

단순 웹브라우저의 설계와 구현 사례 (Case Study of Building Special-Purpose Simple Browser)

우 원 택*
(Won Taek Woo)

요 약 본 연구는 인터넷 익스플로러 컨트롤을 사용하여 자신만의 웹브라우저를 만들어 봄으로써 브라우저에 대한 이해를 높이고 자신만의 고유목적의 브라우저를 제작하여 사용함으로써 특정인의 특정도메인 이외의 검색을 지양한다든지 청소년과 같은 어린이의 경우 특정도메인에 대한 접속을 방지한다든지 하는 기법들을 활용 할 수 있는 능력을 제고하고자 함에 있다. 이를 위해 첫째로 웹브라우저는 어떠한 것이며 어떻게 작용하는지를 알아보기 위해 약간의 문헌조사를 하였고 둘째로 하나의 단순한 웹브라우저를 비주얼 베이직을 이용하여 직접 설계하여 구현해 보았다. 웹브라우저 설계는 이미지컨트롤, 상태바, 이미지컨트롤을 담은 픽처박스, 커맨드버튼, 메뉴 등의 컨트롤들과 프로그램코딩을 종합적으로 하나의 폼에 삽입하여 구현하였고 그 결과는 잘 실행되었다. 셋째로 본 웹브라우저에 필터링 기능을 삽입하여 유해 사이트를 차단할 수 있는 특수웹브라우저 기능도 테스트 해 보았다. 본 연구는 프로그램의 구조나 기능이 다소 단순하다는 제한점은 있으나 웹브라우저와 그 특수기능을 이해하는 데는 상당히 유의하였다.

Abstract The purpose of this study is to build a special-purpose browser using the powerful Microsoft web browser control so that we will be able to code and create a special-purpose browser. For this development, some literature surveys were reviewed to understand how the web browser looks like and is operated. To create our own web browser, we have designed a sample case and coded in Visual Basic. The primary task of this job was accomplished mainly with a form. It is made up with a bunch of image controls, a status bar on the button, a picture box to contain the image buttons, the combo box, the command button, the menus, the statements shown in the following exhibits and tables of the main text respectively. After combining them all together, we have tested the form to see if it works properly. The program was also enhanced to have a special purpose feature that permits the filtering function. The result was somewhat successful. However, it had some limitations such as the simplicity of the programming structures and functions. Overall, this study was useful for understand a general outline of special-purpose web browser development.

1. 서 론

인터넷 프로그램, 네트워크 프로그램 하면 많은 사람들이 생각하기에 어려운 것으로 알고 있다.¹⁾ UNIX 기반에 소켓 프로그램을 작성해 본 사람이라면 통신 프로그램을 작성하는 것이 정말 힘들고 많은 시간과 노력을 요구하는 것이라는 것을 알고 있다. 그렇기 때문에 통신 프로그램은 프로그램 언어를 능숙하게 다루는 사람들의 전유물

로 생각되었다. 하지만 비주얼 베이직에는 이런 수고와 노력 없이 간단하게 네트워크 프로그램과 인터넷 프로그램을 작성할 수 있도록 다양한 Active X 컨트롤을 제공한다.[1] ActiveX는 전통적 의미의 라이브러리에서 시작된다. 라이브러리를 도입하면 한번 제작한 코드를 다른 프로그램에서 재 사용할 수 있다. 라이브러리는 윈도우가 제작되면서 동적 링크 라이브러리(DLL: Dynamic Link Library)로 발전되었다.²⁾ 한편 OLE(Object Linking and

본 논문에 유익한 조언을 해 주신 익명의 심사위원께 감사드립니다.
* 경산대학교 정보과학부 조교수
1) 서현근, 전상훈, 구자광 공저, 비주얼베이직 6.0, 내하출판사, 2000, p.475

2) 일반적인 라이브러리는 각각의 프로그램에 부착된 형태로 사용된다. 즉 만약 3개의 윈도우 프로그램이 같은 라이브러리를 참조한다면 3개의 같은 라이브러리가 메모리에 올라오게 된다. 이러한 메모리의 낭비를 막기 위해 도입된 개념이 DLL이다. DLL은 응용프로그램이 시작될 때 메모리에

Embedding)는 객체간의 상호 통신과 파일 저장구조에 관한 기술로 OLE 컨트롤은 컴포넌트 기반의 프로그램의 중요한 개념으로 성장하였다. 많은 사람들에 의하여 OLE의 중요성이 인지되었고 OLE 기술은 컨트롤 직접화를 지원하기 위한 개발 환경 표준을 제시하게 되었고 1996년 마이크로소프트사는 이러한 컴포넌트 기반의 개발 환경을 인터넷 기술과 접목할 수 있는 ActiveX를 발표하게 되었다. 비주얼 베이직에서 제공하는 대표적인 인터넷 관련 컨트롤은 Winsock(TCP/UDP) 컨트롤³⁾과 Internet Transfer(Inet) 컨트롤⁴⁾, Web Browser 컨트롤⁵⁾ 등이 있다. 마이크로소프트의 비주얼 베이직은 그 자체가 인터넷 익스플로러가 설치되어야만 설치가 가능한 프로그램이다. 따라서 모든 비주얼 베이직 환경 하에서는 인터넷 익스플로러 컨트롤을 사용할 수 있다. 즉 Web Browser 컨트롤은 인터넷 익스플로러의 껍질만 벗겨버리고 핵심 엔진만 남은 알맹이와 같다고 말할 수 있다. 컨트롤에게 '어느 주소로 가라' 라든지 '검색 페이지로 등록된 홈페이지로 가라'라는 식으로 단순한 명령만 내려주면 된다. 모든 세부적인 동작은 컨트롤 객체 안에 캡슐화되어 사용자에게 숨겨진다. 이러한 컨트롤을 사용하면 쉽게 웹 브라우저를 만들 수는 있지만 자세한 내용은 전혀 알 수가 없으므로 이러한 컨트롤을 사용하는 것에 만족한다면 그저 단순한 노동자로 남게 될 뿐이다.[2] 다시 말해서 컨트롤을 사용한 프로그램이라는 것의 한계를 보여주고 있다고 생각할 수도 있다. 이와 같이 Window 프로그램은 객체지향 프로그래밍의 원리인 캡슐화(Encapsulation)⁶⁾, 상속성(Inheritance)⁷⁾, 다형성(Polymorphism)⁸⁾ 등을 통하여 프로그램 함으로써 프로그래밍을 하기는 용이하나 그 속의 깊이를 알기는 어렵다는 것을 알 수 있다. 결론적으로 단 몇 시간만에 웹 브라우저를 만들 수 있게 된다. 본 연구에서는 인터넷 익스플로러 컨트롤을 사용하여 자신만의 웹 브라우저를 만들어 봄으로써 브라우저에 대한 이해를 높이고 자신만의 고유목적의 브라우저를 제작하여 사용함으로써 특정인의 특정도메인 이외의 검색

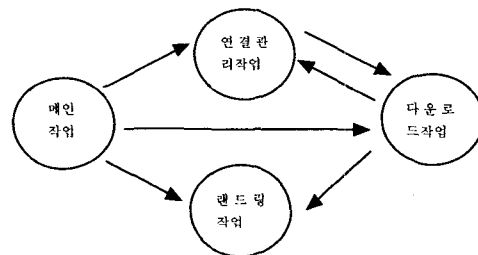
을 지양한다든지 청소년과 같은 어린이의 경우 특정도메인에 대한 접속을 방지한다든지 하는 기법들을 활용할 수 있는 능력을 제고하고자 함에 있다.

