

# 상호작용적 학습 객체 지원을 위한 I-HTTP 개발

## Development of I-HTTP for supporting Interactive Learning Object

정영식(Young-Sik Jeong)\*

### 요 약

본 연구에서는 학습자간 상호작용 도구를 재사용하기 위해서 학습 객체 콘텐츠 표준화 기술을 적용하여 상호작용적 학습 객체인 ILO를 정의하고, 이것이 LMS와 원활하게 통신할 수 있도록 I-HTTP를 개발하였다. I-HTTP는 비지속적 연결성을 갖는 기존의 HTTP를 개선하여 일정 세션 유지 시간 동안 접속 상태를 지속하도록 하였다. 이러한 지속적인 연결로 인해 ILO 내에서 빈번하게 발생하는 학습자간 상호작용을 실시간으로 제공할 수 있었다. 또한 I-HTTP는 기존의 HTTP를 확장하였기 때문에 ILO는 물론 일반 HTML 문서도 처리할 수 있게 하였다. 특히 LMS와 ILO간의 표준화된 런치 과정을 프로토콜 내의 INIT, GETVAL, SETVAL, COMMIT, FINISH 메소드를 추가하여 구현하였으며, ILO의 학습자간 상호작용 결과는 별도로 정의된 데이터 모델을 통하여 데이터베이스 서버에 저장될 수 있도록 하였다.

### Abstract

The purpose of this study was to define an interactive learning object of ILO through implementation of learning object content standardization technology for the reuse of interactive tools between learners, and to develop I(Interactive)-HTTP for the ILO to properly communicate with LMS. I-HTTP developed here was enabled to keep connection status during the entire session by improving the existing HTTP with its stateless connection property. This ceaseless connection made it possible to provide users with the real-time interactivity between learners that happened frequently in the ILO. Also, because the I-HTTP was an expanded version of HTTP, it was possible to conduct general HTML documentation as well as ILO. In particular, the standardized launch process between LMS and ILO was embodied in adding the INIT, GETVAL, SETVAL, COMMIT, FINISH methods in the protocol, and the results from the interactivity between ILO learners were channeled to the database storage to save them through separately defined data models.

## 2. 관련 연구

### 1. 서론

지식 정보화 시대의 학습자의 역할은 지식을 수동적으로 전수하는 역할에서 능동적으로 구성하는 역할을 해오고 있다. 또한 특징인예 의해 소유되고 독점되던 정보는 네트워크 기반의 통신 기술의 발달로 공유하는 개념으로 발전하고 있다. 따라서 다양한 정보와 아이디어를 함께 나누면서 자신의 의견을 표출하고 나아가 지식을 구성해나갈 수 있는 학습자간 상호작용의 역할이 크게 강조되고 있다.

이러한 학습자간 상호작용이 웹을 통해 활발하게 이루어지기 위해서는 학습 콘텐츠뿐만 아니라 그 콘텐츠를 통해서 다양한 의견을 주고받을 수 있는 풍부한 상호작용 도구를 제공해야만 한다. 왜냐하면 개인마다 상호작용 도구에 대한 선호도가 다르고, 상호작용 도구마다 지원하는 상호작용 활동이 서로 다르기 때문이다[6].

다양한 상호작용 도구를 지원하기 위해서는 먼저 여러 가지 상호작용 도구가 개발되어야 하며, 개발된 도구는 재사용될 수 있어야 한다. 그러나 기존의 상호작용 도구들은 특정 플랫폼 기반으로 만들어져 있어 재사용이 불가능하고, 주로 학습 내용을 그대로 전달하는 위주로 개발되었으며, 학습자의 상호작용 활동 내용을 추적할 수 없었다[1][3][4].

따라서 상호작용 도구를 재사용하기 위해서는 학습 객체(learning object) 기반의 콘텐츠 표준화 기술을 적용하여 상호작용 도구를 학습 객체화해야 한다. 그러나 현재의 콘텐츠 표준화 기술은 군대 훈련처럼 개별 학습이나 자기 조절 학습을 중심으로 한 제한적인 학습 환경만을 제공하기 때문에 학습자와 학습자간의 다양한 상호작용을 제공하지 못하고 있다[8].

따라서 본 연구에서는 상호작용적 학습 객체를 지원하기 위하여 HTTP를 확장한(I(Interactive)-HTTP)을 개발하였다.

