

원격실행 기술을 이용한 강의지원 웹사이트 자동생성시스템 설계 및 구현

임 인 택[†] · 김 재 일^{††} · 송 규 백^{†††} · 김 중 근^{††††}

요 약

인터넷의 급격한 발전에 따라 홈페이지를 쉽고 빠르게 제작할 수 있는 개발 툴들이 소개되고 있다. 하지만, 이러한 툴에 의해 작성된 홈페이지는 아직 기능상에 많은 제약이 있다. 일반적으로 특수한 목적의 홈페이지를 개설하기 위해서는 웹 저작 툴, HTML, CGI 프로그래밍 등에 관한 기술들을 잘 알고 사용할 수 있어야 한다. 특히 강의지원용 웹사이트는 경적인 웹페이지와 능적인 웹페이지를 생성하기 위한 여러 기술(CGI 등)의 집합체로서 교수 본인이 구축하기에는 상당한 기술적/시간적 비용이 요구된다. 본 논문에서는 이러한 어려움을 해결하기 위하여 원격지에서 원격실행을 통하여 응용 S/W를 자동설치/관리하는 하부구조(RASIS)를 개발하였으며, 이를 이용해 웹 브라우저에서 강의과목에 대한 파라미터들만 입력하면, 지정된 웹 서버에 강의지원용 웹사이트를 자동생성하는 시스템을 설계하고 구현하였다.

Design And Implementation Of A Lecture Supporting Web Site Construction System Using Remote Execution Techniques

In-Taek Leem[†] · Jae-Il Kim^{††} · Kyu-Baek Song^{†††} · Chong-Gun Kim^{††††}

ABSTRACT

Recently, various web page development tools for both the beginner and the experienced user are introduced. These tools allow them to generate web pages easily and quickly. However the web pages generated by these tools have lots of functional limitations. Generally authors must have much knowledge for web authoring tools, HTML, CGI programming to open web sites for special purpose. Especially, most of the lecture supporting web site necessarily requires much effort to construct it as well as special functions using CGI, Javascript, Java Applet, etc. to generate dynamic web pages. In order to solve above mentioned limitations, we design and implement an automatic web site construction system using RASIS based on remote execution technologies.

1. 서 론

웹(World Wide Web)은 문자 기반이 아닌 멀티미디어 요소를 포함하며 사용자에게 친숙한 GUI환경을 제

공하고, 인터넷상에 연결된 각종 시스템들에 분산되어 있는 다양한 정보와 서비스들을 사전 지식이 없는 사용자들이 편리하고 손쉽게 접근할 수 있도록 힘으로써, 웹에 대한 수요와 관심이 폭발적으로 증대해 왔다.

웹은 WAN에서 네트워크 기반의 애플리케이션을 개발하는 데 있어서 사실상 표준이 되어 왔다[1]. 웹은 인터넷/인트라넷 애플리케이션을 도입하기 위한 핵심 기술로서 채택되어 발전되고 있으며, 웹을 기반으로 하는 멀티미디어/하이퍼미디어 응용은 주문형 비디오,

* 이 논문은 1999년 정보통신부 우수시범학교 지원사업의 연구비 지원을 받아 연구되었음
† 준 회원 : 대구미래대학 멀티미디어정보학과 교수
†† 정 회원 : 문경전문대학 컴퓨터정보학과 교수
††† 준 회원 : 영남대학교 진한정보원
†††† 정 회원 : 영남대학교 컴퓨터공학부 교수
논문접수 : 1999년 10월 21일, 심사완료 : 2000년 5월 2일

홈쇼핑, 원격 교육 등 다양한 분야로 발전하고 있다. 이러한 응용의 하나로 교육환경을 개선시키기 위해 각 기관 및 개인은 강좌용 홈페이지 등을 구축하고 강의 지원 시스템에 연결시키고 있다 하지만 이러한 강좌용 홈페이지를 작성하기 위해서는 홈페이지를 개설하고자 하는 사용자가 HTML 작성법, 웹 저작 툴, CGI[1] 프로그래밍 등을 잘 알고 사용할 수 있어야 한다는 단점이 있다.

