

학습 활동 추적을 위한 HTTP 기반 확장 프로토콜 개발

박종오[†]

요 약

웹의 구조적인 특성으로 인해서 실제 원격 교육 시스템으로 교육 활동을 수행할 때 많은 제한점이 발생된다. 비연결성 프로토콜인 HTTP는 클라이언트의 요청 각각을 별도로 수행하며 연결 상태를 유지하지 않는다. 이것은 학습자의 접속상황을 유지 관리하고 학습자의 요청 정보를 추적하기 어렵게 만든다. 그리고 이는 한 학습자의 연결유지만 아니라 다른 학습자들간의 상호작용 자체를 불가능하게 만든다. 본 연구는 학습자의 학습 활동을 추적할 수 있도록 하기 위해서 HTTP를 기반으로 하여 확장된 연결성 프로토콜인 CHTP를 개발하고, 새로운 원격교육 플랫폼을 구축하였다. 프로토콜 차원에서 표준화된 방식을 제안하기 때문에 개발된 웹 확장 플랫폼은 원격교육을 위한 시스템 구축을 효과적이고 용이하게 만들어 줄 것이다.

Development of HTTP-based extension Protocol for Tracking Learning Activities

Jong-O Park[†]

ABSTRACT

In the distance education systems, there are limited things while performing educational activities because there are to be problems in structural features of the Web. HTTP, a connection-less protocol, performs requests of client, however, does not hold on the status. Thus, by the features of the Web, it is difficult to hold on the connection of learners and trace information asked by learners. Moreover, these problems make impossible not only a learner's connection continuity but also on-line interaction among the learners in the distance education. This thesis developed CHTP, an connection-based hypertext transfer protocol, based on HTTP and a new platform of distance education in order to track activities of learners. The developed web extension platform will make it easy to build up system for being helpful in a distance education because this thesis proposes a standardized way in the protocol.

1. 서 론

원격교육을 위한 기술적인 지원 수단으로써 이용되는 현재의 웹 플랫폼은 이전의 통신기술보다 훨씬 효과적이고 유용한 원격교육 환경을 제공한다. 웹이 갖는 특성은 원격 교육의 활성화를 촉진시키고 발전시켰으나, 반대로 웹의 구조적인 특성으로 인해서 실제 원격 교육 시스템으로 교육 활동을 수행할 때 많은 제한점이 발생된다. 그것은 웹의 목적 자체가 불특정 다수를 대상으로 한 공개적인 정보 배포 및 상호간 정보 공유를 목적으로 개발된 플랫폼이기 때문이다. 즉, 웹에서 사용되는 프로토콜인 HTTP와 마크업 언어인 HTML, 그리고 관련 웹 기술 요소들은 그 자

진시키고 발전시켰으나, 반대로 웹의 구조적인 특성으로 인해서 실제 원격 교육 시스템으로 교육 활동을 수행할 때 많은 제한점이 발생된다. 그것은 웹의 목적 자체가 불특정 다수를 대상으로 한 공개적인 정보 배포 및 상호간 정보 공유를 목적으로 개발된 플랫폼이기 때문이다. 즉, 웹에서 사용되는 프로토콜인 HTTP와 마크업 언어인 HTML, 그리고 관련 웹 기술 요소들은 그 자

[†] 정 회 원: 한국교원대학교 컴퓨터교육과 강사
논문접수: 2003년 3월 2일, 심사완료: 2003년 4월 1일

체만으로 원격교육을 위한 기술 도구로서 한계를 갖고 있다.

이 중 웹 프로토콜 측면에서의 구체적인 문제점들은 HTTP가 상태를 유지하지 않는 비연결성 프로토콜임에 기인한다. HTTP는 클라이언트의 요청 각각을 별도로 수행하며 연결 상태를 유지하지 않는다. 이것은 요청하는 자료들이 서로 다른 서버들에 흩어져 있는 분산환경에서는 효율적이지만, 학습 자원이 한 서버에 집중해 있거나 학습자의 접속상황을 유지 관리하고, 학습자의 요청 정보를 추적하기는 어렵게 만든다. 그리고 이러한 문제는 한 학습자의 연결유지만 아니라 다른 학습자들간의 상호작용 자체를 불가능하게 만든다.

이에 본 논문에서 이러한 노력들을 살펴보고 이를 기반으로 하여 연결상태를 유지, 관리하는 웹 확장 프로토콜을 개발한다. 개발된 웹 확장 프로토콜은 기존 웹이 갖는 장점을 그대로 수용하고, 단점을 보완하기 위해 추가적인 세션 지원이 이뤄진다. 이 세션 정보를 이용하여 이용자의 이동 경로를 추적할 수 있고, 이것은 원격교육 시스템에서 학습자 학습 활동을 추적하고 평가 및 피드백을 위한 중요한 정보가 된다.

2. 웹 확장 시도 분석

웹에서 사용되는 프로토콜은 그 특징과 장점으로 다양하게 활용되고 있으나, 원격교육 측면에서는 프로토콜의 근본적인 문제점과 제한점으로 있다. 원격교육의 관점에서 현재 표준의 웹 환경은 사용자 신원을 확인하고 활동을 추적하는 능력에 제한이 있다[9]. 이는 웹 기반의 환경에서 학습자를 관리하는데 중대한 딜레마에 빠지게 한다[11]. 그것은 HTTP가 비연결성(stateless) 프로토콜이기 때문이다. 즉, 서버에 대한 모든 요청은 개별적인 것으로 취급된다. 따라서 지금 요청에 대해서 이전 어떤 요청과 관계 있는지에 대한 정보를 갖고 있지 않다. 특별히 어떤 학생이 연속적인 두 요청을 할 경우 서버 입장에서는 이 사항의 연관관계를 알 수 없다. 더욱이 브라우저는 웹 탐색을 위해 자신을 구분하는 어떠한 정보도 갖고 있지 않고, 이는 사용자가 누구인지 알 수

없게 되며 사용자를 추적할 수도 없다. 그래서 별도의 자바 애플릿이나 ActiveX 컨트롤을 사용하여 HTTP와 다른 TCP 연결 지원하도록 구현되고 있다. 이에 웹 관리 기구인 W3C 및 개인들에 의해서 많은 확장시도들이 있었다.