2. 웹브라우저의 구조

웹브라우저는 웹 문서를 사용자에게 보여주는 기능을 한다. 기존의 익스플로러나 넷케이프와 같은 범용 웹브라우저는 HTML로 작성된 웹문서를 취득하고 해석하여 브라우저 사용자에게 시각적 인터페이스를 통하여 문자나 그래픽을 보여준다.[3] 일반적 웹브라우저의 기능은⁹⁾ 첫째 텍스트, 이미지, 오디오, 비디오등의 멀티미디어 지원 기능 둘째 항해를 위한 Home, Back, Forward, Goto, Reload, Bookmark, HistoryList 등의 컨트롤 기능 셋째 서버설정, 보기옵션 설정등의 환경설정 기능과 일반적 접속기능을 가지고 있다. 이러한 웹브라우저는 HTML브라우저와 HTML엔진으로 나누어지며 HTML브라우저는 GUI와 디스플레이 및 이벤트 처리부분을 담당하고 HTML엔진은 HTTP연결, HTML파싱, 그래픽디코딩 및 전체관리를 담당한다.[4]

2.1 웹브라우저의 Task

일반적으로 웹브라우저의 타스크는 크게 메인작업(Web Browser Task), 연결관리작업(Connection Management Task), 다운로드작업(Download Task), 렌더링작업(Rendering Task)의 4개 과업을 수행한다.



<그림 1> 웹브라우저의 Task

상기 그림은 웹 브라우저의 각 Task 들간의 관계를 나타내고 각각의 역할은 다음과 같다.

① 메인 작업

사용자의 입력 이벤트를 처리하고, HTML문서를 파싱 하면서 문서내에 서버로부터 전송 받은 데이터가 있을

적재되는 것이 아니라 필요한 경우에만 동적으로 메모리에 올라온다. 메모리에 올라온 DLL은 다른 프로그램에서 참조 가능하며 메모리의 효율적인 사용을 제공하게 된다.

3) SMTP, Telnet, HTTP와 같은 프로토콜에서 사용된다.
 4) HTTP(HyperText Transfer Protocol)와 FTP(File Transfer Protocol)를 제공하는 컨트롤이다.
 5) Microsoft Internet 컨트롤이라고도 하고 IE(Internet Explorer)를 설치할 때 등록되는 컨트롤이다.
 6) 정보은닉(Information Hiding)이라고도 하며 내부의 상세한 정보를 공개하지 않고 추상화된 컴포넌트를 실행하는 것을 말한다.
 7) 하나의 클래스가 다른 클래스의 속성을 물려받아 사용할 수 있는 것을 의미한다. 이때 상속되어지는 클래스를 베이스(base) 클래스라 하고 상속받는 클래스를 파생(derived) 클래스라 한다.
 8) 다양한 형태(many forms)라는 의미로 기능적으로 서로 다른 두 개 이상의 함수를 하나의 이름으로 사용할 수 있도록 하는 기법이다. 일종의 컴포넌트간의 Interface 기법이다.

9) 무선웹브라우저나, TV웹브라우저가 아닌 일반 PC용 웹브라우저를 말한다.

경우 데이터의 다운로드를 연결관리작업에게 요구한다. 또 시스템 전반적으로 과업들을 관리한다.

② 연결관리작업

웹 서버와 효율적인 자원의 이용 및 응답대기 시간 최소화를 고려한 TCP 연결설정에 관련된 전반적인 일을 수행한다.

③ 다운로드작업

연결관리작업에서 설정된 연결을 통하여 서버로부터 응답메시지를 기다리며 전송이 완료되면 렌딩작업에게 출력을 요구하고 웹서버 응답대기 시간테이블을 수정한다.

④ 렌딩작업

다운로드작업으로부터 출력 요구를 받으면 해당 데이터를 렌딩하여 화면에 출력한다.[5]

2.2 익스플로러의 구조 개관

인터넷익스플로러는 컴포넌트들과의 연결을 위해 마이크로소프트의 ActiveX컨트롤이나 활성문서접속기능(Active Document Interfaces)등을 사용한다. 아래의 그림에서 IExplore.exe는 최고상위계층에 존재하는 작은 응용프로그램으로 익스플로러가 로드될 때 실행되며 인터넷 익스플로러의 컴포넌트 프로그램을 이용하여 브라우저의 항해(Navigation), 항해기록(History Maintenance), 즐겨찾기(Favorites Maintenance), HTML문서분석(Parsing)과 출력(Rendering)등의 작업을 수행하게 한다. 그리고 이 프로그램은 브라우저의 프레임과 툴바 등의 브라우저의 외관상의 모양을 사용자에게 나타내게 한다. 전술한 바와 같이 IExplorer.exe는 Shdocvw.dll 컴포넌트 파일을 직접 호출한다. 그러면 Shdocvw.dll 파일은 MShtml.dll 컴포넌트 파일과 Microsoft Office 응용프로그램과 같은 다른 활성문서(Active Document) 컴포넌트파일들을¹⁰⁾ 호출한다. Shdocvw.dll은 항해(Navigation), 링킹, 즐겨찾기, 항해기록관리등과 같은 기능을 제공한다. 이DLL(Dynam-

ic-Link Library)파일은 ActiveX 컨트롤처럼 호스트에 독립적으로 호출될 수 있는 호스트인터페이스 기능을 제공한다. Shdocvw.dll 컴포넌트 파일을 통상적으로 WebBrowser 컨트롤 이라고 부른다.

내부링크(In-Place Linking)는 같은 웹브라우저컨트롤 하에서 새로운 HTML 문서를 링크를 이용하여 호출할 수 있으나 만약 Mshtml.dll 파일만을 호출하였을 경우에는 링크는 새로운 다른 웹브라우저를 호출하게 된다. Mshtml.dll 파일은 인터넷익스플로러 4.0이상에서 HTML 구문을 분석하여 화면에 출력하는 컴포넌트를 말한다. 이는 동적 HTML 객체모델을 통하여 HTML문서를 출력한다. 이 컴포넌트는 스크립트엔진, 마이크로소프트가사 머신, ActiveX컨트롤, 프로그래밍과 적재된 HTML문서에서 참조하는 다른 객체들을 호출하는 역할을 한다. Mshtml.dll 파일은 활성문서서버접속(Active Document Server Interfaces)기능들을 실행시켜 자신이 표준컴포넌트객체모델(COM: Component Object Model)을 통하여 호출될 수 있도록 하는 기능을 제공한다. 이것은 흡사 OLE형태의 구조(OLE-based Architecture)로 ActiveX컨트롤에 의해 통상적으로 사용되는 여러 속성들은 또한 인터넷 익스플로러 컴포넌트에서도 사용될 수 있다. 이러한 방법으로 웹브라우저컨트롤의 사용은 적재된 문서에서 호출하는 모든 프레임과 컨트롤들을 거를 수 있는 여러 가지 속성을 설정할 수가 있게 한다.[6]

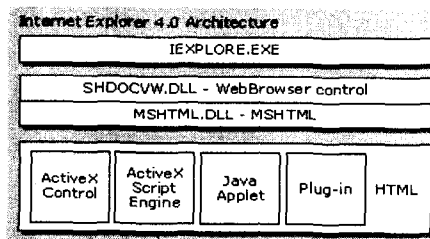
2.3 WebBrowser 컨트롤의 기능

웹브라우저 컨트롤은 그 자체로 웹 브라우저의 거의 모든 핵심 기능을 전부 제공하고 있는 엔진과 같다. 이 컨트롤을 사용하면 시스템에 설치되어 있는 인터넷 익스플로러 4.0을 사용하는 것과 마찬가지다. 아래에서 웹브라우저 컨트롤이 가지는 여러 가지 속성과 각종 메소드와 자주 사용하는 이벤트들을 보면 다음과 같다.