본 논문에서는 인터넷을 통하여 Unix 기반의 어플리케이션을 원격에서 자동설치/관리하기 위한 하부구조(infrastructure)로서 RASIS(Remote Automatic Software Installation System)를 개발하고, 이에 대한 적용 사례로서 웹 상에서 쉽게 강의지원 웹사이트를 자동생성하는 시스템을 설계하고 구현하였다. 본 시스템은 강의지원용 홈페이지를 구축하고자 하는 교사, 혹은 교수가 웹 브라우저를 통해 몇 가지 필요한 파라미터만 지정해 주면, 원하는 디자인과 기능을 모두 갖추고 있는 홈페이지가 지정된 웹서버에 생성되고, 설치된 홈페이지에 대한 정보가 강의지원 웹사이트 자동생성시스템에 연결된다. 추후 이러한 정보를 이용하여 생성된 웹사이트에 대한 업그레이드 및 패치, 웹사이트 디자인 변경, 타 응용과의 통신 등을 제공한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련기술에 대해 논하고, 3장에서는 인터넷을 통하여 Unix 기반의 어플리케이션을 원격에서 자동설치/관리하는 하부구조인 RASIS의 구성요소를 기술한다. 4장에서는 RASIS의 적용 예로서 강의지원 웹사이트 자동생성시스템의 설계 및 구현에 대해서 설명하고, 5장에서는 결과와 고찰, 그리고 6장에서 결론을 맺는다.

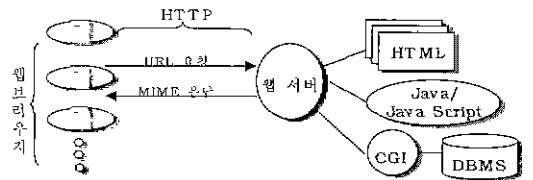
2. 관련 기술

2.1 웹(World Wide Web)

웹은 다양하고 광범위한 문서들로의 편리한 접근 방법을 제공해주는 분산 하이퍼미디어 정보 검색 시스템으로서, 초기에는 각종 연구 자료와 다양한 정보를 효율적인 방법으로 공유하기 위한 것이었다.

웹은 클라이언트와 서버간의 통신 프로토콜로 HTTP를 사용한다. HTTP는 분산 네트워크 환경에서 텍스트, 그래픽 이미지, 사운드, 비디오, 그리고 다른 멀티미디어 파일들을 교환하기 위한 프로토콜이며, 전세계에 분산되어 있는 정보를 하이퍼미디어 형태로 문서들

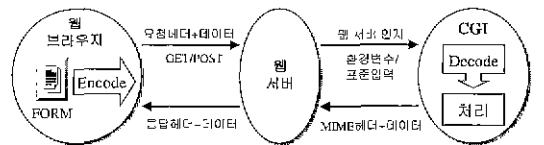
을 주고받을 수 있게 되어 있다. 웹은 클라이언트/서버 모델을 기반으로 동작하며, 클라이언트는 Netscape, Explorer 등과 같은 웹 브라우저를 통하여 웹 상에 있는 문서들을 액세스한다. 즉, 웹브라우저가 웹 서버에 요청을 보내고, 웹 서버는 MIME 형식에 따라 클라이언트가 요청한 문서들로 응답한다 이를 (그림 1)에서 나타낸다[3]



(그림 1) 월드 와이드 웹의 동작 개념

2.2 CGI (common gateway interface)

CGI는 외부 애플리케이션과 HTTP 또는 웹 서버와 같은 정보 서버를 연결하는 표준이다. 웹 서버가 정적인 문서를 일반적으로 제공하는 데 그치는 것이 아니라, CGI에 의해 사용자의 요구를 동적으로 처리할 수 있게 만들어진 것이다. 일반적으로 HTML 문서로 만들어진 정적인 문서는 에디터로 직접 수정하기 전에는 그 내용이 바뀌지 않는다. 반면에, CGI는 사용자의 요구에 따라 동적으로 문서를 생성하여 사용자에게 제공한다. (그림 2)는 이와 같은 CGI의 동작 원리를 나타낸다.



