

WWW에서의 과제처리 시스템 구현

장우현[†] · 이남준^{††} · 박문호^{†††} · 이하욱^{†††} · 이봉환^{†††}

요 약

본 논문에서는 대학내의 연구자들로부터의 과제제안서를 인터넷을 이용하여 처리하는 시스템을 구현하였다. 인터넷에 접속되어 있는 연구자 단말, 연구처 호스트 및 과제 수여재단의 호스트 사이에 제안서를 네트워크를 통하여 전송하고 제안서 검토, 결재, 평가 등의 업무도 온라인으로 처리한다. 연구자는 단말에 Web 브라우저를 설치하고 대학 및 과제수여재단에서는 과제 처리를 위하여 Web 서버 및 홈페이지를 구축한다. 연구자는 작성한 제안서를 대학의 연구처 Web 서버에 접속하여 업로드 하고 연구처에서는 내부 전자결재를 거쳐 제안서들을 과제 수여재단으로 전송한다. 과제수여재단의 Web 서버에는 각 대학에서 접수되는 과제들을 기관별로 분리하여 검토할 수 있는 기능을 구현하였다. 검토가 종료된 과제는 평가위원을 위촉하여 평가를 의뢰하고 그 결과를 과제수여재단의 서버에 직접 입력할 수 있는 기능을 구현하였다.

Implementation of a Proposal Processing System on the World Wide Web

Woo-Hyun Jang[†] · Nam-Jun Lee^{††} · Moon-Ho Park^{†††} · Ha-Wook Lee^{†††} · Bong-Hwan Lee^{†††}

ABSTRACT

This paper presents a proposal processing system which can transmit and receive research proposals using the Internet. The system is composed of three parts: principal investigator's terminal, host computer of university's proposal processing department, and host computer of research foundation. The key functions of the system include file download, upload, directory listing, and authentication among principal investigator's terminal, Web server at the university, and Web server at the research foundation. A Web browser is to be installed at researcher's terminal while a Web server and home page for proposal processing need to be constructed at university's host computer and research foundation's host computer, respectively. The Web server provides directory listing and authentication. Personnel at the research foundation can retrieve received proposals and send them to the preassigned reviewers to evaluate the proposals. The reviewers can access the Web server at the research foundation and input scores of the evaluated proposals.

1. 서 론

※본 연구는 한국과학재단 핵심전문연구비(961-0905-034-1)지원으로 수행되었으며 지원에 감사드립니다.

† 정희원: 한국전자통신연구원 정보통신표준연구센터

†† 정희원: 한국전자통신연구원 이동통신기기연구실

††† 정희원: 대전대학교 공과대학 정보통신공학과

논문접수: 1997년 6월 13일, 심사완료: 1997년 12월 3일

인터넷이 연구자들의 정보교환 및 원격 시스템 접속 등으로 널리 이용되면서 인터넷 응용서비스들은 연구과제 처리시스템에 활용되게 되었다. 미국에서는 수년 전부터 NSF에서 연구과제 처리를 위해 기존의 인터넷이 제공하는 FTP 및 e-mail 기능에 관련 소프트웨어를 개발하여 연구과제 처리를 위해 사용하고

있다. 최근에는 WWW(World Wide Web) 기술이 널리 활용됨에 따라 NSF에서는 Web에 CGI 인터페이스 기술을 응용하여 Fastlane이라는 이름으로 전국에 있는 연구자들이 과제 응모에 관한 정보를 검색할 수 있게 하고 과제의 신청, 처리, 평가 등 전반적인 서비스를 제공하여 연구자 및 과제 관리기관의 업무에 지대한 도움을 주고 있다[1]. 국내에서도 대부분의 대학이 인터넷에 연결되어 있고 여러 통신회사에서 상용 인터넷 서비스를 제공하고 있으나 연구과제 수여재단에서는 PC 통신으로 연구과제에 대한 안내 서비스를 제공하거나 WWW 서버를 구축하여 연구 관련 정보의 열람 및 관련 파일의 다운로드 정도만의 서비스를 제공하고 있다. 따라서 본 연구에서는 연구과제 처리를 위한 전반적인 업무를 온라인으로 처리할 수 있는 시스템을 개발하였다.