<표 1> 웹브라우저 컨트롤의 주요 속성, 메소드 와 이벤트

속성	LocationName LocationURL Busy Container Resizable ToolBar Visible Height, Left, Top 등
메소드	GoBack, GoForward, Stop, Refresh, GoHome, GoSearch, Navigate Navigate2 ShowbrowserBar GetProperty ExecWB ClientToWindow 등
이벤트	BeforeNavigate DownloadBegin DownloadComplete NavigateComplete ProgressChange StatusTextChange TitleChange FileDownload PropertyChange NavigateComplete2 NavigateError 등

이상과 같은 여러 속성과 메소드와 이벤트들은 익스플로



<그림 2> 인터넷익스플로러의 구조도

10) 사용자가 특정유형의 문서를 열거나 할 때 브라우저에 적재되는 응용프로그램을 말한다.

러의 버전이 업데이트되면서 계속 새로운 것들이 추가되고 있으며, 이러한 메소드와 이벤트를 알면 쉽게 웹브라우저를 만들 수 있다. 즉 프로그래밍 경험이 많지 않은 사람이라도 웹브라우저 컨트롤을 사용하여 웹브라우저를 만들 수 있다. 웹브라우저 컨트롤은 웹브라우저를 완전히 추상화된 하나의 객체로 제공하고 있기 때문에 웹브라우저가 HTML문서를 구체적으로 어떻게 분석하고, 어떻게 화면에 표현하는지에 대해서는 알 수가 없다.

3. 웹브라우저 프로그램 디자인

본 연구에서 제작할 웹 브라우저 프로그램의 디자인은 마이크로소프트의 인터넷 익스플로러와 유사하며 웹 브라우저의 프로그램 기능 또한 마이크로소프트 익스플로러의 기능과 유사하다. 이하에서 Visual Basic을 이용해서 브라우저를 개발할 때 필요한 기본적인 요건과 디자인 개요를 보면 다음과 같다.

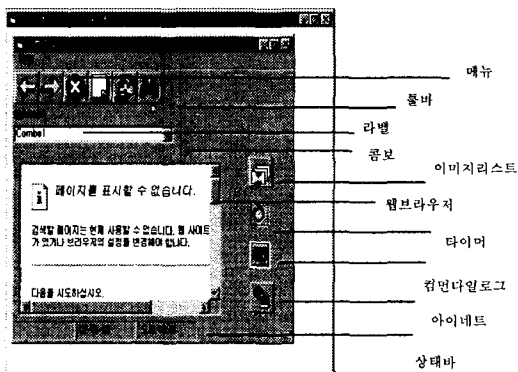
3.1 개발을 위한 기본 요건

웹 브라우저를 개발하는데 필요한 기본적인 작업요건은 다음과 같다.

- 1) 모델 또는 직접적 인터넷 접속
- 2) Winsock으로 알려진 Windows TCP/IP 스택¹¹⁾
- 3) Microsoft Internet Explorer 4.0 이상이 설치되어 있어야 한다.[7]

3.2 폼 디자인

상기와 같은 기본적인 실험 환경이 설정이 되면 이제 브라우저의 기본 폼을 디자인하게 되는데 여러개의 폼을



<그림 3> 폼 디자인

11) 일반적으로 학교에서 인터넷을 바로 접속할 경우는 이 요건을 충족하고 있고 집에서 전화접속을 할 경우 마이크로소프트의 인터넷연결마법사를 이용하게 된다.

생성하여 이를 연결 사용할 수도 있으나 본 연구에서는 하나의 폼만을 사용하여 모든 프로그램을 만들려고 한다. 우선 폼에 들어갈 주요디자인 컨트롤을 보면 메뉴, 툴바, 콤보박스, 웹브라우저, 상태 바 등이다. 메뉴는 메뉴 에디터를 통하여 메뉴 구성표를 참조하여 메뉴를 디자인하며 툴바는 이전페이지, 다음페이지, 불러오기 정지, 새로 고치기, 홈페이지로 가기, 찾기 페이지로 가기 등의 툴바 버튼을 가지고 있다. 읽은 페이지를 나타내는 ComboBox는 불러온 페이지의 리스트를 나타내 준다.

또한 웹문서를 보여주기 위한 웹브라우저 컨트롤을 폼의 중앙에 위치해 놓는다. 이 컨트롤들은 모두 기본적으로 컨트롤 도구 상자에 나타나 있는 것은 아니므로 구성요소 다이얼로그 박스를 열어서 추가해 주어야 한다. 웹브라우저 컨트롤은 Visual Basic의 도구상자의 구성요소 중 Microsoft Internet Controls인 SHDOCVW.DLL 파일을 가져와야 하며 그 외에 파일 열기 메뉴에 사용할 CommonDialog 박스를 가져오기 위해서는 구성요소의 Microsoft Common Dialog Control인 COMCTL32.OCX 파일이 필요하다. 그리고 툴바의 컨트롤 이미지에 사용하는 ImageList 컨트롤, 프로그램의 상황을 나타내는 Status Bar 컨트롤 등을 가져와야 한다. 타이머 컨트롤은 웹 문서를 읽어오기까지의 시간 동안 상태 바에 문서 읽는 중 표시를 해주기 위하여 사용하는데 이것은 기본 도구상자에서 가져 올 수가 있다. 도구 상자 컨트롤과 상태 바 컨트롤은 모두 한 가지 항목에 함께 묶여 있다. 따라서 각각 별개의 항목을 추가하는 것이 아니라 하나만 추가하면 도구 상자에 모두 추가된다. 이것은 구성요소 다이얼로그박스에서 Microsoft Windows Common Controls 6.0 항목을 체크하여 추가하면 된다. 상태바 컨트롤의 사용은 상태바의 두 번째 패널에서 현재 날짜를 표현하기 위하여 세 번째 패널에서 시간을 표현해 주기 위해 불러온 것이다. 이미지리스트 컨트롤은 툴바 컨트롤과 연계하여 툴바의 버튼에 그림을 표현하기 위해 불러온 것이다. 보통 툴바에 버튼을 표시하는 방법으로는 픽처박스위에 여러 이미지컨트롤을 버튼대신에 사용하여 그 위에 해당 그림 아이콘을 올려놓고 마치 툴바의 버튼과 같이 사용하거나 아니면 여러 커맨드버튼을 픽처박스위에 올려놓고 해당 그림 아이콘을 텍스트 대신에 사용하여 그 용도를 표시하여 툴바의 버튼과 같이 사용할 수가 있다. 이러한 디자인 옵션은 설계자의 계획과 사용자의 용이성을 고려하여 변경 제작할 수가 있다. 웹의 내용이나 적정사이트의 URL을 체크하기 위해 Internet Transfer Control(MSINET.OCX)을 사용한다. 이 컨트롤은 인터넷에서 가장 많이 사용하는 FTP나 HTTP 프로토콜을 다룰 수가 있다. 즉 FTP나 HTTP를 통한 파일의 다운로드 또는 파일의 체크등을 행할 수가 있다. 이는