(그림 2) CGI의 동작 원리

CGI와 같이 웹에서 사용자의 요구를 동적으로 처리할 수 있는 유사한 기술로는 ASP(Active Server Page)[4], Javascript[5] Java Servlet[6] 등이 있다

2.3 HTML과 XML/SGML

웹은 HTML(HyperText Markup Language)을 기반으로 하여 전 세계의 모든 정보를 거대한 정보망의 형태로 구성하였으며, 다양한 '사이버 공간'을 통하여 누구나 손쉽게 다양한 정보를 습득/이해하게 해준다. 또

한 이것은 정보의 폭발적인 증가로 이어지고, 필연적으로 정보의 효율성 측면을 고려하게 하였다. 이러한 배경하에 정보의 생성단계에서부터 정보의 재사용을 고려하여 새로이 나타나게 된 언어가 XML(Extensible Markup Language)이다. 1998년 W3C에 의해 소개된 XML은 1986년 국제표준화된 SGML(Standard Generalized Markup Language)을 웹 환경에 맞도록 변형한 언어로 처리의 용이성을 추가한 언어이다[7]

HTML은 정보의 제시만을 목적으로 하는 언어(presentation only)로서 제작된 정보는 일회성을 갖는다. 이는 정보를 담은 그릇과 그 내용을 분리하지 않고 하나로 녹여 놓음으로 인해 생기는 비용의 추가 발생이다. XML/SGML은 정보를 표현하는 외형(physical structure or formatting information)과 그 내용(logical structure, content information)을 분리하여 정보를 구조화함으로써 정보의 전달/공유/재사용을 극대화하기 위해 제안된 언어이다. (그림 3-a,b)에서 같은 정보를 표현하기 위해 사용된 HTML과 XML의 문서를 비교하면 그 차이가 확연하게 드러난다.

```
<HTML><BODY>
<TABLE>
<TR><TD>자바프로그래밍</TD></TR>
<TR><TD>컴퓨터공학과</TD></TR>
<TR><TD>홍길동</TD></TR>
<TR><TD>김철수</TD></TR>
<TR><TD>이명희</TD></TR>
<TR><TD>장정수</TD></TR>
</TABLE>
</BODY></HTML>
```

(그림 3-a) HTML 문서

```
<?xml version="1.0"?>
<Course>
<Name>자바프로그래밍</Name>
<Department>컴퓨터공학과</Department>
<Teacher><Name>홍길동</Name></Teacher>
<Student><Name>김철수</Name></Student>
<Student><Name>이명희</Name></Student>
<Student><Name>장정수</Name></Student>
</Course>
```

(그림 3-b) XML 문서

(그림 3-a,b)에서 정보를 이해하는 입장에서 보면

XML 문서에 있는 '자바프로그래밍'이 '컴퓨터공학과'의 강좌명을 의미하고, '홍길동'이 강사라는 정보를 훨씬 쉽게 이해할 수 있다. 또한 위와 같은 문서들에서 강좌명과 강사들만을 따로 모아 리스트를 만들고자 한다면 HTML 문서는 추가 수작업이 항상 필요한 반면 XML 문서에서는 이를 프로그램이 자동으로 수행할 수 있다. 이렇게 구조화된 정보는 한번 생성된 정보를 추가적 작업없이 정보추출/전달/재생산 할 수 있게 한다. XML 문서에 대한 정보검색은 사용자로 하여금 자신이 원하는 정보를 세분화하여 검색할 수 있는 정보 표현의 확장성을 제공하여 대량의 정보에서 원하는 정보만을 빠르게 찾을 수 있게 한다. 예를 들면, '컴퓨터공학과에서 실시하는 자바프로그래밍이라는 강좌의 강사 리스트'와 같은 구조화된 검색이 가능해진다. 이와 같이 XML/SGML을 이용한 정보 구조화 효과는 다음과 같이 요약할 수 있다.

- 1) 정보생산비용 절감(정보재사용 극대화)
- 2) 정보관리비용 감소(정보관리 효율화)
- 3) 정보이용 극대화(보다 신속/저능적 검색)
- 4) 유연성, 확장성

2.4 원격실행 기술

2.4.1 shell script

셸 스크립트는 유닉스 기반 운영체제의 명령들의 시퀀스를 포함하는 텍스트 파일이다. 한번에 하나씩 키보드에서 시스템에 보내야 했던 명령들의 시퀀스를 하나의 파일로 된 스크립트로 합치기 때문에 셸 스크립트라고 불리어진다. 셸은 운영체제의 명령 해석기이고 시스템과 통신하기 위해서 사용하는 명령의 집합이다 [8]