2. 연구과제 처리 시스템

2.1 개요

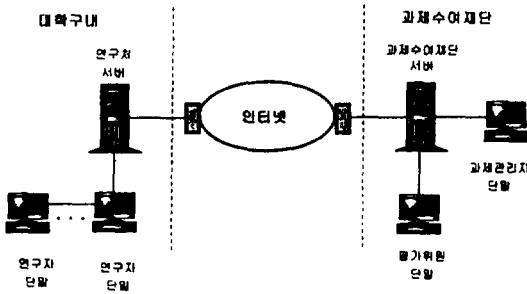
연구과제 처리시스템은 최근 널리 사용되고 있는 WWW를 이용하여 연구 과제 제안에 필요한 정보를 제공하고 제안서 제출, 평가 등의 제반 업무를 온라인으로 처리하기 위한 시스템이다. 국내에서도 일부 과제 수여재단에서 대학이나 연구기관으로부터의 과제 제안서에 대한 정보를 Web을 통하여 제공하고 있으나 대학 내의 결제 과정, 재단으로의 송부, 평가위원으로의 제안서 송부 등의 모든 절차는 아직도 우편을 통하여 이루어지고 있다. 과제 처리시스템에서 대학 구내에 있는 과제 제출자들의 단말이 구내 LAN을 통하여 서로 연결되어 있고 이들은 다시 연구처 Web 서버와도 연결되어 있으며 과제 수여재단의 Web 서버와 인터넷 상에서 상호연결되어 있다. 과제 공모 시 수여재단은 호스트 컴퓨터에 등록되어 있는 전자우편 리스트(Mailing list)를 참조하여 각 대학 및 연구기관의 과제 관리부서로 과제 신청에 관한 전자메일을 보낸다. 연구자들은 재단의 Web 서버에 접속하여 제안서 작성요령, 표준 서식 파일 등을 열람하고 필요한 파일을 자신의 단말로 다운로드 한다. 표준서식 파일을 수정하여 제안서 작성이 완료되면 연구자는 대학 내의 연구처 Web 서버에 접속하여 제안서 파일을 업로드 시키고 연구처에 전자메일을 보낸다. 연구

처 담당 직원은 송부된 제안서 파일들이 있는 디렉토리에 사용자명과 패스워드를 사용하여 로그인하여 연구자들로부터의 제안서를 일목요연하게 검색하여 검토하게 된다. 연구처에서는 검토가 종료된 제안서들을 취합하여 결제권자에게 넘기고 결제 요청을 위한 전자메일을 보낸다. 결제권자는 각 제안서를 검토하고 전자결제를 한 다음 이를 다시 연구처로 송부한다. 연구처에서는 재단에서 사전에 부여한 대학별 사용자명과 패스워드를 사용하여 재단의 Web 서버에 접속하여 해당 사업에 적합한 디렉토리에 연구 제안서들을 업로드 한다. 이 때 과제별로 재단에 전송되는 정보는 사전에 할당된 대학별 접수번호, 연구자명, 대학명, 과제명 등이다. 재단의 과제 담당자는 대학별로 송부된 과제 제안서를 대학 별로 열람하여 검토하게 된다. 검토가 종료된 제안서를 평가하기 위해 재단의 과제 담당자는 과제별로 평가위원을 선정하여 평가위원에게 전자우편을 보내어 평가의뢰를 하고 평가할 제안서 파일을 함께 보낸다. 이 때 재단에서는 평가위원에게 재단의 Web 서버 가운데 평가용 홈페이지를 액세스 할 수 있는 패스워드를 제공한다. 평가위원은 전자메일로 받은 제안서를 심사하고 그 결과를 재단의 평가용 홈페이지를 액세스하여 온라인으로 입력한다. 각 과제별 평가위원들은 이러한 과정을 수행하게 되고 여러 명의 평가위원으로부터 입력되는 평가결과는 자동으로 취합되어 평균점수가 산출된다. 평가위원으로부터의 평가의견 및 평가점수를 토대로 재단에서는 선정위원회를 개최하여 최종 선정을 하고 그 결과를 각 대학으로 통보하고 동시에 재단의 홈페이지에 공고한다.