전술한 바와 같이 파일의 내용을 InStr 함수나 Fuzzy 탐색 등을 통하여 이에 대한 접근을 제어하는 특수목적의 브라우저를 개발할 때 사용하기도 한다. 모든 컨트롤을 어떻게 사용하는지를 보여주는 것은 생략하고 대신 아래와 같이 각 컨트롤의 조정 속성을 보여주고 이를 대체하고자 한다.

<표 2> 컨트롤의 속성 지정

컨트롤의 유형	이름	속성	설정값
WebBrowser	WebBrowser	-	기본값
ToolBar	ToolBar1	Align ImageList Button	1-위맞추기 ImageList1 1:Back, 2:Forward, 3:Stop, 4:Refresh, 5:Home, 6:Search
PictureBox	picAddress	Align BordersStyle	1-위맞춤 0-없음
LabelBox	lblAddress	Caption	&Address
ComboBox	Combo1	Text	Combo1
ImageList	ImageList1	Index	1:Back, 2:Forward, 3:Stop, 4:Refresh, 5:Home, 6:Search
Timer	Timer1	Enabled Interval	False 5
CommonDialog	CommonDial	-	기본값
Inet	Inet1	-	기본값
StatusBar	StatusBar1	Panel	2: Style-6 sbrDate, 3: Style-5 sbrTime

3.3 메뉴와 도구 상자의 제작

1) 메뉴의 제작

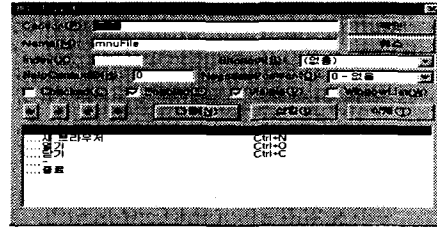
폼에 위치하는 메뉴의 구성은 다양하게 만들 수 있으며 그 서브 메뉴 또한 여러 구성요소를 가지게 할 수가 있다. 우리가 만들고자 하는 파일기능 외에 편집, 즐겨찾기, 도구, 도움말 기능 등을 추가할 수 있으며 이러한 기능의 추가는 VB응용프로그램매사를 이용하여 간단하게 작성할 수도 있다. 본 연구에서는 하나의 주메뉴에 4개의 서브메뉴를 가지도록 제작되었다.

<표 3> 메뉴 디자인

캡션	이름	단축키	설명
파일	mnuFile		
...새 브라우저	mnuNew	Ctrl + N	'새 브라우저를 다른 창으로 열기
...열기	mnuOpen	Ctrl + O	'파일 열기
...닫기	mnuClose	Ctrl + C	'파일 닫기
-	separator		
...종료	mnuExit		

계획된 메뉴디자인은 <그림 4>와 같은 메뉴편집기를 이용하여 이를 폼에 배치할 수가 있다. 메뉴편집기에서는

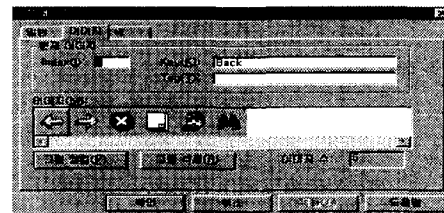
캡션, 이름, 단축키를 지정할 수 있으나 설명은 덧붙일 수가 없다.



<그림 4> 메뉴편집기의 보기

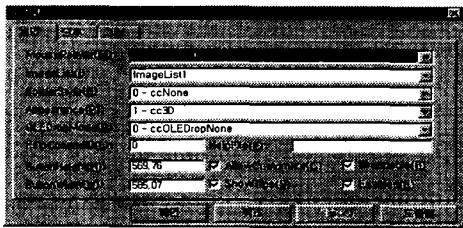
2) 도구 상자의 제작

메뉴를 작성한 후 다음으로 할 작업은 도구 상자에 그래픽을 사용한 버튼들을 만드는 일이다. 사용할 각 버튼의 이미지는 뒤로, 앞으로, 중지, 새로고침, 시작페이지로, 검색페이지로의 6개의 버튼이며 각 그래픽 이미지는 32*32 픽셀의 BMP 파일형식을 취하기로 한 후 ImageList 컨트롤을 연후 속성 창에서 가장 위에 있는 항목인 (사용자 정의) 항목을 선택하여 이미지를 삽입한다. ImageList 컨트롤에는 세 개의 탭이 있는데 <그림 5>에서 보는 바와 같이 가장 먼저 '일반' 탭에서는 리스트에 저장할 그래픽 이미지의 크기 등을 설정한다. 그 다음 '이미지' 탭으로 넘어가 리스트에 포함할 그래픽 파일을 읽어서 실제 목록을 작성한다. Index 값은 반드시 필요하지만, Key 값이나 Tag 값은 필요한 경우에만 사용한다. 마지막으로 '색' 탭에서 색상에 대한 설정을 한다. 이것은 컨트롤에서 사용할 BackColor 속성 값과 MaskColor 속성 값을 조정할 수 있도록 인터페이스를 제공하는 편리한 기능이다. 속성 페이지의 모든 설정이 끝나고 버튼에 사용할 이미지의 배열이 완성되면 이제 도구 상자를 만드는 작업을 할 차례이다. 도구 상자를 만드는 방법은 이미지리스트를 만드는 방법과 유사하며 Toolbar 컨트롤을 선택한 상태에서 속성 창에 있는 (사용자 정의) 항목을 선택하여 다이얼로그 박스를 불러내서 작성한다.



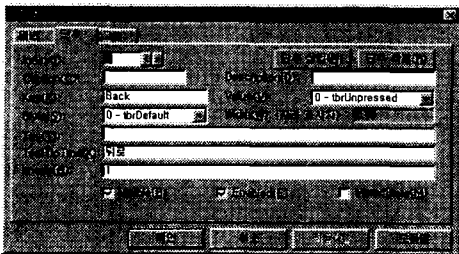
<그림 5> 이미지리스트를 이용한 버튼 삽입의 보기

<그림 6>과 같이 Toolbar 컨트롤의 속성 페이지도 세 개의 탭을 가지고 있다. 처음의 '일반' 탭에는 많은 종류의 항목이 있다. 이 중에서 두 번째 줄에 있는 콤보 박스는 사용할 이미지 리스트컨트롤의 이름을 지정하는 것으로 앞서 작성한 ImageList1 컨트롤의 이름을 선택하면 된다. 그러면 이제부터 Toolbar 컨트롤에서 ImageList1 컨트롤이 가지고 있는 그래픽 이미지들을 사용할 수 있다.