2.1. HTTP 자체 개선 노력

HTTP 규격이 정해졌을 당시 이전의 정보 검색 및 정보 전달에 관련한 프로토콜을 고려하여 통합된 형태로 브라우징 할 수 있도록 지원한 측면이 있지만, 이후의 기술의 발전과 사용자의 요구 및 필요에 의해 프로토콜 확장을 꾀하는 많은 시도들이 있었다. 이 중 기능상의 확장에 대한 것은 HTTP Extension Framework와 HTTP-NG, PEP 등이 있다.

2.1.1. HTTP Extension Framework

Nielsen 등[10]에 의해서 제안된 HTTP Extension Framework[RFC 2774]는 표준 HTTP 규격을 확장하기 위한 실험적인 시도로서, 기존 프로토콜과의 호환성을 유지하면서 요청 및 응답 시 별도의 정보를 추가하여 확장하고자 하고 있다. 이 실험적인 제안은 HTTP를 지원하는 서버, 서버 컴포넌트, 클라이언트 등이 확장을 유동적으로 하기 위한 지침을 제시하며 응용수준의 규격은 아니다. 기존 프로토콜과 비교하여 주요한 특징은 단일 요청 및 응답에 대한 HTTP 메시지 확장을 들 수 있으며, 기존의 비연결성 구조를 그대로 갖고 있다.

2.1.2. HTTP-NG

현재 사용되고 있는 HTTP/1.1이 기존의 HTTP/1.0의 부족한 부분을 어느 정도 채워주고 있지만, 프로토콜의 구조적인 특성이 변화된 것은 아니다. HTTP/1.0과의 하위호환성을 엄격하게 유지하면서 개선시키는 작업으로는 이 프로토콜 구조의 총체적인 개선을 기대할 수가 없기 때문이다. 따라서 이러한 하위호환성에 대한 압박감에 벗어나 약간의 융통성을 발휘하여, HTTP-NG(HTTP-Next Generation)는 여러 특징을 포함하고 있다.

HTTP-NG은 거대한 규모로 성장을 이끌어온 현재의 HTTP 1.X의 문제점 보완과 상업적인 응용에 대한 요구사항을 충족하기 위해 새롭게 규정하고 있는 차세대 프로토콜 규격을 의미한다. IETF HTTP-NG 워킹 그룹에서 작업을 진행하였으나 현재는 초안수준에서 개발 논의가 중단된 상태이다.

HTTP-NG는 HTTP 1.X의 간편함을 유지하면서 보안이나 인증, 모듈화, DCOM(Distributed Component Object Model) 또는 CORBA와 같은 원격 기능 호출(Remote Procedure Call, Remote Method Invocation) 직접 지원 등의 중요한 사항을 더하고 있는데, 주요한 특징 중의 하나로서 HTTP-NG는 상태유지(state)를 사용함으로써 HTTP 1.X에 비해서 훨씬 더 나은 성능을 갖게 된다는 것이다.

HTTP-NG는 연결이 명백히 닫혀질 때까지 연결을 유지한다. 또한 요청과 응답은 모두 비동기적으로 처리된다. 즉, 브라우저는 1번째 요청이다 끝날 때까지 기다린 뒤 2번째 요청을 보내지 않고 비동기적으로 함께 보낸다. 이와 유사하게 서버도 응답을 순서에 구애받지 않고 보낼 수 있다. 또한 일부 요청을 채널(channel)이라는 개념을 사용하여 동시에 보낼 수도 있다.

2.1.3. PEP

PEP(Protocol Extension Protocol)은 HTTP 프로토콜에 사용자가 직접 설계한 프로토콜을 추가할 수 있게 하는 것이다. HTML이나 URI쪽을 변경 내지 확장하다 보면 HTTP를 확장해야 할 필요가 생긴다. 그러나 프로토콜의 확장이라는 것이 그리 쉽지 않아서, 한쪽만 지원한다고 되는 것이 아니다. 그렇다고 확장 한번 할 때마다 버전 번호를 높일 수도 없는 일이다.

이러한 문제점들에 대한 해결책을 제시하기 위한 시도로서 몇 개의 헤더 이름과 콘텐츠 타입으로서 PEP의 개념을 제안하게 되었다. 이것을 이용함으로써 HTTP 프로그램들은 알려진 확장 기능들이든 알려지지 않은 확장 기능들이든 적절하게 동작할 수 있으며, 양쪽에 가능한 프로토콜 확장 기능을 선택할 수 있고, 특정한 수행 능력에 대해 상대방에서 물어볼 수도 있게 된다.

이를 위해 Protocol, Protocol-Request, Protocol-Query, Protocol-Info 등의 네 가지 추가적인 헤더 필드를 사용한다. Protocol 헤더를 이용하여 확장 기능의 이름, 적용 범위, 중요도, 또는 관련한 데이터 헤더들을 나타낼 수 있고, Protocol-Request 헤더 필드를 이용하여 상대방에게 특정 확장 기능을 사용하게 요청할 수 있으며, Protocol-Query 헤더 필드를 이용하여 상대방에게 특정 확장 기능이 지원되는지에 대해 물어볼 수 있고, Protocol-Info 헤더 필드를 이용하여 어떤 사항들이 지원되는지에 대한 정보를 알려줄 수 있게 된다.

2.2. HTTP 보조 지원 노력

프로토콜이 아닌 어플리케이션 차원에서 HTTP를 보완하는 방법에도 역시 많은 노력이 있었다.

웹 환경에서 연결성을 확보하고자 하는 보조 지원 노력중 대표적인 방법은 Java Servlet 기술을 이용하는 것이다. Java는 플랫폼에 관계없이 Java 가상 머신을 이용하여 구현되기 때문에 어떠한 원격교육시스템이라도 세션 확보를 비롯한 원하는 기능을 부여할 수 있도록 해준다. 또한 서버측 애플릿이기 때문에 클라이언트의 브라우저 환경에 관계없이 운영할 수 있다는 장점이 있다.

다만 이 경우 충분한 고려 없이 이용하게 된다면 해당 시스템 종속적인 기능이 될 수밖에 없고, 호환성을 확보하기 어렵다.

2.3. 학습관리시스템 표준화

사이버공간을 중심으로 교육서비스는 기관 및 전문업체들이 다양한 원격교육 기술들을 바탕으로 개발한 상용 교육서비스 플랫폼에 의해서 서로 다른 형태로 제공되고 있다. 그러나 독자적으로 개발해서 운영하고 있는 교육서비스 플랫폼이 제공하는 서로 다른 양식의 교육서비스는 원격교육 및 서비스 제공자간의 교류 활성화에 걸림돌로서 작용할 가능성이 커지고 있다[5]. 이는 서비스를 위한 플랫폼을 만드는 회사, 이 플랫폼을

이용하여 서비스하는 회사 등의 e-Learning 업계 및 학습자 등 모든 원격교육 관련 주체들이 활용하고 발전시키는데 도움이 되지 않는다.