2.2 하드웨어 구성

본 연구에서 구현한 전자 과제 처리 시스템의 구성도를 도시하면 (그림 1)과 같다. 연구자 단말과 연구처 호스트는 대학의 구내 LAN 및 인터넷을 통하여 연결되어 있고 재단의 호스트 및 평가위원들의 단말도 인터넷을 통하여 서로 연결된다.

연구자 단말, 연구처 단말 및 과제관리자 단말에는 펜티엄 PC에 Win95를 설치하여 사용하였으며, 연구처 서버와 과제 수여재단 서버로는 Sparc10 UNIX 시스템에 Solaris 2.5를 탑재하여 TCP/IP로 연결 하였다[2][3][4].



(그림 1) 과제처리 시스템의 구성도
(Fig. 1) Configuration of the proposal processing system

2.3 소프트웨어 구성

본 절에서는 전자 과제처리 시스템 구축을 위하여 필요한 Web 서버 및 클라이언트의 기능에 대하여 기술하고 Web 서버에서 과제처리를 위해 필요한 CGI 프로그래밍에 대하여 기술하기로 한다.

2.3.1 Web 서버 및 클라이언트

전자 과제 처리 시스템의 Web 서버로는 NCSA HTTPd를 사용하였으며 연구자, 연구처 직원, 제단의 과제 담당직원 및 평가위원 등의 단말을 위한 Web 브라우저로는 현재 가장 널리 사용되고 있는 Netscape 브라우저를 사용하였다[5].

2.3.2 CGI 프로그래밍

연구자가 작성한 제안서의 연구처 서버로의 전송, 전송된 과제의 열람 및 결제, 또는 평가위원들의 과제 평가 후 점수 입력, 사용자 확인 등의 기능을 구현하기 위해서는 CGI 프로그래밍이 필요하다. 본 연구에서는 이러한 CGI 프로그래밍을 위하여 PERL 스크립트를 이용하였다[6][7]. Web 클라이언트와 서버 사이의 상호 작용을 기술하면 다음과 같다.

(1) Web 브라우저와 서버가 데이터를 교환하는 과정

Web 브라우저와 Web 서버 사이에 데이터가 교환되기 위해서는 소켓이 설정되어야 한다. Web 서버는 소켓을 생성하면서 TCP 프로그램에게 특정 포트 번호를 지정하여 그 포트 번호로 들어오는 데이터를 자신에게 전해줄 것을 요청하게 된다. 소켓이 설정된

후에 Web 서버는 해당 포트 번호를 통한 연결 요청이 들어오기를 기다리게 되고 Web 서버 쪽의 TCP 프로그램은 IP 프로그램으로부터 데이터가 전달되기를 기다리게 된다. 한편, Web 브라우저는 URI(Uniform Resource Identifier)가 선택되면 이 URI의 도메인 이름과 포트 번호를 분석하여 자신의 TCP 프로그램과 소켓을 설정한 후에 Web 서버와의 연결을 설정할 것을 요청하게 된다. 연결 요청을 받은 TCP 프로그램은 연결 요청 메시지를 만들고 Web 서버의 IP 어드레스와 함께 이들을 IP 프로그램에게 전달한다. IP 프로그램은 TCP 프로그램으로부터 전달 받은 메시지를 서버에 전송한다. 이 때 사용되는 전송 프로토콜이 HTTP이다. Web 서버쪽의 IP 프로그램은 다시 이 메시지를 상위 계층에 있는 TCP 프로그램에게 전달하게 된다. TCP 프로그램은 전달받은 메시지의 헤더에서 포트 번호를 이용하여 해당 포트 번호를 나타내는 소켓이 설정되어 있는가를 확인한다. Web 서버가 연결 설정을 기다리고 있으므로 Web 서버에게 연결 요청을 알리고 연결이 되면 이를 Web 브라우저의 TCP 프로그램에게 알리기 위해서 응답 메시지를 보내게 된다. 연결이 이루어진 후에는 Web 서버는 소켓을 통해서 데이터가 들어오기를 기다리게 된다. 연결 응답을 받은 Web 브라우저의 TCP 프로그램은 소켓을 통해서 Web 서버에게 전달될 데이터를 전송하게 된다. 이 데이터를 받은 TCP 프로그램은 데이터에 포트 번호를 추가하여 IP 프로그램에 전달한다. IP 프로그램은 TCP 프로그램으로부터 전달받은 데이터를 Web 서버쪽 IP 프로그램에게 인터넷을 통해서 전송한다. Web 서버쪽 IP 프로그램은 인터넷을 통해서 수신된 데이터를 TCP 프로그램에게 전달한다. TCP 프로그램은 IP 프로그램으로부터 받은 데이터를 포트 번호를 보고 이미 연결이 설정되어 있음을 확인한 후에 해당 소켓을 통해서 Web 서버에게 전달하게 된다. 이렇게 해서 Web 브라우저가 보낸 데이터가 Web 서버에게 전달되게 되는 것이다[8].