<그림 6> 툴바컨트롤의 사용자 정의 창의 보기

그 다음 <그림 7>과 같은 '단추' 탭에서는 '단추 삽입' 버튼을 눌러서 새로운 버튼을 추가하면 폼에 배치된 도구 상자에 버튼이 생성된다. 원하는 버튼을 만들기 위해서 조정해야 할 것은 Image 항목 번호와 ToolTipText 내용이다. Image 항목의 번호를 ImageList 컨트롤이 가지고 있는 이미지중 하나의 번호를 기재하여 현재 버튼의 표면에 해당 그래픽의 이미지가 나타나게 한다.



<그림 7> 툴바 속성페이지의 단추창의 보기

그 다음, ToolTipText 항목에 각 버튼의 기능을 설명할 수 있는 간단한 문자열을 넣고 이 버튼 위에 마우스를 가져가면 나타나는 풍선 도움말을 넣는다. 이대로 계속 모든 버튼을 추가해 나가면 된다. 도구 상자에 다른 기능의 버튼을 추가해도 된다. 만들기에 따라서 사용 웹 브라우저만큼 보기 좋고 다양한 기능을 가진 형태로 만들 수 있을 것이다.

4. 프로그램의 코드 작성

웹브라우저의 디자인이 완료되고 난 후 웹브라우저를 동작하도록 할 프로그램은 그리 많지 않은 코드만으로도 웹 브라우저를 만들 수 있다. 코딩에서 필요한 가장 먼저 할 일은 Form의 Load시에 수행할 작업을 명시하는 프로시저로 프로시저 작성 이전에 필요할 경우 변수 설정을 먼저 할 수도 있다. 본 프로그램에서는 처음 시작할 URL의 위치를 나타내는 변수 StartingAddress와 웹 브라우저의 파일을 읽는 중이거나 Navigate 중일 때 사용하는 함수 mbDontNavigateNow를 Public StartingAddress As String, Dim mbDontNavigateNow As Boolean 으로 선언해 준 후 폼로드 프로시저를 아래의 <표 4>와 같이 선언해 준다.

<표 4> 프로시저와 프로그램코드

이벤트/프로시저 이름	사용 예소드	프로그램 코드
Load 프로시저	Show Resize Enabled Navigate-등	<pre>Private Sub Form_Load() StartingAddress = "http://kr.yahoo.com/" On Error Resume Next Me.Show Toolbar.Refresh Form_Resize Combo1.Move 50, lblAddress.Top + lblAddress.Height + 15 If Len(StartingAddress) > 0 Then Combo1.Text = StartingAddress Combo1.AddItem Combo1.Text 'starting address 찾기시작 Timer1.Enabled = True WebBrowser1.Navigate StartingAddress End If End Sub</pre>
Resize 프로시저	Width Height	<pre>Private Sub Form_Resize() Combo1.Width = Me.ScaleWidth - 100 WebBrowser1.Width = Me.ScaleWidth - 100 WebBrowser1.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height + StatusBar1.Height) - 100 End Sub</pre>
Timer 프로시저	Busy Enabled LocationName	<pre>Private Sub Timer1_Timer() If WebBrowser1.Busy = False Then Timer1.Enabled = False StatusBar1.Panels(1).Text = WebBrowser1.LocationName Else StatusBar1.Panels(1).Text = "읽는 중..." End If End Sub</pre>

다음으로 폼의 Resize 이벤트 프로시저 부분을 만드는 것으로 여기에서는 웹브라우저 컨트롤의 크기만 조정해 주면 된다. 웹브라우저 컨트롤은 URL일 경우 문자열이 "http://"로 시작되어야 하며 파일일 경우 "File://"로 시작되어야 한다. 이는 웹브라우저 컨트롤이 마이크로소프트 인터넷 익스플로러와 비슷하기 때문이다. 시작주소로 URL, "http://kr.yahoo.com"으로 지정되었고 콤보 박스 컨트롤과 라벨박스의 위치를 고정해 주고 찾은 URL의 History를 더해 주는 절차를 폼로드 이벤트 프로시저

에 더해 준 후 검색 중임을 표시하기 위하여 Timer의 작동을 시작하며 이를 위한 Timer 프로시저는 위의 코드와 같다. 타이머 컨트롤이 가동되면 웹브라우저는 콤보박스 컨트롤에서 선택된 파일을 Navigate 하게 된다. 이를 위한 Navigate 이벤트 프로시저는 아래의 <표 5>코드와 같다. 타이머 컨트롤이 작동될 때 웹브라우저가 작동 중이면 웹브라우저의 Busy 속성이 True 일 때 상태바의 첫 번째 패널의 텍스트를 “읽는 중...”이라 표시하고, 그렇지 않을 경우 웹브라우저에 의해 찾고(Location) 있는 파일명을 출력해 준다.

<표 5> 프로시저와 프로그램코드 계속

Navigate 프로시저	LocationName LocationURL	<pre>Private Sub WebBrowser1_NavigateComplete2(ByVal pDisp As Object, URL As Variant) Dim i As Integer Dim bFound As Boolean StatusBar1.Panels(1).Text = WebBrowser1.LocationName For i = 0 To Combo1.ListCount - 1 If Combo1.List(i) = WebBrowser1.LocationURL Then bFound = True Exit For End If Next i mbDontNavigateNow = True If bFound Then Combo1.RemoveItem i End If Combo1.AddItem WebBrowser1.LocationURL, 0 Combo1.ListIndex = 0 mbDontNavigateNow = False End Sub</pre>
Download Complete 프로시저	LocationName	<pre>Private Sub WebBrowser1_DownloadComplete() On Error Resume Next StatusBar1.Panels(1).Text = WebBrowser1.LocationName End Sub</pre>

다음으로 웹브라우저의 콤보 박스의 KeyPress 프로시저에서 입력된 키가 Return 키라면 콤보 컨트롤의 Text 속성에 지정된 주소지로 이동해서 웹 페이지를 읽어오게 되며 그렇지 않으면 시작주소를 읽어오게 된다. 웹브라우저의 Navigate가 완료되면 상태바의 첫 번째 패널에 방문한 주소를 표시한 후 콤보 박스의 주소에 이를 추가해 준다. 이를 위한 코드는 Combo1의 프로시저중 KeyPress 와 Click 이벤트의 경우를 위한 코드는 <표 6>과 같다.

<표 6> 프로시저와 프로그램 코드 계속

Combo1 프로시저	Click	<pre>Private Sub Combo1_Click If mbDontNavigateNow Then Exit Sub Timer.Enabled = True WebBrowser.Navigate Combo1.Text End Sub</pre>
	KeyPress	<pre>Private Sub Combo1_KeyPress(KeyAscii As Integer) On Error Resume Next If KeyAscii = vbKeyReturn Then Combo1_Click End If End Sub</pre>

이제 메뉴들이 명령들을 동작시키기 위해서는 아래의 <표 7>의 프로시저들이 각각 필요하게 된다. 이것은 툴바의 경우 여러 버튼들이 있지만 이를 하나의 컨트롤로 생각하는 것과는 사뭇 다른 점이다. 우선 새브라우저 프로시저에서는 frmB 라는 변수명을 선언해 준 후 개체를 생성할수 있도록 As New 메소드를 더해준다. 다음으로 열기 메뉴에서는 디스크의 웹 문서 파일을 읽어 웹브라우저 컨트롤에 넣어 주는 역할을 하는데 CommonDialog 박스로 파일을 연 후 파일의 필더기능을 통하여 HTML 파일과 HTM 파일만 읽어와서 웹브라우저에게 넘겨주게 된다. 파일메뉴중 닫기는 폼을 닫는(Hide) 역할을 하는데 폼이 여러개인 경우 원하는 폼만 닫으면 된다. 종료메뉴는 End 메소드를 사용하여 프로그램을 종료하면 된다.