#### 2.1 상호작용적 학습 객체(ILO)

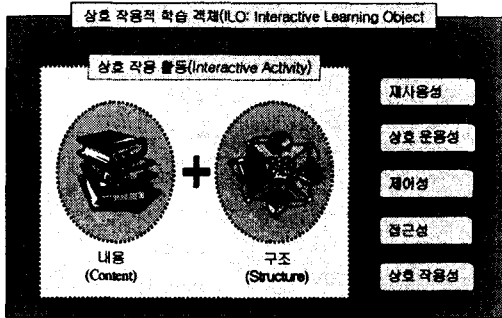
학습 객체는 명확한 교수·학습적 목표를 가진 독립적이고 재사용이 가능한 콘텐츠 조각을 말한다. 그 크기는 작게는 하나의 미디어 파일일 수도 있고 크게는 하나의 코스일 수도 있다. 또한 메타데이터(meta-data)로 설명되어 있어 자료 저장소(repository)에 저장되어 사용자들이 웹 기반 환경에서 쉽게 검색하여 다른 학습 객체들과 통합하여 코스를 구성하는 것이 가능한 전자적 형태의 콘텐츠이다[5].

학습 객체의 대표적인 예로는 ADL(Advanced Distributed Learning)의 SCO(Sharable Content Object)가 있다. 그러나 SCO는 개인적이고 개별화된 상호작용만을 제공한다. 즉 SCORM이 일반적인 평생 교육이 아니라 미군의 특수한 목적을 위해 만들어진 시스템 내에서 기계적이고 개별화된 훈련을 위해 설계되었기 때문에 학습자와 시스템간의 상호작용만을 제공한다. 또한 SCORM 명세에 나와 있듯이 협력 학습을 전혀 제공하지 못하고 있다[5].

따라서 학습자와 학습자간의 상호작용을 제공하기 위해서는 학습 관리 시스템에서 [그림 1]과 같은 상호작용적 학습 객체(Interactive Learning Object : ILO)를 지원해야 한다.

ILO는 내용과 구조로 구성된다. 내용은 학습 목표를 달성하기 위해서 학습자에게 전달되는 매개물을 말하며, 구조는 내용을 전달하기 위해 사용되는 다양한 상호작용 도구를 말한다. 또한 구조에 내용이 담겨질 때 상호작용 활동이 이루어진다. 이러한 상호작용 활동이 학습 객체로서의 속성인 재사용성, 상호 운용성, 제어성, 접근성, 상호작용성을 갖을 때 ILO로서의 의미를 갖는다.

재사용성은 한 번 만들어진 ILO는 별다른



[그림 1] 상호작용적 학습 객체(ILO)

수정 없이 다른 과정에서도 사용될 수 있음을 말하며, 상호 운용성은 서로 다른 시스템에서 소스 수정 없이 사용될 수 있을 말한다. 제어성은 학습자의 상호작용 상태를 추적할 수 있음을 말하며, 접근성은 메타데이터로 ILO가 설명되어 있어 누구든지 검색하여 사용할 수 있음을 말한다. 끝으로 상호작용성은 콘텐츠와 관련된 학습자와 학습자간의 다양한 정보를 교환할 수 있음을 뜻한다.

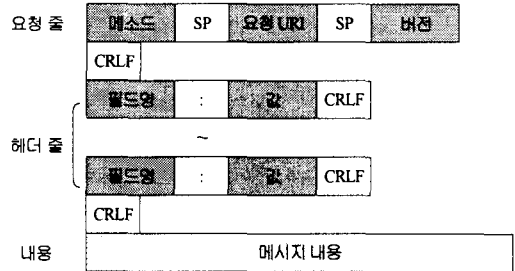
## 2.2 HTTP

HTTP(Hypertext Transfer Protocol)는 WWW에서 HTML(Hyper Text Markup Language) 문서를 송수신하기 위해서 사용하고 있는 프로토콜로서 TCP를 전송 계층 프로토콜로 이용해서 이루어지는 애플리케이션 계층 프로토콜 가운데 하나이다[7].

### (1) HTTP의 메시지

HTTP 메시지는 HTTP 요청 메시지와 응답 메시지의 두 가지 형태를 갖는다.