(2) Web 서버의 CGI 기능

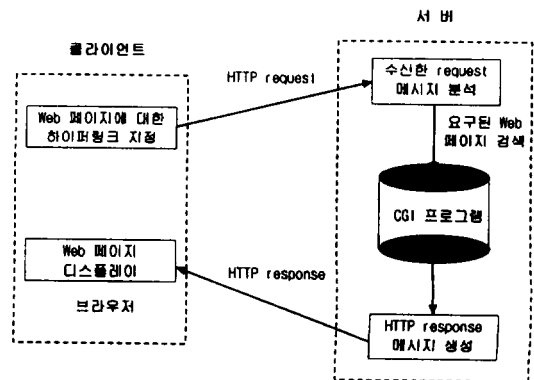
HTML은 Web 페이지를 통해서 다양한 형태의 입력을 받아들일 수 있는 그래픽 사용자 인터페이스 부분을 구성하는데 사용된다. HTML에 의해서 구성되는 윈도우를 통해서 CGI 프로그램이 사용자들의 입

력을 받아들이고 결과를 출력하게 된다. HTTP는 사용자의 입력을 CGI 프로그램으로 전달하는데 사용된다. Web 브라우저는 HTML form 입력 데이터를 HTTP request 메시지 형태로 서버에게 전달한다. HTTP request 메시지는 여러 종류의 헤더들과 서버에게 전달되는 데이터 부분으로 구성된다. HTML form 입력 데이터를 처리하는 두 가지 방식(GET과 POST)에 따라서 HTML form 입력 데이터가 서버에게 전달되는 방법과 서버가 수신된 HTML form 입력 데이터를 CGI 프로그램에게 전달해 주는 방법도 달라지게 된다. 이와 같은 방법은 HTML form을 정의할 때 지정하게 되며 HTTP request 메시지의 첫 부분을 구성하는 Request-line을 통해서 서버에게 전달된다. 또한 Request-line을 통해서 HTML form 입력 데이터를 처리할 CGI 프로그램의 URI 값도 서버에게 전달되게 된다. GET이 사용되는 경우에는 HTML form 입력 데이터가 Request-line을 구성하는 요소 중의 하나인 URI 필드에 추가되어 전달된다. 그러나, POST인 경우에는 헤더와는 별도의 데이터로 서버에게 전달되게 된다. Web 브라우저의 request 메시지를 수신한 Web 서버는 request 메시지의 Request-line에서 CGI 프로그램의 URI 값을 찾고 이 값을 이용해서 CGI 프로그램을 실행시키게 된다. 이때 서버는 Request-line에서 지정된 방법에 따라서 HTML form 입력 데이터를 CGI 프로그램에게 넘겨준다. 또한, request 메시지에 포함된 다른 헤더들의 정보도 환경 변수를 통하여 CGI 프로그램에게 전달된다.

CGI 프로그램에서는 HTML form 데이터를 이용하여 지정된 동작을 수행한 후에 Web 브라우저에게 전달될 데이터를 서버에게 전달하게 된다. 이때, CGI 프로그램은 Web 브라우저에게 전달될 데이터의 타입을 지정하는 헤더를 가장 먼저 출력해야 한다. CGI 프로그램의 출력은 대부분의 경우에 HTML 코드가 된다. HTML 코드를 출력하는 CGI 프로그램에서는 최소한 Content-type 헤더는 반드시 출력하여야 한다. Content-type 헤더에는 CGI 프로그램이 출력하는 데이터의 미디어 타입이 지정된다. 서버는 이와 같은 Content-type 헤더 외에도 몇가지 헤더를 더 추가하여 CGI 프로그램의 출력 데이터와 함께 Web 브라우저에게 HTTP response 메시지를 전달하게 된다. Web 브라우저는 이와 같은 Content-type 헤더값을

참조해서 전송되어온 데이터를 그 타입에 적절한 방식으로 처리할 수 있게 된다.