<표 7> 프로시저와 프로그램 코드 계속

새브라우저 프로시저	Show	<pre>Private Sub mnuNew_Click() Dim frmB As New frmBrowser frmB.StartingAddress = "http://kr.yahoo.com/" frmB.Show End Sub</pre>
열기 프로시저	Filter FileName Navigate	<pre>Private Sub mnuOpen_Click() Dim sFile As String With CommonDialog1 .Filter = "HTML File(*.html)*.htmlHTML File(*.htm)*.htm " .ShowOpen If Len(.FileName) = 0 Then Exit Sub End If sFile = .FileName End With WebBrowser1.Navigate "file://" + CommonDialog1.FileName End Sub</pre>
닫기 프로시저	Hide	<pre>Private Sub mnuClose_Click() Me.Hide End Sub</pre>
종료 프로시저	End	<pre>Private Sub mnuExit_Click() End End Sub</pre>

도구상자의 버튼들이 제대로 동작하려면 아래와 같은 <표 8>의 코드가 필요하다. 도구상자는 비록 많은 버튼들을 가지고 있지만 결국은 하나의 컨트롤로 각 버튼들에 대한 이벤트는 한꺼번에 모아서 같은 이벤트프로시저에서 처리한다. 툴바 컨트롤에 있는 버튼을 클릭하면 버튼클릭이벤트가 발생한다. 버튼의 Index 값이 1 이면 Back 메소드를 호출하고, 값이 2 라면 두 번째 버튼이 눌러진 것이므로 Forward 메소드가 호출된다. 마찬가지로 값이 3 이면 Refresh, 4 인 경우 Home, 5 인 경우 Search, 6 인 경우 Stop 메소드를 호출한다. 웹브라우저 프로그램의 코드 작성을 웹 브라우저의 실행 순서에 맞추어 작성하여 설명하였다. 이의 실행을 보면 아래와 같다.

<표 8> 프로시저와 프로그램코드 계속

```

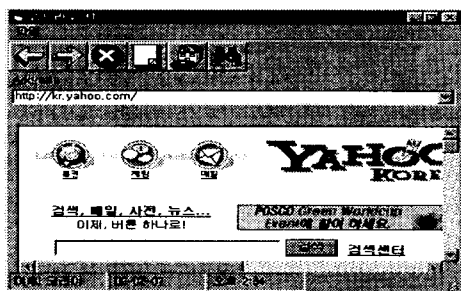
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
On Error Resume Next
Timer1.Enabled = True
Select Case Button.Key
Case "Back"
WebBrowser1.GoBack
Case "Forward"
WebBrowser1.GoForward
Case "Refresh"
WebBrowser1.Refresh
Case "Home"
WebBrowser1.GoHome
Case "Search"
WebBrowser1.GoSearch
Case "Stop"
Timer1.Enabled = False
WebBrowser1.Stop
StatusBar1.Panels(1).Text = combo.Text
End Select
End Sub
    
```

5. 프로그램의 실행

웹브라우저는 전술한 바와 같이 VB응용프로그램마법사를 이용하여 간편 제작할 수도 있고 상기와 같이 직접 자신이 디자인하여 프로그램코드를 작성한 후 이를 실행시킬 수도 있다. 웹브라우저를 실행하기 위한 기본적인 작업요건은 우선 모뎀 또는 직접적 인터넷 접속과 Winsock으로 알려진 Windows TCP/IP 스택과 Microsoft Internet Explorer 4.0 이상이 설치되어 있어야 한다. 보통은 가정에서 PPP접속을 통하여 TCP/IP에 연결이 되고 학교에서는 LAN을 통하여 바로 연결된다. PPP접속이 되어 홈페이지를 연결하면 <그림 8>과 같은 그림이 나타날 것이다.

5.1. 범용웹브라우저의 실행

마이크로소프트 비주얼 베이직은 그 자체가 인터넷 익스플로러가 설치되어야만 설치가 가능한 프로그램이다. 따라서 모든 비주얼 베이직 환경에서는 인터넷 익스플로러 컨트롤을 사용할 수 있다. WebBrowser 컨트롤은 인



<그림 8> 웹브라우저의 실행 사례

터넷 익스플로러의 껍질만 벗겨버리고 핵심 엔진만 남은 알맹이와 같다고 말할 수 있다. 컨트롤에게 '어느 주소로 가라'라든지 '검색 페이지로 등록된 홈페이지로 가라'라는 식으로 단순한 명령만 내려주면 된다. 모든 세부적인 동작은 컨트롤 객체 안에 캡슐화 되어 사용자에게 숨겨져 있다. 이러한 컨트롤을 사용하면 쉽게 웹브라우저를 만들 수는 있지만 자세한 내용은 전혀 알 수가 없으므로 컨트롤을 사용한 프로그래밍의 한계점이 여기에 있다 하겠다.

프로그램을 실행하면 되면 <그림 3>의 화면에서 <그림 8>의 화면으로 바뀌게 된다. 첫 번째의 홈페이지가 "야후 코리아"이므로 이곳으로 이동하게 된다. 폼에서 [URL] 영역에 방문하고 싶은 사이트의 URL을 입력하여 보면 정확하게 입력한 사이트의 홈페이지로 연결된다. 우리가 기존에 사용하고 있던 웹 브라우저에 비해 기능은 미약하지만 똑 같이 웹사이트를 방문하고 [앞으로], [뒤로] 또한 [홈으로] 이동이 가능한 웹 브라우저임을 알 수 있다. 프로그램의 이해를 높이기 위해 다른 여러 가지의 메뉴 기능들을 포함시키지는 않았지만 그 모양이 인터넷 익스플로러와 유사함을 알 수 있다. 그리고 VB응용마법사를 이용하여 간단히 제작한 웹브라우저의 모양과 실행 결과는 <그림 9>와 같다. 프로그래밍 CASE 툴과 같이 프로그램코드, 폼모양, 컨트롤 삽입 등은 마법사에 의해 자동으로 생성 삽입되므로 간단하게 작성 이용할 수가 있다.