요청 메시지의 구조는 [그림 2]와 같이 요청 줄(request line)과 헤더 줄(header lines), 내용(entity body)로 구분된다.

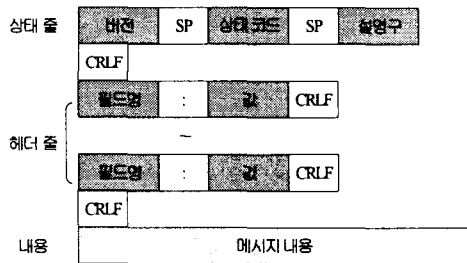


[그림 2] HTTP의 요청 메시지

요청 줄의 메소드는 GET, POST, HEAD 등과 같은 종류가 있다. 요청 URI는 서버에게 요구하려는 페이지나 자료의 위치를 나타내며, 버전은 현재의 HTTP 버전을 나타낸다. 헤더 줄은 필드명과 값으로 구성되며, 내용은 실제 서버에 전달할 메시지의 내용을 담고 있다.

응답 메시지의 구조는 [그림 3]과 같이 상태 줄(status line)과 헤더 줄(header lines), 내용(entity body)으로 구분된다.

서버에서 사용되는 프로토콜 버전, 클라이언트의 요청(Request)에 대한 응답 결과, 전달해 줄 데이터의 형식, 데이터 길이 등과 같은 추가적인 정보가 전달된다. 헤더 정보가 끝나면 빈줄(CRLF)를 이용하여 실제 전달될 메시지 내용과 구분한다.



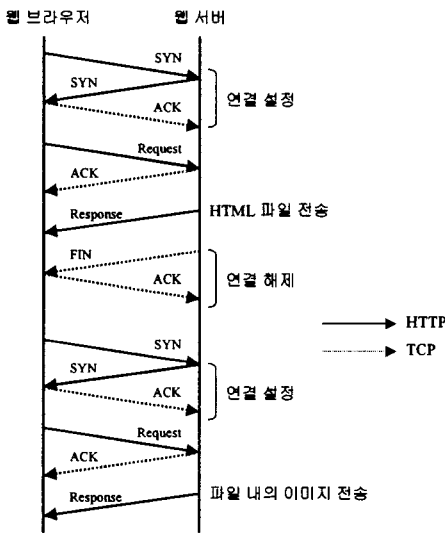
[그림 3] HTTP의 응답 메시지

상태 코드는 요청 결과의 상태를 간단한 10진수로 나타낸 것이며 설명구는 그 상태 코드에 대한 간단한 설명을 나타낸다. 예를 들면

클라이언트의 요구를 정상적으로 처리했을 경우 상태 코드는 '200', 설명구는 'OK'를 나타낸다.

(2) HTTP의 연결 관리

HTTP/1.0의 비지속적(stateless) 연결 방식은 [그림 4]와 같이 클라이언트가 같은 서버에 연속적인 문서 요청을 하더라도 서버는 요청한 문서 파일을 전송하고 나면 자동적으로 연결을 해제한다. 구체적인 연결 과정은 다음과 같다.



[그림 4] HTTP/1.0의 연결 관리

첫째, 웹 브라우저는 SYN 패킷으로 TCP 연결을 요청하면 서버는 성공적인 수신 의미로 SYN 패킷을 클라이언트에 전달한다. 클라이언트는 SYN 패킷을 서버에게서 받으면 TCP 연결이 정상적으로 연결되었다는 의미로 ACK 패킷을 전달한다.

둘째, 웹 브라우저는 사용자의 요청 혹은 입력에 따라 요청 메시지를 생성한다. 즉, 헤더 정보를 구성하고 이어 보낼 데이터가 있으면 준비한다.

셋째, 웹 브라우저는 서버에게 요청 메시지를 보낸다.

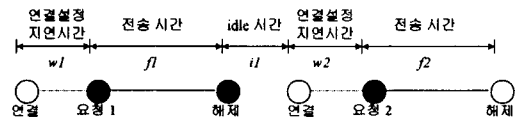
넷째, 서버는 브라우저의 요청 메시지를 분석하여 응답 메시지를 생성한다. 응답 메시지의 헤더는 요청 결과에 대한 정보와 데이터형을 나타낸다.

다섯째, 응답 메시지의 헤더 정보를 보낸 뒤에 곧바로 실제 데이터를 보낸다.