(그림 2)는 Web 브라우저에서 CGI 프로그램의 URI를 지정하는 하이퍼링크를 선택한 경우 URI가 지정하는 CGI 프로그램이 실행되고 그 결과가 Web 브라우저에게 전달되는 과정을 나타낸다. 먼저 CGI 프로그램이 있는 서버를 나타내는 URI가 선택되면 Web 브라우저는 HTTP request header를 만들어 Web 서버에게 전달하게 된다. 이 HTTP request header에는 CGI 프로그램의 URI가 포함되어 있다. Web 브라우저로부터 HTTP request header를 받은 Web 서버는 header에 포함된 URI 정보를 분석하여 CGI 프로그램을 찾아 실행시키게 된다. 이때, path information이나 query string 데이터가 URI에 포함되어 있으면 CGI 프로그램이 실행되기 전에 그 값들은 각각 환경변수에 저장되게 된다. CGI 프로그램은 Web 브라우저에게 전달될 HTTP response header와 HTML 파일을 출력하여야 하고 이것이 Web 서버에 의해서 Web 브라우저에게 전달되게 된다.

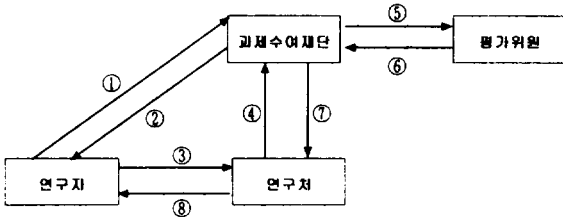


(그림 2) 브라우저와 서버 사이의 메시지 전달과정
(Fig. 2) Message transfer between Web browser and server

3. 전자 과제 처리 시스템의 구현

3.1 과제 처리 시나리오

전자 과제 처리 시스템의 전체적인 처리 과정에 대한 흐름도를 블록 다이어그램으로 나타내면 (그림 3)과 같다.



(그림 3) 전자과제 처리 시스템의 흐름도
(Fig. 3) Flow diagram of the proposal processing system

(그림 3)에서의 처리과정을 차례로 기술하면 다음과 같다.

- ① 연구자는 과제수여재단 웹 서버에 접속하여 과제 관련 Web 페이지로 하이퍼링크한다.
- ② 해당 Web 페이지에서 과제에 관련된 표준 서식 파일들을 다운로드하여 연구제안서를 작성한다.
- ③ 교내 연구처 서버에 접속하여 제안서를 전송하고 전자메일을 보내 연구처 담당 직원에게 통보한다.
- ④ 접수된 제안서들을 다운로드하여 작성 양식에 맞게 작성되었는지를 확인하고 오류가 없으면 이 파일들을 결재권자에게 전송한다. 결재권자는 자신의 사용자명과 패스워드를 사용하여 교내 Web 서버에 접속한 다음 자신의 홈페이지에 접수된 제안서 파일들을 검토하고 전자 결재를 한 다음 다시 연구처 과제 담당 직원에게 전송한다. 담당 직원은 사전에 부여 받은 대학별 사용자명과 패스워드를 이용하여 과제수여재단 Web 서버에 접속하여 제안서 파일들을 업로드 한다.
- ⑤ 과제수여재단의 담당 직원은 대학별로 전송된 제안서들에 대하여 각 과제별로 평가위원을 선정하여 평가위원들에게 과제를 전자메일을 통하여 송부한다. 이 때 평가위원에게는 각 과제별로 평가 결과를 입력할 때 필요한 사용자명과 패스워드도 함께 전송한다.
- ⑥ 평가위원은 송부된 제안서에 대한 평가를 한 다음 과제수여재단 Web 서버에 접속하여 해당과제에 대한 점수를 입력한다. 평가위원이 결과를 입력하면 자동적으로 합계 및 평균값이 계산된다.
- ⑦ 재단의 과제 담당 직원은 입력된 과제 평가 결과를 토대로 선정 위원회를 개최하여 수행 예정과

제를 선정하고 그 결과를 각 대학의 연구처로 통보한다. 이 때 선정된 과제에 대한 평가위원의 평가의견도 함께 전송하고 탈락된 과제에 대해서는 평가의견을 개인별로 전자메일을 통하여 통보한다.