이렇게 비효율적이고 배타적인 개발 구조를 극복하기 위해서 e-Learning 관련 연구자들은 한번 만들어진 콘텐츠를 재사용하고, 제3자가 만든 콘텐츠를 공유할 수 있는 방법을 모색하게 되었다. 이미 개발된 콘텐츠의 일부 혹은 전체를 재사용하거나 제3자가 만든 콘텐츠를 공유할 수 있는 체계를 구축함으로써 개발에 소요되는 시간 및 비용을 대폭 절약할 수 있기 때문이다. 이러한 노력들이 1997년경부터 미국을 중심으로 이뤄졌고 e-Learning 표준화와 관련하여 현재 상당부분 진행되고 있다.

현재 AICC, IEEE, ISO/IEC, IMS Global Learning Consortium, ADL, ARIADNE 등에서 상호보완적인 원격교육 플랫폼 표준화작업을 진행하고 있으며, 이 작업들은 크게 학습관리시스템(Learning Management System; 이하 LMS)과 콘텐츠관리시스템(Learning Contents Management System) 차원에서 구분된다.

특히 표준화 작업의 완성도가 높은 ADL의 SCORM에서 보면 플랫폼 차원에서의 LMS는 자바 가상머신상에서 AICC의 통신 API를 근거한 표준화 작업을 진행하고 있다. 즉, 학습 활동 추적에 위해서 HTTP로서는 해결할 수 없는 세션 확보를 자바로 구현되고, 이를 API 수준에서 표준화를 시도하고 있다.

3. 프로토폴 설계

3.1. 교육용 웹 플랫폼의 특성 탐색

광역으로 분산된 웹 자원의 정보를 불특정 다수에게 제공하는 웹 방식과는 달리 원격교육을 위한 웹 플랫폼은 몇 가지 다른 차이점을 갖는다. 웹은 원격교육에서 교육 콘텐츠 전송과 상호작용 면에서 그리고 필요한 통신기술의 선택차원에서 웹은 그 어떤 방식보다 효율성과 편리성을 지원하기는 하지만, 원격교육을 위해 생성된 플랫폼이 아닌 만큼 어느 정도의 제한점을 갖고 있다. 이 때문에 원격교육을 위한 기술 표준이 요

구되며, 기존의 웹 플랫폼의 장점을 그대로 유지하면서 원격교육에 맞도록 확장시킬 필요가 있다.

다음은 웹기반 교육 시스템을 구축하기 위한 고려 요소로서, 일반 웹 플랫폼과 다른 교육용 웹 플랫폼의 특성을 정리한 것이다.

첫째, 교육용 웹 플랫폼은 접속하는 사용자에 관심을 둔다. 즉 사용자의 인증과 권한 그리고 그 사용자의 활동 과정에 관심을 둔다. 이를 위한 현재의 프로토폴에서 지원되는 웹 로그, 쿠키 등의 추적관리 기술은 너무 제한적이다. 이런 제약은 탐색학습(exploratory learning) 측면에서 학습자의 행동패턴을 분석하기 위한 도구 개발을 어렵게 하기도 한다.

둘째, 교육용 웹 플랫폼은 진입시점 뿐만 아니라 학습과정의 종료시점까지 관심을 둔다. 이를 통해 학습시간이 결정되고, 학습시간은 평가 및 과정 이수에 영향을 끼친다.

셋째, 교육용 웹 플랫폼은 제공하는 콘텐츠의 권리보호 및 상업적인 이용에 관심을 두고 있다. 비록 상업적인 이용에 관심을 두고있지 않더라도 비인가된 사용자의 무분별한 이용에 부정적이다.

넷째, 교육용 웹 플랫폼은 사용자 상호작용 지원을 중요시한다. 학습중인 시간이거나 아니거나에 관계없이 학습자간, 교수자와의 상호작용을 지원하여 자율학습의 어려움을 해결할 필요가 있다.

다섯째, 교육용 웹 플랫폼은 학습자에게 정보를 밀어주는 푸시(push) 메카니즘을 활용한 능동적인 학습환경을 조성할 필요가 있다. 일반 웹 플랫폼에서 사용하고 있는 풀(pull) 방식은 전적으로 사용자의 노력여하에 따라 교육효과가 좌우되기 때문에 교수자의 능력이 적극적으로 이용되지 못한다.

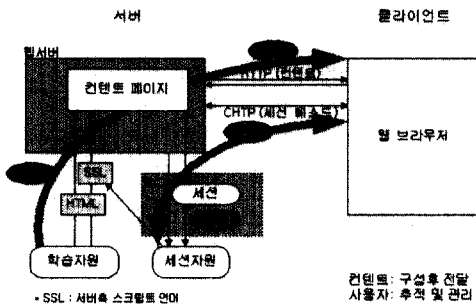
여섯째, 교육용 웹 플랫폼의 웹 페이지는 주로 같은 시스템내에 존재한다. 비록 웹이 분산된 자원을 통합하여 보여주는 분산처리 기술을 갖고 있지만, 교육용 웹 플랫폼은 이러한 자원들이 같은 시스템내에 존재한다.

일곱째, 교육용 웹 플랫폼은 한 페이지에 머무는 시간이 상대적으로 길어 페이지 노출시간이 길다. 일반 웹 플랫폼에서 필요한 정보를 찾기

위한 목적으로 각기 다른 사이트와 페이지들을 향해하는 것과는 달리, 교육용 웹 플랫폼에서는 콘텐츠 및 교수 안내의 내용에 관심을 두고 한 곳에 계속 머물러 있다. 따라서 학습자의 피로도를 고려한 효과적인 페이지 설계 등의 인터페이스 설계가 중요하다.

3.2. 플랫폼 구조

개발되는 전체 플랫폼 구조는 크게 웹 콘텐츠 처리부와 사용자 세션 관리부로 나뉘어진다. 웹 콘텐츠 처리부는 기존의 HTTP 기반의 웹 전송과 같은 구조로 호환성을 갖고 있으므로 기존의 하이퍼텍스트 문서를 그대로 이용할 수 있다. 이 처리부는 HTML를 이용하여 자신의 서버 및 다른 서버의 있는 학습자원들을 연결하여 하나의 콘텐츠 페이지로 구성하고 이를 사용자의 웹브라우저로 전송한다. 사용자 세션 관리부는 세션으로 유지하여 학습자 및 교수자 등의 사이트 이용자들을 최초 접속시부터 사이트 이용 종료시까지의 상황을 추적하며 관리할 수 있도록 지원하는 확장 모듈이다. 이 모듈과 세션 메소드와 서버측 스크립트 언어를 통해서 접속자의 상황을 알 수 있고, 접속자간의 상호작용을 위한 기본 정보를 제공해 준다.