2.4.2 rexec(Remote Execution)

rexec는 원격 호스트 상에 있는 명령이나 실행 가능한 파일을 원격으로 실행한다. 즉, 인증을 거친 후 원격 호스트 상에 있는 명령 혹은 프로그램을 실행하는 아주 간단한 방법이다. rexec가 실행된다는 것은 원격 호스트 상에 자동으로 프로세스들이 생성된다는 것을 의미한다. rexec를 사용하기 위해서는 사용자 ID와 패스워드가 필요한데, 이들은 원격 호스트에 대한 인증을 위해 사용된다. 그리고 rcc를 사용하려면, 원격

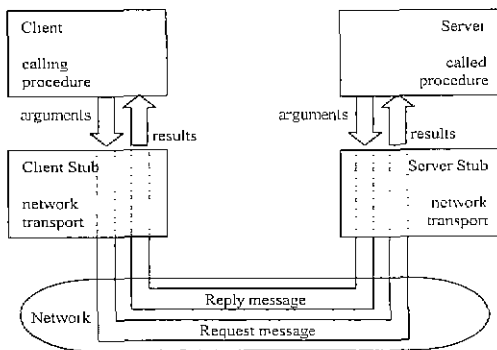
호스트 상에 반드시 rexecd 데몬이 동작하고 있어야 한다. 하지만, 이 인터페이스는 멀티스레드 애플리케이션에서는 불안정하기 때문에, 메인 스레드에서만 호출되어야 한다는 단점이 있다.

2.4.3 rsh(Remote Shell)

rsh는 원격 호스트에 연결하여 지정한 명령을 실행하기 위해서 원격 호스트 상에 원격 셸을 생성한다. rsh는 로컬 사용자 ID를 원격 사용자 ID로서 사용하기 때문에 사용자 ID와 패스워드를 필요로 하지 않는다. 원격 호스트 상에 로컬 호스트명과 사용자 계정명 /etc/hosts equiv 파일에 포함될 경우는 시스템 전반에 영향을 미칠 수 있고, 사용자 홈 디렉토리 상의 .rhost 파일에 포함될 경우는 특정 사용자의 권한만 사용할 있도록 제한을 줄 수도 있다. 하지만, rexec()처럼 rsh 또한 실행되기 위해서는 원격 호스트 상에 반드시 rshd 데몬이 동작하고 있어야 한다.

2.4.4 RPC(Remote Procedure Call)

RPC는 직접적인 통신과 적은 부하로 원격으로 프로그램들을 실행하는 기능을 제공한다. RPC 애플리케이션은 실제 프로시저를 실행하는 하나의 서버와 통신하는 하나의 클라이언트로 구성되어 있다[9].



(그림 4) 클라이언트/서버간의 RPC 통신

RPC를 이용하여 분산 응용 프로그램을 작성하여 실행할 경우 주프로그램과 원격 프로시저 사이의 통신을 관리하는 스텝(stub)이라는 코드가 자동으로 생성된다. 스텝은 RPC 프로토콜을 수행하고 메시지가 어떻게 만들어지고 교환되는지를 지정하는 통신 인터페이스로서,

이를 통해 클라이언트와 서버 프로그램이 연결된다 [10]. (그림 4)는 RPC를 이용한 클라이언트와 서버간의 통신을 나타낸다

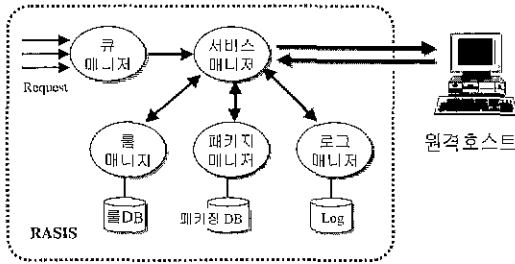
하지만, RPC를 이용하기 위해서는 스텝과 원격 프로시저 호출 런타임 라이브러리와 같은 원격 프로시저 호출 소프트웨어가 서버-클라이언트 간에 반드시 존재해야 한다. 또한 IDL(Interface Definition Language, 인터페이스 정의 언어)을 사용하여 통신 인터페이스를 정의하고, IDL 컴파일러를 통해 서버와 클라이언트에서 공통으로 사용하는 헤더파일, 클라이언트 스텝 코드와 서버 스텝 코드를 생성해야 한다. 이와 더불어 클라이언트 개발시 통신과정에서 발생해야 하는 예외 처리를 위한 제어 코드를 고려해야 하는 등 개발 환경이 용이하지가 못하다.