여섯째, 서버는 응답 메시지를 보낸 후 바로 FIN 패킷을 통해 연결을 종료한다.

일곱째, HTML 파일 내부에 이미지 파일 등이 있으면 웹 브라우저는 연결 설정 과정을 똑같이 반복한다.

HTTP/1.0은 TCP를 기반으로 하는 비지속적 프로토콜로서 각각의 요청에 대해 개별적인 TCP 연결을 설정한다. 따라서 [그림 5]와 같이 지연 시간이 발생하여 학습자간의 빈번하고 지속적인 상호작용을 제공하는 데는 많은 문제점이 있다.



[그림 5] HTTP의 전송 지연

즉, [그림 5]에서 HTTP를 이용하여 n회의 요청이 있을 경우 최초 연결부터 마지막 해제 때까지의 총 지연 시간은 다음과 같다.

$$W_n = w_1 + w_2 + \dots + w_n = \sum_{i=1}^n w_i$$

ILO에서는 작은 데이터를 빈번하게 전송하게 되므로  $W_n$ 의 값이 전송할 때마다 늘어나 전송 지연은 그만큼 커질 것이다.

### 3. I-HTTP의 설계 및 구현

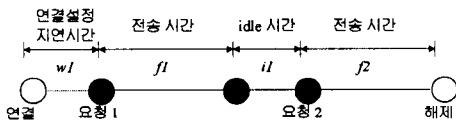
#### 3.1 설계

##### (1) 설계 원칙

ILO를 이용하여 학습자간 상호작용이 가능하기 위해서는 세션 관리가 가능하고 요청 메시지의 메소드를 이용하여 학습 객체와 관련

된 데이터 모델(Data Model)을 주고받을 수 있는 새로운 프로토콜이 필요하다. 따라서 I-HTTP를 설계할 때 다음과 같은 점을 고려해야 한다.

첫째, 접속 상태를 세션 유지 시간 동안 유지해야 한다. 관련 연구에서 살펴 본 바와 같이 HTTP는 비지속적 연결로 인해 연결 지연 시간이 발생하여 학습자간 빈번한 상호작용이나 실시간 상호작용을 처리하기에는 어려움이 많다. 따라서 웹 브라우저를 통해 학습자가 서버에 접속할 때 세션 상태 유지 시간동안 접속 상태를 유지해야 한다. 만약 I-HTTP에서 [그림 6]과 같이 접속 상태를 유지한다면 ILO에서 빈번하게 학습자간 상호작용을 요구한다고 할지라도 실제 전송 시간만 소요될 것이다.



[그림 6] I-HTTP의 전송 지연

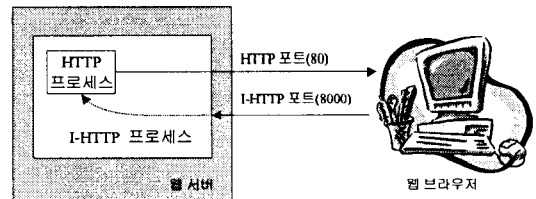
둘째, 세션 정보와 접속 상태를 관리해야 한다. 각 사용자별 접속 상태와 세션 정보를 유지함으로써 중복 로그인을 방지할 수 있고 학습자간의 빈번한 상호작용을 실시간으로 지원할 수 있다. 이러한 세션 정보와 접속 상태를 관리하기 위해서 <표 1>과 같은 테이블을 사용하여 로그 파일과 데이터베이스에 저장하였다.

<표 1> 세션 관리 테이블

sessionid	ip	logtime	logid
20030901000	210.93.99.	2003-09-01	stu1
01	1	14:00:01	
20030901000	210.93.99.	2003-09-01	stu2
02	2	14:10:01	
20030901000	210.93.99.	2003-09-01	stu3
03	3	14:11:22	
20030901000	210.93.99.	2003-09-01	
04	4	14:11:41	

셋째, 상향 호환성을 가져야 한다. I-HTTP를 이용하는 클라이언트에서 기존 HTTP를 이용하여 일반적인 HTML 문서를 요청할 경우 기존의 HTTP 웹 서버에 전달하여 응답해야 한다.

[그림 7]은 I-HTTP에서의 HTTP 처리 과



[그림 7] I-HTTP에서의 HTTP 처리 과정

정을 나타낸 것이다.