(그림 1) 플랫폼 구조

플랫폼내 구성요소는 콘텐츠 처리와 사용자 세션 관리에 따라 크게 자원(resource)과 활동(activity)으로 구분된다. 자원은 콘텐츠 구성을 위한 학습 자원과 사용자 관리를 위한 세션 자원으로 구분된다. 활동은 콘텐츠 측면에서는 콘텐

츠 구성과 전달로 구성되며, 사용자 관리 측면에서는 사용자 추적과 사용자 세션 관리로 구분되어 처리된다.

이렇듯 두 개의 구조로 분리하는 이유는 기존 웹서버와의 호환성을 확보하고 모듈화를 통해서 개발을 용이하게 하여 전체적인 성능의 향상을 꾀하기 위함이다. 또한 기존의 웹 방식의 사용자 인터페이스를 그대로 이용하여 사용자의 전면부(front-end)에 들으로써 사용의 혼란을 방지하도록 하였다.

3.3. 프로토콜 구조

사용자의 세션을 처리하기 위해 기존의 웹에서 사용하는 80번 포트이외의 별도의 포트(180번 포트)를 할당하여 TCP 연결을 만든다. 새로운 포트에서는 CHTP로 통신하게 된다. 사용자는 특정 사이트에 HTTP 혹은 CHTP를 통해 접속할 수 있고, 어떤 포트로 접속해도 동일한 홈페이지를 제공할 수 있도록 주소 포워딩(forwarding, redirection) 기능을 제공한다.

CHTP에서 전송되는 메시지 형식은 HTTP의 메시지 형식과 유사한 텍스트 형식으로 주고받으며, HTTP와 달리 호출은 클라이언트뿐만 아니라 서버도 가능하다. 즉, 비동기식 양방향 통신을 수행한다. 단, 이 경우 보안측면에서 이용자의 요청 권한을 적절히 제한한다. CHTP의 요청 명령은 HTTP와 같이 메소드로 지칭한다. 요청에 따른 응답 메시지는 정상적인 경우 실행결과를, 비정상적인 경우 상태코드와 메시지를 호출자에게 되돌려준다.

다음의 프로토콜 명세는 BNF(Backus-Naur Form)에 근거한 [RFC 822]을 기준으로 표현된다. 특별히 명시하지 않은 내용은 HTTP 1.1 [RFC 2068]에 준한다.

(1) URL 확장

웹 서버에 접속시 URL에서 포트번호를 지정하지 않으면 기본적인 HTTP 포트(80)와 CHTP 포트 번호(180)로 접속된다. 그러나 확장된 웹 서비스가 두 개의 포트를 사용하기 때문에 웹 서버가 기본 포트로 서비스하지 않을 경우 이를 구별할 방법이 필요하다. 따라서 기존의 URL 표기 형식

에서 포트 지정부를 확장하여 추가적으로 지정하도록 하였다.

```

HTTP_URL = "http:" "/" host [ ":" port ]
↳ [ abs_path [ "?" query ] ]

CHTP_URL = "chtp:" "/" host [ ":"
↳ [ http_port ] [, chtp_port ] ]
↳ [ abs_path [ "?" query ] ]
    
```

(2) 프로토콜 버전

CHTP는 HTTP와 마찬가지로 “<major>.<minor>” 수치로 프로토콜 버전을 표시한다. 이 버전은 CHTP 상호 통신에서 서버 및 클라이언트가 처리할 수 있는 한계를 의미한다.

```
CHTP-Version="CHTP" "/" 1*DIGIT "." 1*DIGIT
```

기타 명시되지 않은 프로토콜 버전 표시 방법은 [RFC 2145]를 따른다.

(3) 메소드

CHTP는 HTTP의 기본 메소드와 더불어 사용자 추적을 위한 확장 메소드를 가진다.

<표 1> CHTP 메소드 기능

메소드	접속 방식	전송 형식	응답 대상	용도
GET	S←C	GET Request-URI window-id	HTTPd	HTTP, CHTP
HEAD	S←C	HEAD Request-URI	HTTPd	HTTP, CHTP
POST	S←C	POST Request-URI	HTTPd	HTTP, CHTP
LOGIN	S←C	LOGIN User-ID Password	CHTPd	CHTP
LOGOUT	S←C	LOGOUT	CHTPd	CHTP
GETVAL	S→C	GETVAL object-id	CHTPd	CHTP
SETVAL	S→C	SETVAL object-id value	CHTPd	CHTP
GOURL	S→C	GOURL redirect-URI window-id	CHTPd	CHTP
NOOP	S←C	NOOP	CHTPd	CHTP

* C: 클라이언트, S: 서버,
HTTPd: HTTP 서버, CHTPd: CHTP 서버

(4) 객체

CHTP에서 사용되는 내부적으로 관리하는 객체는 세 종류로 구분된다. 윈도우 및 프레임 등의 브라우저 인터페이스에 해당하여 문서(document)를 구분하는 대상 객체(target object)와 대상 객체내의 내용의 값에 해당하며 GETVAL, SETVAL에 의해 제어되는 값 객체(value object), 기타 접속 유지와 관련한 서버 세

션 정보를 위한 세션 객체(session object) 등이 그것이다. 해당 객체의 이름은 상황에 따라 새로 만들어지거나 없어질 수 있지만, 전역의 공통적으로 사용되는 객체 이름은 '\$'로 시작된다.

대상 객체를 사용하는 메소드는 GOURL 이다. GOURL에서는 해당 대상 객체의 주소를 변경하여 이동하도록 지시한다. 대상 객체는 HTML에서 지정된 태그 형식에 따른 프레임 이름 혹은 창 이름으로 지정된다. 또한 공통적으로 사용되는 대상 객체는 \$STOP, \$NEW, \$PARENT, \$OPENER, \$SELF 등이 있다. GOURL에서 대상 객체가 지정되지 않는다면 \$SELF를 기본으로 한다.