<그림 9> VB응용마법사를 이용한 웹브라우저 제작 및 실행 사례

5.2 특수목적웹브라우저의 실행

상기의 웹브라우저를 개발한 후 자기에게 맞는 특수목적용 웹브라우저를 개발할 수도 있다. 예를 들면 청소년들의 유해사이트를 차단하는 정보윤리용 브라우저를 개발하고자 할 경우 등에 있어서 프로그램코드의 약간의 변경을 통하여 이를 만들 수가 있다. 물론 모양 3와12)

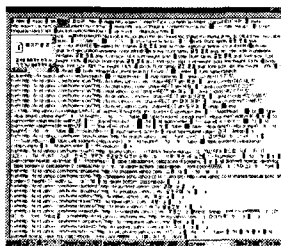
같은 전문유해차단 프로그램을 사용할 수도 있고 인터넷 익스플로러에서 제공하는 유해정보차단 기능을¹³⁾ 이용할 수도 있다. 그리고 HTTP프락시서버 레벨에서 유해 정보를 차단할 수도 있다. 필터링 방법은 주소차단, 단어차단, 접근가능사이트를 제외한 차단, 카테고리차단 등 여러 가지가 있다. 이들 차단기능을 요약해보면 첫째 부적절한 사이트의 URL주소를 차단하거나 둘째 페이지의 내용을 체크할 수도 있다. 이 두가지 방법은 웹브라우저콘트롤에게 이 페이지를 디스플레이하라고 요청하기 전에 수행되어야 하며 Internet Transfer Control(MSINET.ocx)를 사용하여 이를 수행할 수 있다. 이 컨트롤은 인터넷에서 가장 많이 사용하는 FTP(file transfer control)와 HTTP(hypertext transfer protocol)를 다루는 컨트롤로 이를 실행하는 방법은 폼에다 이 컨트롤을 구성요소에서 호출하여 추가하고 텍스트박스의 입력라인을 멀티플로 바꾸어 준 후 아래의 코드를 입력하여 실행시키면 웹사이트를 <그림 10>과 같이 HTML의 원시코드로 볼 수 있다.

<표 9> 인터넷트랜스퍼컨트롤의 기본이용 코드

```
Sub Form_Load( )
Show
WindowState = 2
Text1.Font.Size = 12
Text1.Width = ScaleWidth
Text1.Height = ScaleHeight
Text1.Text = Inet1.OpenURL("http://www.yahoo.com")
End Sub
```

상기코드는 이후의 메인페이지를 HTML의 원시코드로 보기 위한 것인데 이것은 특수목적용 웹브라우저의 재구에 이용할 수 있다.

<그림 10> 인터넷트랜스퍼컨트롤을 이용한 웹사이트의 HTML원시코드보기



- 12) 학부모정보감시단홈페이지(http://www.cyberparents.or.kr/campaign/campaign3_3.htm)에 접속하면 청소년 유해 정보 차단 프로그램 '모야 3'을 다운로드 하여 유해사이트 차단, 컴퓨터 사용 시간 관리, 유해사이트 신고, 모야 최선 DB업데이트 등의 환경을 설정한 후 익스플로러등의 웹브라우저를 실행시키면 된다.
- 13) 인터넷 익스플로러의 [도구] -> [인터넷 옵션] -> [내용관리자]에서 등급 수준 범주에 의한 차단과 승인사이트와 비승인사이트를 직접 입력하여 이를 차단하는 두가지 방법을 이용할 수 있다.

그렇게 할려면 우선 단순히 HTML코드를 텍스트박스(Invisible Text Box)에 저장하는데 텍스트박스의 속성을 Invisible로 설정하여 화면에 보이지 않게 한 후 만약 어린이들이 보기를 원하지 않는 정보를 가지고 있는 페이지가 있는 경우 이를 InStr이나 Fuzzy 서치를 통하여 체크할 수 있다. 만약 원하지 않는 단어나 구문을 가지고 있을 경우 단순히 그 브라우저에 페이지를 디스플레이하지 않으면 된다. 이를 위한 코드는 아래 <표 10>과 같다.

<표 10> 홈페이지 내용 체크를 위한 코드 사례

```
Private Sub cmdGo_Click( )
On Error Resume Next
txtContent = Inet1.OpenURL(txtAddress.Text)
If Not ContentOK(txtContent.Text) Then
MsgBox "Sorry, that is not a site I want you to visit."
Else
WebBrowser1.Navigate txtAddress.Text
End If
End Sub
```

상기의 코드는 체크하고자 하는 내용을 담고 있는 보이지 않는 텍스트박스(Invisible Text Box)와 ContentOK 체크함수를 작성하여야 이를 이용할 수 있다. 만약 단순히 인터넷트랜스퍼 컨트롤을 사용하지 않고 프로그램만으로도 이를 체크할 수 있는데 그때의 필요한 코드는 아래와 같다.

<표 11> 단순 URL접근 방지 특수목적 프로그램

```
Private Sub Combo1_Click()
Dim a As String
Dim check As Boolean
If mbDontNavigateNow Then Exit Sub
Timer1.Enabled = True
Open "A\Censored.txt" For Input As #1
check = False
Do While Not EOF(1)
Input #1, a
If Combo1.Text = a Then
check = True
End If
Loop
Close #1
If check = True Then
MsgBox "사이트를 방문 할수 없습니다.", "방문권없음"
Else
WebBrowser1.Navigate Combo1.Text
End If
End Sub
```

상기코드는 방문을 원하지 않는 URL주소를 아예 텍스트파일로 만들어서 웹브라우저에서 이 URL주소를 <그림 11>과 같이 입력하면 <그림 12>와 같이 사이트를 방문할 수 없음을 표시하게 된다. 만약 일반 브라우저로 돌아 갈려면 <표 6>의 Combo1프로시저의 Click 이벤트 코드를 <표 11>대신에 입력하면 되고 반대로 특수목적용으로 전환할려면 <표 11>의 코드로 대체하면 된다. <그림 12>는 원하지 않는 사이트를 내부적으로 필터링한



<그림 11> www.playboy.com의 주소 입력사례



<그림 12> 유해 사이트 필터링 사례

후 그 결과만 보여주는 것으로 필터링해야 할 URL주소가 많은 경우 위의 예와 같이 TEXT파일로 만들거나 모야와 같은 전문유해사이트인 경우 URL 주소를 데이터베이스화 하여 일정 기간마다 갱신한 파일을 링크시켜 필터링하기도 한다.

6. 결 론

앞의 연구에서 본바와 같이 WebBrowser 컨트롤은 인터넷 익스플로러의 껍질만 벗겨버리고 핵심 엔진만 남은 알맹이와 같다고 말할 수 있다. 컨트롤에게 '어느 주소로 가라' 라든지 '검색 페이지로 등록된 홈페이지로 가라'라는 식으로 단순한 명령만 내려주면 되며 모든 세부적인 동작은 컨트롤 객체 안에 캡슐화 되어 사용자에게 숨겨진다. 이러한 컨트롤을 사용하면 쉽게 웹 브라우저를 만들 수는 있지만 자세한 내용은 전혀 알 수가 없으므로 이러한 컨트롤을 사용하는 것에 만족한다면 그저 단순한 작업자로 남게 될 위험이 있다. 다시말해서 컨트롤을 사용한 프로그램이라는 것의 한계를 보여주고 있다고 생각할 수도 있다. 하지만 본연구는 웹브라우저의 구조를 이해하기 위해서는 한번쯤 해볼 만한 프로그래밍은 틀림없다. 그리고 인터넷 익스플로러의 버전이 업데이트 될 때마다 새로운 웹브라우저 기능들이 계속 추가 되고 있으므로 사실 단순한 프로그래밍 작업인 것 같지만 이들을 응용해 보기 위한 프로그래밍 작업은 여전히 많은 노력과 참고자료가 필요하게 되며 종합적인 비주얼 프로그래밍 기술의 구사가 필요하게 된다. 본연구는 이러한 면에서 다소의 일조를 하기위해 시작하였으나 웹브라우저의 내부기능의 이해나 보안필터링, 차단, 인증 등의 웹브라우저의 신기능들을 위한 프로그래밍 기법은 살펴보지

못해 제한점이 있다고 하겠다. 그리고 이밖의 각종 윈도우의 컨트롤에 대한 이해도 계속되어야 하겠다.