I-HTTP 포트를 통해 HTTP 요청이 들어오면 HTTP 프로세스가 처리할 수 있도록 클라이언트에 HTTP 요청 페이지를 리다이렉트한다.

(2) I-HTTP와 HTTP의 비교

I-HTTP의 특성을 HTTP와 비교하면 <표 2>와 같다.

<표 2> I-HTTP와 HTTP의 비교

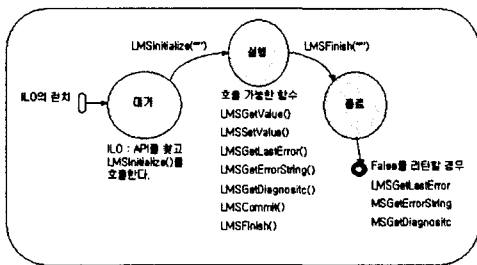
구분	HTTP	I-HTTP
연결성	비연결성	연결성
세션유지	유지 안함	유지함
포트	80	8000
추적기능	추적 안함	추적함
전달기능	HTML 문서 전달	데이터 모델 송수신

I-HTTP는 세션 유지가 가능하고 일정 시간 동안 지속적으로 연결되어 있으며 포트는 8000 번을 사용한다. 또한 학습자의 정보를 데이터 모델을 통해서 추적할 수 있다.

### 3.2 I-HTTP 개발

#### (1) ILO의 통신 과정

ILO가 재사용되기 위해서는 콘텐츠 표준화 규격을 따라야 한다. [그림 8]은 ILO가 SCORM 기반의 LMS(Learning Management System)와 통신하는 과정을 나타낸 것이다. ILO는 API를 이용하여 대기, 실행, 종료 단계를 거쳐 LMS와 통신한다.



[그림 155] ILO의 통신 과정

첫째, 대기 단계에서는 자바 애플릿으로 된 API 어댑터를 찾는다. API 어댑터가 발견되면 LMSInitialize()를 호출하여 ILO와 관련된 schema.xml을 읽어들이고 후 필요한 테이블과 인덱스를 생성한다.

둘째, 실행 단계는 LMSInitialize()에 의해 초

기화된 후에 시작된다. 이 단계에서는 LMSSetValue()와 LMSGetValue()를 이용하여 LMS와 ILO가 학습자간 상호작용 결과를 주고 받을 수 있다. 또한 현재 호출되고 있는 API 함수의 실행 상태와 에러 정보를 파악하기 위해서 LMSGetLastError(), LMSGetErrorString(), LMSGetDiagnostic()을 호출할 수 있다.

셋째, 종료 단계는 LMSFinish()를 호출하면서 시작된다. 이 단계에서는 학습자의 정보를 저장한 후 현재 사용 중인 ILO를 빠져 나와 다른 학습 객체로 이동하거나 별도의 지정된 활동을 수행한다.

#### (2) I-HTTP의 메소드

I-HTTP가 ILO의 지원과 HTTP의 호환을 위해서 기존의 HTTP 요청 메시지에 <표 3> 과 같은 메소드를 추가하였다.

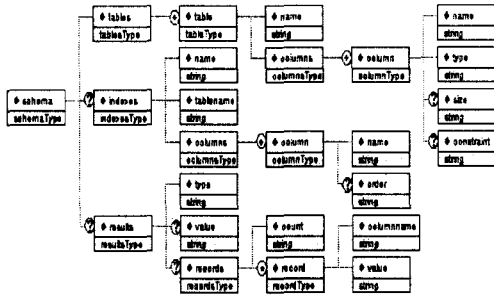
<표 3> I-HTTP의 메소드

메소드	전송 방향	전송 형식	호출 API
INIT	C→S	INIT	LMSInitialize()
GETVAL	C↔S	GETVAL param	LMSGetValue()
SETVAL	C↔S	SETVAL param value	LMSSetValue()
COMMIT	C→S	COMMIT	LMSCommit()
FINISH	C→S	FINISH	LMSFinish()

I-HTTP에서 제공하는 각 메소드의 기능은 다음과 같다.

첫째, INIT 메소드는 클라이언트에서 호출하는 ILO를 초기화한다. 이때 ILO와 관련된 schema.xml을 읽어들이고 데이터베이스 서버에 필요한 테이블을 생성한다.