값 객체를 사용하는 메소드는 GETVAL, SETVAL 등이다. 값 객체는 필요시 실시간 정보 변경 등의 기능으로 웹 내용 전체 혹은 일부를 변경할 때 관리되는 객체이며 페이지별로 즉, URL별로 객체 이름이 별도로 관리된다.

세션 객체 역시 GETVAL, SETVAL 등에 의해 정보가 전달된다. 기본적으로 서버에서만 관리되지만, 로그인 과정이나 외부의 모듈 등에 의해 참조되어질 때 정보가 전달될 수 있다. 세션 객체에서는 전역적으로 사용되는 객체는 \$SESSIONID, \$LOGID, \$LOGNAME 등이 있다.

(5) 상태코드

CHTP에서 응답시 사용되는 상태코드(status code)는 HTTP와 유사하다. HTTP에서는 3자리의 상태 코드와 함께 이유에 대한 설명문(reason phrase)이 클라이언트에게 전송된다.

```
HTTP Status-Code = 3*DIGIT
```

CHTP는 HTTP의 상태코드와 마찬가지로 3자리 형태로 구성된다.

(6) 세션 정보 저장소

사용자의 세션관리를 지원하기 위해 CHTP내 세션 정보, 접속 현황 정보, 현재 값 정보 등 세 개의 정보 저장소를 갖는다. 세션 정보는 클라이언트의 인증 여부와 함께 현재 웹서버에 접속된 모든 클라이언트의 정보를 보관한다. 세션 정보는 시스템에서 할당하는 고유한 세션 아이디, 접속자 IP 주소, 접속 시간, 인증여부에 대한 로그인 아이디와 이름으로 구성된다. 접속 현황 정보)

는 인증된 사용자의 활동을 추적하기 위해 필요한 정보를 보관하며, 세션 아이디, 현재 URL, 진입 시간 등이 보관된다. 현재 값 정보는 동적인 웹 콘텐츠 표현을 위한 것으로, 페이지 URL, 변수 명, 현재 값 등이 저장된다.

```

struct session_info {
    // 세션 정보
    char session_id[MAX_SESSION_ID];
    // 고유 세션 아이디
    char client_ip[MAX_CLIENT_IP];
    // 접속자 IP 주소
    time_t log_time; // 접속 시간
    char logid[MAX_LOGID]; // 접속자 아이디
    char logname[MAX_LOGNAME]; // 접속자 이름
    struct session_info *next;
    // 다음 자료에 대한 포인터
};

struct access_info {
    // 현재 세션별 액세스(접속현황) 정보
    char session_id[MAX_SESSION_ID];
    // 고유 세션 아이디
    char current_url[MAX_URL]; // 현재 접속 페이지
    time_t entry_time; // 진입시간
    char logid[MAX_LOGID]; // 접속자 아이디
    char logname[MAX_LOGNAME]; // 접속자 이름
    struct access_info *next;
    // 다음 자료에 대한 포인터
};
    
```

4. 플랫폼 개발 및 적용

4.1. 개발 환경

개발되는 웹 플랫폼은 기존의 웹 플랫폼과 마찬가지로 인터넷 TCP/IP 환경에서 클라이언트/서버 네트워크 모형으로 동작한다. 웹 서버는 다양한 클라이언트의 요청에 대응하기 위하여 고효율을 도모하고 안정성 있는 운영환경이 필요하다. 본 논문에서는 이러한 환경에 적합한 운영체제로 서버 개발을 위해 Linux를 정하였고, gcc 등을 개발도구를 이용하여 웹서버를 개발한다. 이러한 개발도구를 포함한 시스템 구현 환경을 구체적으로 제시하면 <표 2>와 같다.

4.2. 서버 운영 및 구조

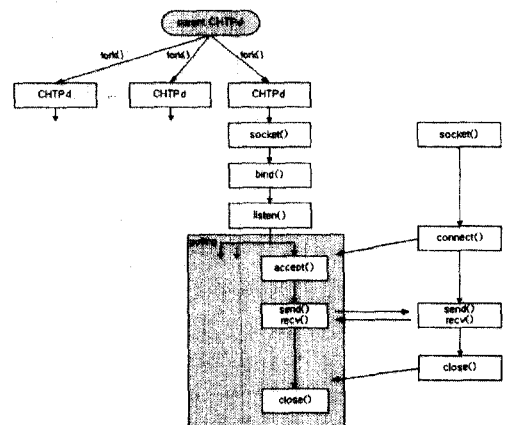
TCP 즉, 연결형 서버는 동작형태에 따라 iterative 형태와 concurrent 형태로 나뉘어진다. Iterative 서버는 클라이언트의 서비스 요구를 순서대로 처리해 주는 서버로서 concurrent 서버에 비해 구현이 간단하지만, 클라이언트로부터의 각

<표 2> 웹 서버 개발 환경

소프트웨어	운영체제	Linux (Kernel/2.4.2)
	저작언어	gcc/2.96, PHP/4.0.4pl
	DBMS	mySQL/3.23.38
하드웨어	CPU	Intel Pentium III Dual
	RAM	1 GB
	VGA	Matrox Millennium G200 AGP
	HDD	40 GB
	NIC	3Com EtherLink XL PCI

요청을 충분히 짧은 시간 동안에 처리할 수 있어 다른 클라이언트들이 기다리는 시간이 거의 없거나 별로 문제가 되지 않는 경우에 사용할 수 있다. 반면 concurrent 서버는 서버 프로그램의 구현이 다소 복잡하지만, 각 클라이언트에 대해 프로세스가 하나씩 생성됨으로써 동시에 여러 클라이언트의 요청 서비스를 개별적으로 처리할 수 있다. 그러나 이 방법은 클라이언트 수가 늘어남에 따라 프로세스 수도 계속 늘어나게 되므로, 다수의 클라이언트가 접속될 수 있는 서비스에서는 프로세스 생성에 대한 서버의 부담이 가중됨으로 사용할 수 없다.

따라서 본 논문의 서버는 새로운 클라이언트들이 새로 접속되어도 이를 처리할 프로세스를 무한히 계속 생성하지 않고, 마치 운영체제가 여러 작업을 스케줄링 하듯이 하나의 프로세스가 모든 서비스를 동시에 처리하는 apparent concurrent 방식을 사용한다.