Windows Common Controls를 사용하면 보다 효율적인 인터페이스를 제작할 수 있고 도구 상자나 탭 컨트롤, 그리고 상태 표시 바 등과 같은 상당히 유용하게 사용될 수 있는 인터페이스 컨트롤을 이해할 수가 있다. 이들 컨트롤에 대해서는 잘 알아두면 틀림없이 상당히 많은 도움을 받을 수 있을 것이다.[2] 차세대 웹이라고 불려질 미래의 웹은 새로운 HTTP프로토콜(HTTP-NG)과 사용자의 서비스에 따라 확장되는 URL, 그리고XML(XHTML)/CSS, SML(Synchronized Multimedia Integration Language), DOM(Document Object(Model), RDF(Resource Description Framework) 등으로 구성되는 사용자 환경으로 구성되며 인터넷 상의 모든 사용자 환경을 대변하는 수단으로 자리매김 할 것으로 보인다. 국내에서도 이를 위해 관련 기술들을 자체 개발하고 이를 통해 기술 축적을 함으로써 새로운 서비스발굴 및 W3C로의 상향식 표준화 작업을 추진해 나가야 할 것이다.[8]

참 고 문 헌

- [1] 서현근, 전상훈, 구자광공저, 「비주얼베이직6.0」, 내하출판사, 2000
- [2] 김원일, 김태한지음, 「Visual Basic 6.0 프로그래밍 마스터」, 정보시대, 1999
- [3] 장영건, 웹브라우저의 보편적 설계와 시각장애인 인터페이스, 정보과학회지 2001년 9월 제19권 제9호(통권 148호) pp.4-11
- [4] 조수선, 김경일, 마평수, "프락시 서버기반의 인터넷 TV용 브라우저 시스템", 한국정보처리학회논문지 2000년 8월 Vol.7, No. 8, pp.2442-2451
- [5] 이동우, 조수선, "웹 브라우저에서 사용자 응답 대기 시간 감소를 위한 설계", 2001년 한국정보과학회 학술발표논문집 Vol. 28, No. 2, pp.517-519
- [6] <http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/ext/overview/overview.asp>
- [7] David I. Schneider, 「An Introduction to Programming Using Visual Basic 6.0」, Prentice Hall, 4th ed., 1999.
- [8] 신명기, 김용진, "W3C에서의 차세대 웹 표준 활동 동향", 정보처리학회지 1996년 5월 Vol.6, No.3, pp.7-17.
- [9] 최영준, 「완벽가이드비주얼베이직5.0」, 영진출판사, 1997.

- [10] 박한주, 서상수, 이필진, 오병균, ISO DSSSL을 기반으로 한 SGML 웹브라우저, 한국정보과학회 가을학술발표논문집 1999년 10월, Vol. 26, No. 2(III), pp.534-536.
- [11] 박우창, 웹브라우저를 이용한 다중 데이터베이스 검색모델, 덕성여대논문집, 1996년 제27집 pp.481-493.
- [12] 이재규, 액티브X 테크니션으로 가는길: 액티브X 스크립트 호스트 제작하기, 마이크로소프트웨어 1998년 7월 pp.344-351.
- [13] 이재규, 웹브라우저컨트롤 SHODCVW, MSHTML 내 것만들기, 마이크로소프트웨어 1998년 8월 pp.298-305.
- [14] 이정숙, 이부권, 서영건, "웹기반의 실시간 원격강의를 위한 서버와 클라이언트간의 웹브라우저 동기화", 정보처리학회논문지, 2001년3월 Vol.8, No.1(A), pp.70-74.
- [15] 조수선, 김경일, 신규상, 실시간 OS용 웹브라우저의 설계, 한국정보과학회 가을학술발표논문집 1999년 10월, Vol. 26, No. 2(III), pp.531-533.
- [16] 정인성, 김태선, 서진호, 김창훈외, 넷스케이프의 폴소스를 분석한다:Mozilla, 마이크로소프트웨어 1998년 6월 p.236.
- [17] 조수선, 김경일, 마평수, 프락시 서버 기반의 인터넷 TV용 브라우저 시스템", 한국정보처리학회논문지 2000년 8월, Vol.7 No.8 pp.2442-2451.
- [18] 한국전산원, "Web브라우저 개발지침서", 연구보고서, 문서번호 NCAIII-RER-9671, 1996년 11월
- [19] Brian Overland, 「Visual Basic 6 in Plain English」, IDG Books 1999
- [20] Gary Cornell, 「Visual Basic 6 From The Ground Up」, Osborne/McGraw-Hill, 1998년, pp.861-890.
- [21] Gary R. Wright and W. Richard Stevens, TCP/IP Illustrated Volume II. Addison Wesley, 1995.
- [22] John Socha, Dan Rahmel and Devra Hall, 「Visual Basic 5」, MIS Press, 1997.
- [23] Loren Eidahl With Michael McMillan, 「Visual Basic 5 Control Creation Starter Kit」, Que Corporation, 1997.
- [24] Michael Halvorson, 「Microsoft Visual Basic 5 Step By Step」, Microsoft Press 1997.
- [25] Robert H. Reid, Architects of the WEB: 1000 Days that Built the Future of Business, John Wiley & Sons, Inc, 1995.
- [26] Rod Stephens, 「Ready -To-Run Visual basic Algorithms」, 2nd ed., John Wiley & Sons, Inc., 1998.
- [27] Rod Stephens, 「Advanced Visual Basic Technologies」, John Wiley & Sons, Inc., 1997 pp.369-405.
- [28] Scott Warner, 「Teach Yourself Visual Basic 6」, Osborne/McGraw-Hill, 1998.
- [29] Tim Patrick, 「The Visual Basic Style Guide」, Microsoft Technologies Series 2000 pp.109-118.
- [30] Vincent Chen, John Montgomery, 「Hacker's Guide To Visual Basic」, Addison-Wesley Developers Press, 1996.
- [31] W. Richard Stevens, TCP/IP Illustrated Volume I: The Protocols, Addison Wesley, 1994.
- [32] Wallace Wang, 「Visual Basic 6 For Dummies」, IDG Books 1998.
- [33] William Stamatakis, 「Microsoft Visual Basic Design Patterns」, Microsoft 2000, pp.9-44.
- [34] http://httpd.apache.org/docs/mod/mod_proxy.htm



우 원 택 (Won Taek Woo)

1979년2월 성균관대학교통계학과학사
 1981년8월 연세대학교경영대학원석사
 1991년2월 영남대학교상경대학박사
 1993년3월~현재경산대학교정보과학부

관심분야 : 행동MIS, 지식전자상거래, 경쟁우위관리, 고객관계관리, 공급망관리, 전략적엔트프로노 싯 등