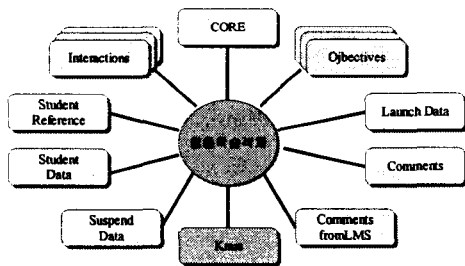
schema.xml의 구조는 [그림 9]와 같다.



[그림 9] ILO의 스키마 파악

둘째, GETVAL 메소드는 서버와 클라이언트 모두 사용 가능하다. 어떤 데이터를 요청할 것인가는 param을 통해 전달가능하며, 이것은 ILO의 데이터 모델만을 사용할 수 있다.

ILO의 데이터 모델은 [그림 10]과 같이 cmi 데이터 모델과 knue 데이터 모델로 구분된다.



[그림 10] ILO의 데이터 모델

cmi 데이터 모델은 콘텐츠 표준화 기술에 제공하는 데이터 모델이다. core는 학습자 아이디나 이름과 같이 LMS에 의해 제공되기 위해 요구되는 기본 정보를 나타내며, suspend\_data는 이전에 종료될 때 현재 사용을 위해 필요한 정보를 나타낸다. launch\_data는 ILO가 생성될 때 만들어지는 정보이다. comments와 comments\_from\_lms는 주석을 수집하고 전달하기 위해 사용된다. objectives는 학습자가 ILO에서 다루는 개별적인 학습 목표

들을 어떻게 수행했는지를 나타내며, student\_data와 student\_preference는 학생들에 대한 추가적인 정보를 나타낸다. interactions는 평가에 사용되는 학습자의 입력 정보를 인식하고 저장할 수 있게 한다.

knue 데이터 모델은 ILO의 학습자간 상호작용 결과를 처리하는 모델로서 세부적인 항목은 [그림 11]과 같이 테이블, 쿼리문, 결과치, 동기화로 구분될 수 있다. 특히 쿼리 실행 결과는 일반적으로 I-HTTP의 응답 메시지에 포함되지만 실행된 결과를 다시 요청할 때에는 결과치와 관련된 데이터 모델을 사용할 수 있다.

테이블	knue.db.table.name	액세스할 테이블명
쿼리문	knue.db.query	실행할 DML 쿼리문
결과치	knue.db.result.type	결과가 레코드셋인가?
	knue.db.result.value	쿼리 수행 결과값
동기화	knue.db.result.count	레코드셋의 전체 수
	knue.sync.id	데이터의 갱신 여부

[그림 11] knue 데이터 모델의 세부 항목

셋째, SETVAL은 클라이언트에서 서버의 데이터를 변경하거나 저장하고 할 때 사용할 수 있다. 특히 학습자간에 발생한 다양한 상호작용 결과는 knue 데이터 모델을 이용하여 저장하거나 수정할 수 있다.

넷째, COMMIT는 SETVAL 메소드를 통해 주고받은 결과를 최종적으로 데이터베이스 서버에 저장하는 역할을 한다.

다섯째, FINISH는 클라이언트와 서버의 연결을 종료할 때 사용된다. 일반적으로 FINISH를 호출하지 않아도 서버 환경 변수에서 설정된 세션 유지 시간이 지날 경우 자동으로 연결이 해제된다.

### (3) 웹 서버 개발

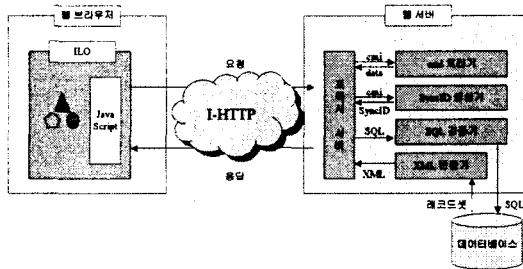
I-HTTP 프로토콜을 실행할 수 있는 웹 서버는 <표 4>와 같은 환경에서 개발·운영되

었다.

<표 4> I-HTTP의 개발 환경

소프트웨어		사양	
소프트웨어	OS	Windows 2000 Server SP3	
	언어	C	
하드웨어	CPU	Intel Pentium IV 1700MHz	
	RAM	261,569KB	

I-HTTP 웹 서버의 기능은 [그림 12]와 같이 프록시 서버, cmi 처리기, SyncID 생성기, SQL 검증기, XML 변환기 등으로 구성된다.