(그림 2) apparent concurrent 방식의 CHTP 서버 프로세스 구조

OSI 7계층에 비추어 볼 때 CHTP는 전송 계층 상의 TCP를 기반으로 하여 HTTP와 같이 응용 계층에서 동작한다. CHTP 서버는 HTTP와 같은 콘텐츠를 위한 마크업 언어 전송용 포트와 세션 유지 및 명령 송수신을 위한 포트 등의 2개가 별도로 필요하기 때문에, CHTP 서버(CHTPd)를 실행하며 2개의 포트에서 대기하게 된다.

```

manan@comedu ~]$ netstat -ss | grep chtpd
(net all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 0*:*:1179          *:*          LISTEN
20003/htpd
tcp        0      0 0*:*:1180          *:*          LISTEN
20003/htpd
[manan@comedu ~]$
    
```

(그림 3) CHTP 서버의 사용 포트

4.3. 로그처리

웹 사이트를 구성하여 사용자나 서버 활동의 로그 항목을 기록할 수 있다. 웹 서버의 로그 데이터는 로그 분석을 통해서 사용자의 콘텐츠 액세스 조절, 콘텐츠의 선호도 판단, 보안 계획, 잠재적인 웹 사이트의 문제를 해결하는데 도움을 준다. 따라서 대부분의 웹 서버는 기본적으로 서버에서 일어나는 모든 사항들을 파일 형태로 남기는 기능을 가지고 있다.

웹 서버의 로그 처리는 대부분이 텍스트로 처리되고 이것은 관리 및 검색의 효율성을 높여준다. 로그 파일의 형식은 CERN 로그 파일 형식 혹은 NCSA 로그 파일 형식을 기초로 구성되는데, 웹 서버의 종류에 따라 형식내의 항목들은 조금씩 다르며, 서버 관리자의 요구사항에 따라 임의적으로 선택이 가능하다.

초기에는 웹서버가 지원하는 로그 데이터 항목들이 웹 서버마다 달라 로그 분석에 힘든 점이 많았고, 항목들이 다르기 때문에, 로그 분석을 위한 하나의 자동화 처리 프로그램이 모든 로그 파일을 분석할 수 없었다. 그래서 등장한 것이 CLF(Common Log Format)이다. 이 경우 공통적인 형식으로 로그 데이터를 저장하기 때문에 웹 서버가 다르다 하더라도 로그 데이터 형식은 비슷하게 된다. 그리고 그 항목들은 대부분의 상황에 필요한 항목들을 제공한다. 현재의 대부분의

웹 로그는 CERN 혹은 NCSA 에서 HTTP 프로토콜로 규정한 CLF를 따른다. CLF 형식에서 각 항목은 공백으로 구분하며 값을 가지지 않는 항목은 '-'로 기록된다. 이 표준을 따르는 로그 항목에는 사용자의 IP 주소, 데이터 전송을 위한 프로토콜, 에러 코드, 전송된 데이터 길이 등이 포함된다.

그러나 이러한 웹 로그들은 HTTP 프로토콜의 특성상 다음과 같은 문제점을 가지고 있다.

첫째, 로그 분석을 통한 통계의 단위인 히트(hit)는 콘텐츠 요구빈도 분석에 큰 의미를 가지지 못한다. 히트란 한 웹 페이지를 요청했을 때 이와 관련된 모든 파일들을 의미한다. 한 웹 페이지를 구성하고 있는 웹 파일, 이미지 파일, include 파일 등이 히트에 해당한다. 로그 파일에 저장된 모든 행이 히트의 단위가 되며, 이것은 한 번의 웹 페이지 요청에 대해서 기록되는 히트의 수는 몇 배의 숫자가 나올 수 있다. 따라서 직접 명시적으로 요청된 의미있는 요청 정보로서의 페이지 뷰(page view) 기록이 필요하다.

둘째, 로그 정보는 서버 시스템의 입장에서 전체 활동상황에 대한 정량값으로서의 의미만 있을 뿐 각 클라이언트를 구별하고 추적하기는 어렵다. 현재는 같은 IP 주소에서 연속적으로 보면서 페이지 사이의 시간 간격이 일정 시간 이내인 경우, 연속된 의미있는 행동으로 파악하여 하나의 세션으로 처리하고 있다. 또한 일정 시간 이상의 경우 단절로 파악하여 세션의 끝으로 판단한다. 그러나 클라이언트를 구별하기 위한 로그 정보로는 IP 주소가 유일하기 때문에 클라이언트에 대한 복수 세션 정보를 처리할 수 없고, 연결유지가 되지 않기 때문에 각 로그 데이터간 연관성의 정확성을 확보할 수 없다.

세션이라는 의미는 연속된 행동의 단위를 의미하는 것이다. 개발된 웹서버는 세션이 지원됨으로 인해서 기존의 웹로그와 함께 추가적인 세션 처리 관련 로그가 형성된다. 따라서 브라우저별로 웹 서버에 접속할 때마다 별도의 세션이 형성되고 구분하고, 클라이언트가 웹 서버의 제어범위 밖에 있어 세션이 종료될 때까지의 클라이언트 행동을 파악할 수 있게 된다.

CHTP 서버에서는 HTTP와의 호환성을 위해

서 기존 형식과 같은 HTTP 웹 로그와 더불어 세션정보 관리를 위해 세션 아이디를 포함한 CHTP 세션로그를 별도로 생성한다. (그림 4)는 CHTP 서버에 의한 세션로그를 보여주고 있으며, 여기에는 각 행별로 세션아이디, 클라이언트 IP 주소, 인증 여부에 따른 사용자 아이디, 요청 주소, 전송 바이트 수 등이 포함된다.

```
(nannan@comedu chttpd) $ tail -f chttpd.sessionlog
[17/Jun/2002:13:35:01 +0900] 210.93.99.88 20020617133452289363 - "GET /cgi-bin/e
nw.cgi?name=value CHTP/1.0" X20 1128
[17/Jun/2002:13:35:07 +0900] 210.93.99.88 20020617133452289363 - "GET / CHTP/1.0
" X20 1132
[17/Jun/2002:13:36:02 +0900] 210.93.99.88 20020617133602510297 - "GET / CHTP/1.0
" X20 1132
[17/Jun/2002:13:36:05 +0900] 210.93.99.88 20020617133602510297 - "GET /realvalu
.html CHTP/1.0" X20 1136
[17/Jun/2002:13:36:05 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:05" X20 -
[17/Jun/2002:13:36:06 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:06" X20 -
[17/Jun/2002:13:36:07 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:07" X20 -
[17/Jun/2002:13:36:08 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:08" X20 -
[17/Jun/2002:13:36:09 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:09" X20 -
[17/Jun/2002:13:36:10 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:10" X20 -
[17/Jun/2002:13:36:11 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:11" X20 -
[17/Jun/2002:13:36:12 +0900] 210.93.99.88 20020617133602510297 - "SETVAL var 13:
36:12" X20 -
[17/Jun/2002:13:36:13 +0900] 210.93.99.88 20020617133602510297 - "GET / CHTP/1.0
" X20 1132
[17/Jun/2002:13:36:16 +0900] 210.93.99.88 20020617133602510297 - "LOGIN nannan pa
sswd CHTP/1.0" X20 5
[17/Jun/2002:13:36:18 +0900] 210.93.99.88 20020617133602510297 nannan "GET /cgi-b
in/session CHTP/1.0" X20 1076
[17/Jun/2002:13:36:25 +0900] 210.93.99.88 20020617133602510297 nannan "GET / CHTP
/1.0" X20 1132
[17/Jun/2002:13:36:27 +0900] 210.93.99.88 20020617133602510297 nannan "LOGOUT CHT
P/1.0" X20 5
[17/Jun/2002:13:36:29 +0900] 210.93.99.88 20020617133602510297 - "GET /realvalu
.html CHTP/1.0" X20 1136
[17/Jun/2002:13:36:34 +0900] 210.93.99.88 20020617133602510297 - "GET / CHTP/1.0
" X20 1132
#
[영어][한성][두벌식]
```