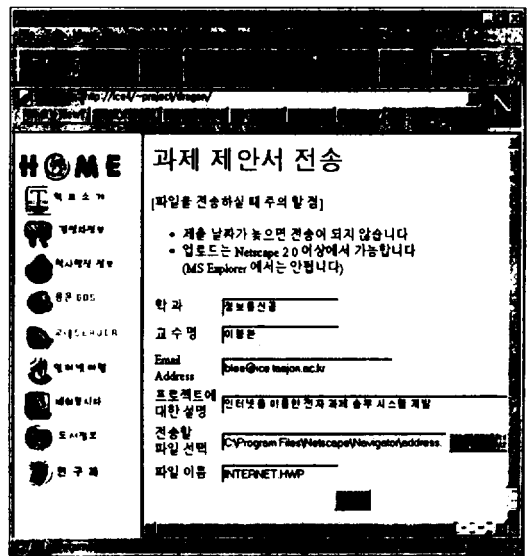
⑧ 연구처 과제 담당자는 선정된 과제의 연구자에게 과제가 선정되었음을 알려준다.

3.2 전자 과제 처리 시스템 구현 예

본 절에서는 구현한 전자 과제 처리 시스템의 실제 사용예를 기능별로 도시하고 설명하기로 한다.

(1) 과제 제안서 전송 화면

연구자는 과제수여재단 서버에 접속하여 표준 서식 파일을 다운로드 받아 과제제안서를 작성한 다음 연구처 홈페이지에 접속한다. 제안서를 연구처로 전송하기 위하여 연구자는 학과, 교수명, 전자메일 주소, 과제명, 파일명을 입력하고 전송할 과제 파일을 하드 디스크에서 검색하여 선택한다. 모든 필드에 대한 입력이 끝나면 “전송”이라는 버튼을 클릭하여 연구처로 파일을 전송한다. 제안서 전송 화면은 (그림 4)와 같다. 과제 전송이 이상없이

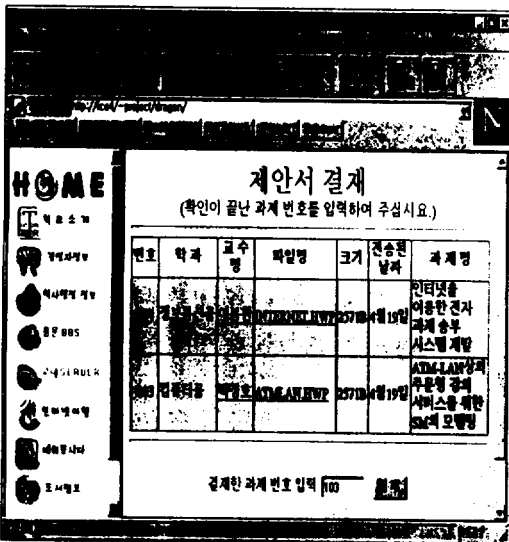


(그림 4) 과제 제안서 전송 화면
(Fig. 4) Proposal transmission window

수행되면 확인 메시지가 다음 Web 페이지에 디스플레이된다.

(2)과제 제안서 결재

연구처로 전송된 제안서들을 검토하기 위하여 연구처 과제 담당자는 인증 절차를 거쳐 각 제안서들을 확인한다. 즉, 각 제안서가 형식에 맞게 작성되었는지를 자신의 워드프로세서를 사용하여 확인한 다음 이상이 있으면 제안자에게 교정을 요구하고 없으면 제안서들을 모아 결재권자에게 송부한다. 과제 제안서 결재권자는 연구처 홈페이지에 접속하여 결재권자 확인 절차를 거친 다음 “제안서 결재” 페이지에서 결재할 목록을 디스플레이 하여 과제 담당 직원과 같은 방법으로 과제를 확인한 후에 전자 결재를 한다. (그림 5)는 결재 페이지에 디스플레이되는 목록 확인 화면을 나타낸 것이다. 과제에 대한 확인이 종료 되면 과제 번호를 입력하고 나서 결재 버튼을 클릭한다.

