, pp. 151-160, 2000.

[5] R. Anderson et. al., *Professional ASP.NET 1.0*, Wrox Press, 2002.

[6] N. Sundaresan and R. Moussa, "Algorithms and Programming Models for Efficient Representation of XML for Internet Applications," *Computer Networks*, Vol.39, No.5, pp. 681-697, Aug. 2002.

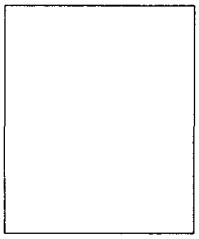
[7] R. Agrawal, R. J. Bayardo Jr., D. Gruhl and S. Papadimitriou, "Vinci: a service-oriented architecture for rapid development of Web applications," *Computer Networks*, Vol.39, No.5, pp. 523-539, 2002.

[8] A. Banerjee et. al., *C# Web Services*, Wrox Press, 200.

[9] http://msdn.microsoft.com/library

[10] http://www.kiec.or.kr

[11] http://www.ebxml.org/



최 동 우

2001년 2월 동명정보대학교 로봇시스템공학과 졸업(학사)
2003년 2월 동명정보대학교 대학원 졸업(공학석사)

관심분야 : 멀티미디어통신, 영상정보처리, 멀티미디어 시스템

안 현 식

1986년 2월 경북대학교 전자공학과 졸업(공학사)
1989년 2월 경북대학교 대학원 전자공학과 졸업(공학석사)
1998년 2월 경북대학교 대학원 전자공학과 졸업(공학박사)
1992년~1998년 포항산업과학연구원(RIST) 선임연구원
1998년~현재 동명정보대학교 로봇시스템공학과 조교수
관심분야 : 영상처리, 컴퓨터비젼, 멀티미디어통신, 지능로봇