(그림 4) CHTP 서버의 세션 로그

4.4. 세션 공유

클라이언트의 접속마다 별도의 프로세스가 생성될 때, 각 프로세스들은 독립적으로 수행된다. 이는 어느 하나의 프로세스에 문제가 발생할 때 다른 프로세스들에게 영향을 미치지 않는다는 장점도 있지만, 프로세스들이 독립적으로 존재함으로써 프로세스별로 코드 영역과 데이터 영역이 별도로 존재하며 서로 공유되지 않는다.

그러나 본 서버에서는 각 프로세스에서의 현재 세션 정보를 알 수 있어야 하기 때문에, 프로세스 간의 정보 교환이 요구된다. 이를 위해서 프로세스간 통신(Inter-Process Communication; 이하 IPC) 기술이 필요하다. IPC를 지원하는 방법으로 Unix에서는 기본적으로 파이프(pipe) 및 피포(FIFO)와 더불어, SVR(System V Release)계

열의 Unix에서는 메시지 큐(message queue), 세마포어(semaphore), 공유 메모리(shared memory) 등이 있고, BSD(Berkeley Software Distribution) 계열의 Unix에서는 소켓(socket)이 있다. Linux를 포함한 최근의 Unix 들은 이 기법들을 모두 지원하는데, 본 시스템에서는 이 중에서 공유 메모리(이하 SHM) 기법을 사용한다.

SHM는 둘 혹은 그 이상의 서로 다른 프로세스간의 통신을 하기 위해 시스템 커널의 메모리 일부(a chunk)를 공유한다. SHM 시스템은 일반적인 프로세스나 파일 시스템과 마찬가지로 메모리 상에서 접근권한을 설정할 수 있다. SHM은 메모리 영역에서 직접 정보가 매핑되고, 중개자가 없기 때문에 IPC의 형태 중 가장 빠르다. SHM는 한 개의 프로세스에 의해 만들어질 수 있고, 그 후에 어떤 프로세스든지 그 메모리에 읽기와 쓰기를 할 수 있다. 따라서 SHM를 사용하면 CHTP 서버의 자식 프로세스들이 갖고 있는 클라이언트의 접속 상황 등의 각종 세션정보들을 부모 프로세스가 액세스하여 관리할 수 있게 된다. 이러한 SHM 상태는 쉘 상태에서 'ipcs -m' 명령에 의해 확인할 수 있다.

```
(nannan@comedu ~) $ ipcs -m
----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x00000000  229279     gdn        777        196608    2          dest
0x00000000  327604     root       644        110532    2          dest
0x00000000  360452     root       644        110532    2          dest
0x00000000  383222     root       644        106496    2          dest
0x41095b66  1310727   chtp       800        100       1
0x36055866  1343486   chtp       800        100       1
0x20055866  1474563   chtp       800        100       1
(nannan@comedu ~) $ #
[영어][한성][두벌식]
```

(그림 5) CHTP 서버의 공유 메모리 상황

4.5. 적용 및 평가

본 연구의 웹 플랫폼과 이를 이용한 원격교육 시스템의 효과를 조사하기 위해 사례를 개발하여 검사하였다. 웹 서비스 운영은 기본적으로 서버에서 모든 작업이 이뤄지며 또한, 개발된 웹 플랫폼은 원격 교육 서비스의 이용자 측면보다는 제공자 및 운영자의 측면에서의 효율성을 지향하기 때문에 검사대상으로 웹 서버에 초점을 두었다.

그 대표적인 예로 브라우저의 화면 공유 기능에 적용했다. 브라우저 화면 공유 기능은 전자칠판 기능의 핵심 요소이다. 동기적 웹브라우저 화면 공유는 학습에 참여하고 있는 모든 교수자 및 학습자들이 같은 웹 페이지 화면을 공유함으로써 실시간 원격강의가 이뤄질 수 있도록 한다. 이것은 학습에 참여하고 있는 교수자 및 학습자들간에 지리적으로 서로 떨어져 있는 원격교육 상황에서, 협동학습 뿐만 아니라 일반적인 실시간 교수학습 활동 중 상호작용의 불편함을 해소하고 교육 효과를 확보하는데 중요한 기능이다.

그럼에도 불구하고 자체적인 웹 플랫폼 기술만으로 처리할 수 없기 때문에 수많은 연구[1,3,4,6,7,8]에서 볼 수 있듯이, 이 문제를 별도의 어플리케이션이나 애플릿 혹은 컴포넌트 기술을 사용하여 비디오 스트리밍 전송을 하거나 운영체제의 이벤트 메시지를 후킹하는 방법으로 해결할 수 밖에 없다. 그러나 이것은 호환성 문제와 더불어 개발 비용과 시스템 부하 측면에서 긍정적이지 못한 방법이다.

본 연구에서 개발된 웹 플랫폼에서 지원하는 상호작용적 양방향 특성을 갖는 메소드를 사용하면 안정적인 화면 공유 기능을 처리할 수 있다. 웹 주소 동기화를 통한 브라우저 화면 공유는 GOURL 메소드를 사용하여 처리할 수 있다. 사례 환경에서 동시시간에 접속한 사용자중 통제권을 갖고 있는 교수자는 웹서버의 GOURL 메소드를 이용할 수 있도록 한다. 요청을 받은 웹서버가 다른 접속자(학습자)에게 GOURL 메소드를 전송하면, 수신한 웹 브라우저는 교수자와 동일한 혹은 교수자가 지정하는 웹 주소로 이동하게 된다.