3. RASIS(Remote Automatic Software Installation System)

앞에서 살펴본 여러 기술들 중에서 RPC, CORBA [11]와 같은 높은 수준의 분산 응용 기술을 이용하려면 서버-클라이언트 측에 모두 관련된 소프트웨어가 반드시 필요하다. 또한 그러한 환경하에서 분산 응용 소프트웨어 개발하려면 상당한 전문적 지식이 요구된다. 본 논문에서는 이러한 서비스 사용자의 부가적인 환경 구축, 개발의 복잡성 문제들을 해결하기 위하여 RASIS를 제안한다. RASIS에서는 각 사용자 호스트의 실행환경은 기본적으로 서로 독립적이다. 즉 RPC나 CORBA 및 기타의 컴퓨팅 환경이 준비되지 않은 불특정 유닉스 사용자에게 대해서, RASIS는 웹 인터페이스를 통해 사용자가 원하는 특정 응용 S/W를 사용자의 사이트에 원격으로 자동 설치, 관리 및 유지보수를 가능하게 한다. RASIS는 사용자 호스트의 실행환경에 어떠한 수정도 요구하지 않고, 서버로 하이급 사용자에게 투명하게 각종 서비스를 제공할 수 있게 한다. 이러한 RASIS는 전자상거래, 소프트웨어 배포, 홈쇼핑 등 거의 모든 웹 솔루션에 적용할 수 있다.

3.1 RASIS의 구성요소

RASIS는 (그림 5)와 같이 크게 5가지의 주요 컴포넌트들로 구성된다

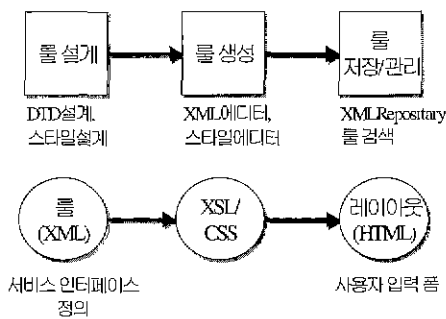


(그림 5) RASIS 구성요소

RASIS에서 각 컴포넌트의 역할은 다음과 같다.

3.1.1 룰(rule) 매니저 (인터페이스 매니저)

룰 매니저는 RASIS 응용 S/W가 적용될 서비스 룰을 생성/관리한다. 서비스 룰은 해당 응용 S/W를 배포/설치하기 위하여 입력받는 사용자 정보를 정의한다. 즉, RASIS 환경하에서 제작된 응용 S/W가 실행되기 위한 호스트 실행환경과 사용자 상세정보, 그리고 서비스 모듈 리스트 등을 정의한다. 이렇게 생성된 룰들은 룰 매니저에 의해 관리되고, 룰에 따른 서비스 인터페이스를 정의하기 위해 XML로 기술되며, XML 파서를 거쳐 룰 매니저에 등록/관리된다. 실제 RASIS를 통한 서비스는 이러한 각 서비스 룰이 헤딩 스타일시트(XLS/CSS)를 거쳐 HTML 문서로 사용자에게 맵핑된다. (그림 6)은 이러한 룰 매니저의 내부동작을 보여준다.



(그림 6) 룰 매니저의 내부동작

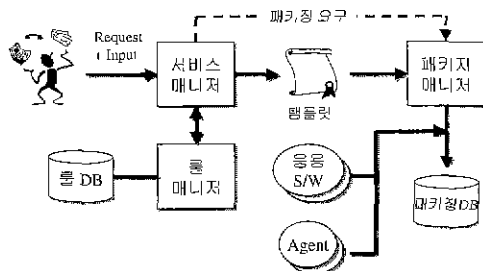
이와 같이 각 응용 S/W에 대한 독자적인 서비스 인터페이스 또는 룰이 다이내믹하게 HTML로 맵핑되어 사용자 입력 폼을 결정하게 된다. 이러한 기능을 수행하는 룰 매니저는 현재 구현 중에 있으며, 현 서비스는 정해진 룰에 대해 이미 만들어진 HTML 폼을 통해

이루어지고 있다.