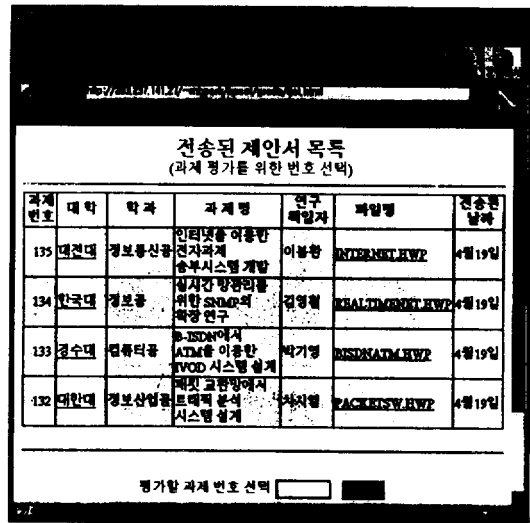


(그림 5) 과제 제안서 결재 화면
(Fig. 5) Proposal approval window

(3)평가 과제 목록

결제가 종료된 제안서는 다시 연구처로 전송되고 연구처 직원은 이 과제들을 취합하여 과제수여재단으로 전송한다. “전송된 제안서 목록” 페이지에서는

(그림 6)과 같이 각 대학의 연구처에서 전송한 과제들의 목록을 일목요연하게 볼 수 있는 기능을 제공한다. 각 과제에 대한 확인이 이루어지고 평가위원 선정이 끝나면 평가위원들에게 전자메일을 이용하여 평가할 제안서와 평가 후 과제수여재단의 홈페이지에 접속하여 점수를 입력할 수 있도록 사용자명과 패스워드를 부여한다.



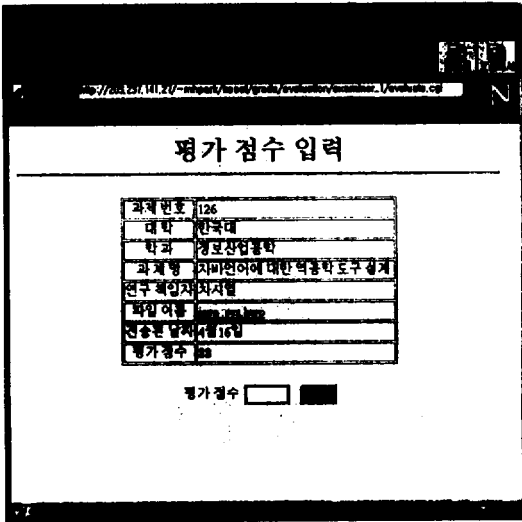
(그림 6) 송부 과제 목록 확인 화면
(Fig. 6) Directory listing window of uploaded proposals

(4)평가위원의 평가 점수 확인

(그림 7)은 과제 평가가 종료된 후에 평가 점수를 입력한 평가위원이 해당 과제에 대한 평가 점수를 확인하는 화면이다. 해당 과제의 평가 점수가 정확히 입력되었으면 “Confirm” 버튼을 클릭하고, 잘못 입력되었으면 “Cancel” 버튼을 클릭하고 다시 점수를 입력한다. 한 과제당 여러 명의 평가위원이 위촉되므로 각 위원이 입력한 평가점수 중 마지막 위원이 점수를 입력하면 자동적으로 평균이 계산되고 각 과제별 평가결과 목록을 출력하도록 구현하였다.