(그림 7) 교수자(좌)와 학습자(우)간의 화면 공유

GOURL 메소드와 함께 window-id를 함께 전송하면 특정 프레임 혹은 윈도 화면에서 제어할

수 있다. 한편, CHTP 서버의 사용자 접속현황 정보를 통해 학습자 브라우저의 현재 상황을 알 수 있다.

(그림 6)은 CHTP 플랫폼에서 웹 브라우저 화면 공유 기능을 구현한 것으로, 사이트에 동시에 접속한 상태에서 각각 교수자와 학습자 상태에서의 실행 화면을 나타낸다. 교수자가 지정한 웹 주소가 다른 학습자에게 전달되어 같은 사이트에 접속하였음을 볼 수 있다. 본 사례는 기본적인 CHTP 플랫폼만으로도 실시간 원격교육 환경을 효과적으로 구축하여 활용할 수 있음을 보여준다.

5. 결 론

웹 기술은 정보통신 영역에서만뿐만 아니라 교육에도 영향을 미쳐 원격교육의 모습과 기능을 새로이 갖추게 하였다. 원격교육은 면대면 교육에서의 시간과 공간의 제약을 극복한 대안교육으로서의 의미와 함께, 웹은 손쉬운 접근, 다양한 기능과 자원으로써 도구로써 융통성 있고 중요한 보조 교육환경을 제공한다. 이러한 원격교육 환경으로서 웹 플랫폼은 콘텐츠를 학습자에게 전달하고, 학습자의 학습활동 등 교육활동 전반에 걸쳐 효과적으로 지원할 필요가 있다.

본 논문은 이러한 문제의식에서 출발하여 원격교육의 관점에서 현재의 웹 기술과 상향 호환성을 갖는 웹 플랫폼을 프로토콜 차원에서 개발하였다.

새로운 웹 플랫폼은 기존의 웹 구조와 기술을 그대로 수용하면서도 동적이고 효과적인 원격교육 환경을 확보할 수 있다는 의미를 갖는다. 새로운 웹 플랫폼의 유용성은 개발 및 운영의 효율성과 더불어 세션로그를 통해서 사용자의 학습과정을 추적할 수 있다는 점에 있다. 더불어 현재 세션확보가 이뤄진 상태이기 때문에 프로토콜 차원에서 상호작용을 지원할 수 있도록 쉽게 확장이 가능하다.

추후 개발된 웹서버를 좀 더 확장하고 모듈화하는 후속 연구가 필요하다. 본 연구에서 개발한 CHTP 서버는 연구의 목적과 내용에 대한 구현과 준거로서 개발되었다. 이로 인해 CHTP 서버

활용의 실제성을 높이기 위해 기능을 모듈화하고
규격화하여 서드파드 모듈의 결합을 원활하게 할
수 있도록 하는 연구가 필요하다. 따라서 PHP,
Python 등의 서버측 스크립트 언어와 MySQL,
Oracle, Informix 등의 RDBMS와의 외부 프로그
램과 연동이 쉽도록 하여 실용성을 높일 필요가
있다.

참 고 문 헌

- [1] 김문석, 성미영(2001). 동기적 웹 브라우저
공유를 지원하는 협동작업 시스템, 정보처리
학회 논문지, 8-B(3), 283-288.
- [2] 김원태(1998), PEP를 이용한 HTTP/1.1의
Method 확장. 가야대학교 논문집, 7(1),
80-90.
- [3] 이점숙, 이부권, 서영권(2001). 웹 기반의 실
시간 원격강의를 위한 서버와 클라이언트
웹브라우저 동기화. 정보처리학회 논문지,
8-A(1), 70-74.
- [4] 황대준(1997). 사이버 스페이스상의 상호참
여형 실시간 원격 교육 시스템에 관한 연구.
정보처리학회 논문지, 4(3), 29-40.
- [5] 황성준(1998). 가상대학 비전과 한계 - 정보
기술의 혁신과 가상대학 출현, 대학교육,
Vol. 98.
- [6] Greenberg, S., & Rseman, M. (1996).
GroupWeb: A WWW Browser as Real
Time Groupware, In ACM SIGCHI'96
Conference on Human Factors in
Computing System, Companion
Proceedings, 271-272.
- [7] Jacobs, S., Gebhardt, M. Kethers, S., &
Rzasa, W. (1996). Filling HTML Forms
Simultaneously: CoWeb Architecture and
Functionality. In Proceedings of the Fifth
International World Wide Web Conference,
1385-1395.
- [8] Makoto Kobayashi, Masahide Shinozaki,
Takashi Sakiri, Maroun Touma, Shahrokh
Daijavad, & Catherine Wolf (1998).
Collaborative Customer Services Using
Synchronous Web Browser Sharing, ACM
conference on CSCW, 99-108.
- [9] Marzo-Lázaro, J. L., Verdú-Carbó, T., &
Fabregat-Gesa, R. (1998). User
Indentification And Tracking In An
Educational Web Environment. Proceeding
of ED-MEDIA/ED-TELECOM 98 World
Conference on Educational Multimedia and
Hypermedia & World Conference on
Educational Telecommunicatins, Freiburg,
Germany, June 20-25. (ERIC Document
Reproduction service No. ED 428 496)
- [10] Nielsen, H., Leach. P., & Lawrence, S.
(2000). An HTTP Extension Framework,
RFC 2774.
- [11] Robinson, P. (2000). Where Is
Every-Body?. In Robert A. Cole (Ed),
Issues in Web-Based Pedagogy. London:
Greenwood Press, 111-123.
- [12] <http://www.w3.org/>.
- [13] <http://www.adlnet.org/>.
- [14] <http://aicc.org/>.
- [15] <http://ariadne.unil.ch/>.
- [16] <http://www.eoe.org/>.
- [17] <http://ltsc.ieee.org>.
- [18] <http://www.imsglobal.org/>.
- [19] <http://jtclsc36.org/>.

박 종 오



1992 한국교원대학교
수학교육과(교육학학사)

1995 한국교원대학교

컴퓨터교육과(교육학석사)

2002 한국교원대학교 컴퓨터교육과(교육학박사)

관심분야: 컴퓨터교육, 원격교육시스템

E-Mail: modu@chol.com