[그림 12] I-HTTP 웹 서버의 기능

ILO와 웹 서버의 통신 절차는 다음과 같다.

첫째, ILO에서는 I-HTTP를 이용하여 INIT를 호출한다. ILO가 최초로 호출될 경우에는 ILO의 schema.xml을 이용하여 데이터베이스 서버에서 상호작용에 필요한 테이블을 생성한다. 테이블의 중복 생성을 막기 위해 매핑 테이블에 의해 관리된다.

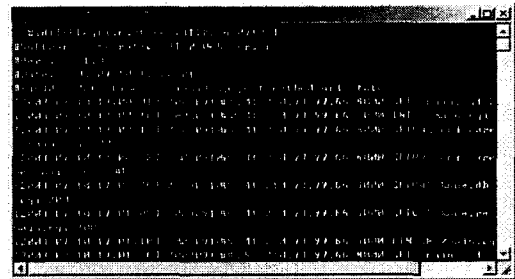
둘째, INIT 이후에 사용자는 GETVAL, SETVAL, COMMIT 메소드를 호출할 수 있다. cmi 데이터 모델은 cmi 처리기에서 응답 메시지를 생성한다. 그러나 knue 데이터 모델에서 SQL 실행을 요청할 경우 SQL 검증기를

통해 데이터베이스에 액세스하고 그 결과를 XML 변환기를 통해 XML로 생성한 후 그것을 응답 메시지에 담아 클라이언트에 전송한다.

셋째, 세션이 종료되거나 학습자가 강제적으로 학습을 종료할 때에는 FINISH 메소드가 호출된다. 웹 서버는 현재의 상호작용 상태를 데이터베이스에 저장하고 종료 상태를 응답 메시지에 담아 클라이언트에 전송한다.

I-HTTP 웹 서버는 사용자의 요청이 발생할 때마다 로그 정보를 로그 파일에 기록하고, 세션 정보를 데이터베이스 서버에 저장한다.

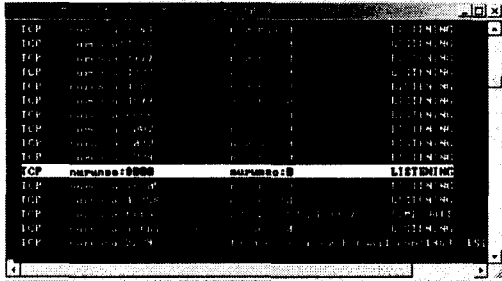
[그림 13]은 웹 서버의 로그 파일 정보를 나타낸 것이다. 웹 서버의 이름과 버전, 그리고 로그 파일의 생성 날짜와 시각을 기록한다. 로그 파일은 각 사용자가 접속한 날짜, 시각, 세션 아이디, 접속자의 IP 주소, 접속 포트 번호, 요청 메소드와 파라미터, 요청 URI, 상태 코드 등을 기록한다.



[그림 13] I-HTTP 서버의 로그 파일

I-HTTP 서버는 [그림 14]와 같이 명령 송·수신과 세션 유지 등을 위하여 8000번 포트와 일반 HTML 문서 요청 응답을 위해 80번 포트 2개를 사용한다.

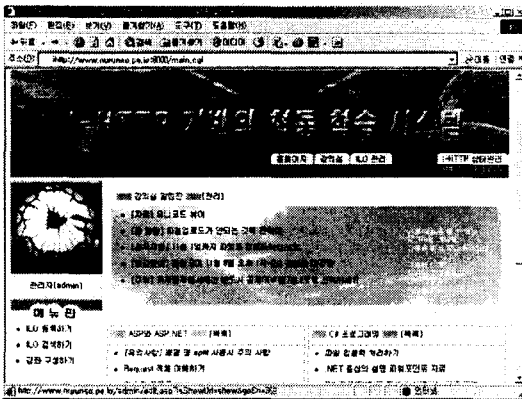




[그림 14] I-HTTP 서버의 사용 포트

(4) I-HTTP 기반 협동 학습 시스템 개발

I-HTTP는 HTTP와 달리 접속 상태를 유지하고 세션 정보를 관리하고 있기 때문에 다양한 상호작용을 제공하는 ILO를 이용하여 [그림 15]와 같은 웹 기반 협동 학습 시스템을 실험 운영하였다.