4. 결 론

본 논문에서는 인터넷을 이용하여 연구과제 업무를 신속, 간편하게 처리할 수 있는 시스템을 구현하



(그림 7) 입력한 평가 점수 확인
(Fig. 7) Score confirmation window

었다. 과제수여재단이 구축해 놓은 홈페이지에 각 사업별 응모요령, 서식파일 등을 올려 놓으면 과제 응모에 관심이 있는 연구자는 Web 서버에 접속하여 필요한 정보를 검색하고 제안서 서식 파일을 자신의 단말로 다운로드 한다. 연구자는 이 파일을 수정하여 제안서를 작성하고 이를 대학의 연구처 Web 서버에 접속하여 업로드 한다. 연구자로부터 연구처에 전송된 과제를 검토하고 대학 내의 결재를 위하여 대학의 Web 서버에 파일 다운로드 기능 및 결재권자로의 파일 업로드 기능을 구현하였다. 결재가 종료된 과제들은 일괄적으로 이미 구축해 놓은 과제 수여재단의 홈페이지 가운데 해당 사업에 대한 홈페이지로 업로드 되고 파일 전송 후 전자메일을 통하여 과제 송부 사실을 통보 받게 된다. 과제 수여재단에서는 각 대학에서 송부된 과제를 검토한 후 각 과제별 평가위원을 선정하여 각 위원에게 평가 의의를 위한 전자메일을 보내고 제안서 파일도 함께 전송한다. 평가가 종료되면 평가위원은 수여재단의 홈페이지에 접속하여 해당 과제에 대한 평가 결과를 입력한다. 전체 과제에 대하여 1차 평가가 종료되면 재단에서는 선정 위원회를 거쳐 최종 결정을 하고 그 결과를 각 대학의 연구처로 전자메일을 통하여 송부하고 평가위원으로부터 의견 등은 각 연구자에게 전자메일을 통하여 직접

전송한다.

본 연구에서 구현한 시스템이 실제 업무에 활용될 경우 연구자 및 과제관리자의 업무효율 제고 뿐 아니라 기존의 우편에 의한 문서처리 과정이 없어지게 되므로 경제적인 측면에서도 기여하는 바가 클 것으로 사료된다.

참고 문헌

- [1] B. H. Lee, Final Report of EPS(Electronic Proposal Submission) Project, Dept. of Elec. Eng. of Texas A & M Univ., 1993.
- [2] D. E. Comer and D. L. Stevens, Internetworking with TCP/IP, Vol. I, II, III, Prectice Hall, 1991.
- [3] M. A. Miller, LAN Protocol Handbook, M&T, 1992.
- [4] W. R. Stevens, TCP/IP Illustrated, Vol. I, II, Addison Wesley, 1994.
- [5] E. Nemeth, G. Snyder, and S. Seebass, UNIX Administration Handbook, Prentice Hall, 1989.
- [6] E. Hermann, CGI Programming with PERL5, Sams. net, 1996.
- [7] W. Satanek, HTML CGI SGML VRML JAVA, Sams. net Publishing, 1996.
- [8] J. December and M. Ginsburg, HTML & CGI, Sams. net, 1995.



장 우 천

1982년 중앙대학교 전자계산학과 졸업(학사)
1991년 중앙대학교 전자계산학과 졸업(이학석사)
1992년~현재 한국전자통신연구원 정보통신표준연구센터 책임연구원

관심분야: 소프트웨어공학, 분산시스템



이 남 준

1978년 서강대학교 전자공학과 졸업(학사)
1983년~현재 한국전자통신연구원 선임연구원
관심분야: 전자교환기, 이동통신, 지능망, FPLMTS



박 문 호

1996년 대전대학교 정보통신공학과 졸업(학사)

1996년~현재 대전대학교 정보통신공학과 석사과정

관심분야: ATM LAN, 멀티미디어 통신



이 봉 환

1985년 서강대학교 전자공학과 졸업(학사)

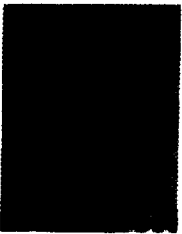
1987년 연세대학교 대학원 전자공학과 졸업(공학석사)

1993년 텍사스 A&M 대학교 전기과 졸업(공학박사)

1987년~1995년 한국통신 연구원

1995년~현재 대전대학교 정보통신공학과 조교수

관심분야: 컴퓨터네트워크, ATM 네트워크, 멀티미디어 통신



이 하 옥

1996년 대전대학교 정보통신공학과 졸업(학사)

1996년~현재 대전대학교 정보통신공학과 석사과정

관심분야: ATM LAN, 멀티미디어 통신