[그림 15] I-HTTP 기반의 협동 학습 시스템

4. 결론

본 연구에서는 상호작용 도구를 재사용하기 위해서 학습 객체 콘텐츠 표준화 기술을 적용하여 상호작용적 학습 객체인 ILO를 정의하고,

이를 지원하기 위해 HTTP를 확장한 I-HTTP 프로토콜을 개발하였다.

I-HTTP는 기존의 HTTP의 비지속적 연결을 개선하고 ILO의 상호작용 정보를 추적하고 저장하기 위해서 일정 세션 유지 시간 동안 접속 상태와 세션 정보를 유지하고 관리할 수 있도록 하였다.

또한 I-HTTP는 기존의 HTTP를 확장하였기 때문에 ILO 뿐만 아니라 일반 HTML 문서도 처리할 수 있으며, SCORM 기반의 학습 객체도 추적할 수 있다. 또한 LMS와 ILO간의 표준화된 런치 과정을 프로토콜 내의 INIT, GETVAL, SETVAL, COMMIT, FINISH 등을 확장하여 지원하였으며, ILO의 학습자간 상호작용 결과는 별도로 정의된 데이터 모델을 확장하여 데이터베이스 서버에 저장될 수 있도록 하였다.

본 연구를 통해서 콘텐츠와 학습자간의 개인적 상호작용만을 중시하는 SCORM 기반의 콘텐츠 표준화 기술의 문제점을 해결하고, 자바 애플릿을 기반으로 한 애플리케이션 수준의 통신 방법을 I-HTTP를 이용한 프로토콜 수준의 통신 방법으로 개선함으로써 콘텐츠 표준화 기술의 교육적 가치를 높이고자 하였다.

그러나 본 연구에서 ILO의 상호작용 수준이 채팅과 같은 실시간 상호작용을 제공하기보다는 게시판 수준의 비실시간 상호작용만을 제공하기 때문에 I-HTTP에 대한 보다 다양한 연구와 보완이 필요하며, I-HTTP를 기반으로 한 협동 학습 시스템을 실제 학교 현장에 적용하여 문제점을 해결하려는 노력이 필요하겠다.

## 참고 문헌

- [1] 권숙진, 웹 기반 학습 환경에서 학습자간 상호작용 지원을 위한 협력 학습 플랫폼 프로토타입 개발. 한양대학교 석사학위논문, 2000.
- [2] 박인우, 임진호, SCORM에서의 학습 객체 (Learning Object) 개념의 교육학적 시사점, 2003년 동계 학술발표논문지, 7(1), pp.55-61. 2003.
- [3] 이성주, 김희수, CMC 활용 학습에서 면대면 수업, 교사 관여 및 메타인지가 학업 성취와 상호작용에 미치는 영향. 교육정보방송연구, 9(1), pp. 95-128, 한국교육정보방송학회, 2003.
- [4] 정웅진, 정순영, 실시간 피드백을 기반으로 하는 협동학습지원 시스템. 한국컴퓨터교육학회 하계 학술발표논문지, 7(2), pp.392-399, 2003.
- [5] ADL(Advanced Distributed Learning), Sharable Content Reference Model Version 1.2, The SCORM Overview. [Online] Available: <http://www.adlnet.org>, 2003.
- [6] Everett, D. R, Taking instruction online: The art of delivery. *CSS Journal*, 8(2). [Online] Available: <http://www.cssjournal.com/everett.html>, 1998.
- [7] W3C(World Wide Web Consortium), Hypertext Transfer Protocol-HTTP/1.0. [Online] Available: <http://www.w3.org/Protocols/rfc1945/rfc1945>, 2003.
- [8] Wiley, D, Learning Objects: Difficulties and Opportunities. [Online] Available: [http://wiley.ed.usu.edu/docs/lo\\_do.pdf](http://wiley.ed.usu.edu/docs/lo_do.pdf), 2003.

정영식



1996년 춘천교육대학교 수학교육학과(교육학학사)  
2001년 한국교원대학교 컴퓨터교육과 졸업(교육학석사)  
2001~현재 한국교원대학교 대학원 컴퓨터교육과 박사과정  
관심분야 : 컴퓨터교육, 프로그래밍 교육, 원격교육