RASIS는 패키지가 실제 배포되기 전, 각 패키지에 대하여 여러 사용자 템플릿들을 생성하게 된다. 여기서 사용자 템플릿이란 각 응용 S/W가 적용된 일종의 청사진과 같은 것으로서 서비스 룰에 대한 사용자별 인스턴스이다. 이것은 크게 실행환경 템플릿, 디자인 템플릿, 모듈 템플릿의 3종류로 구분된다. 실행환경 템플릿은 해당 응용 S/W가 실행될 사용자 호스트의 환경정보를 저장하며, 디자인 템플릿은 각 응용 S/W에 대해 사용자가 선택한 디자인 패턴 정보를 저장한다. 모듈 템플릿은 서비스되는 응용 S/W의 버전 및 사용자가 선택한 기능 모듈 정보 등을 저장한다. 이러한 템플릿들은 웹 상에서 사용자와의 상호작용을 통해 해당 서비스마다 미리 정의된 룰들을 기반으로 서비스 매니저에 의해 생성된다. 그리고 응용 S/W 배포/설치 시에 패키지 매니저에 의해 실제 응용 S/W와 함께 패키징된다.

3.1.2 패키지 매니저

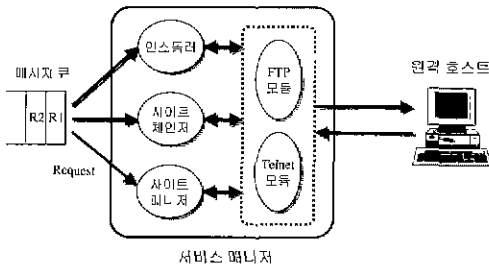
RASIS를 통해 서비스되는 응용 S/W 패키지에는 사용자 템플릿, 응용 S/W, 에이전트 모듈 등으로 구성된다. 여기서 에이전트 모듈은 RASIS의 서비스 관리를 목적으로 사용자 호스트 상에서 실행되는 모듈로서, 대표적으로 사이트관리 에이전트, 큐잉 에이전트 등이 있다. 서비스 매니저가 각 서비스마다 미리 생성된 서비스 룰에 따라 사용자 입력을 기반으로 템플릿들을 생성한 후, 패키지 매니저는 이 템플릿들을 기반으로 준비된 응용 S/W와의 패키징을 수행한다. 이때 패키지 매니저는 각 서비스 패키징 정보를 저장/관리하게 되고, 이를 이용하여 이후의 사용자 요구에 따라 서비스 매니저의 패키지 디자인 변경 패키지 업그레이드 등에 필요한 패키징을 수행한다. (그림 7)은 이러한 패키징 과정을 보여준다.



(그림 7) 각 서비스 요구에 따른 패키징 과정

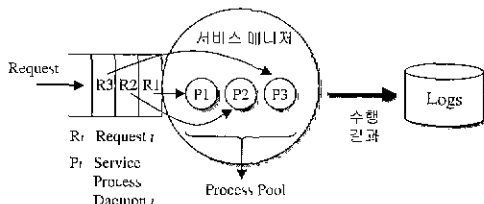
3.1.3 서비스 매니저

서비스 매니저는 (그림 8)에서와 같이 크게 5개의 모듈로 구성된다.



(그림 8) 서비스 매니저

서비스 매니저는 자동생성된 웹사이트들에 대한 유지보수(업그레이드 및 패치), 웹사이트 디자인 변경, 서비스 패키지 교체 등의 다양한 사용자 요구를 처리하게 된다. 기본적으로 웹을 통한 모든 사용자 요구는 RASIS의 메시지 큐로 보내지고, 서비스 매니저는 메시지 큐를 확인하고 각 요구를 해당하는 서비스 모듈로 전송, 처리를 한다. 각 서비스 모듈은 데몬 형태로 동작하고, 때 요청이 올 때마다 재사용하게 된다 이것은 기존 CGI 방식의 문제점인 프로세스 생성 및 관리 비용이 보다 효율적이고, 문맥 교환(context switch) 시간 감소 등의 장점을 가짐으로써, 보다 많은 사용자 요구를 수용할 수 있다. 또한 사용자 요구 및 처리 결과를 메시지 큐로 처리함으로써, 사용자의 요구를 비동기적으로 처리할 수 있다. 이것은 RASIS를 통해 이루어지는 대부분의 서비스(원격 응용 S/W 자동 설치/관리 등)가 다소의 시간이 걸리므로, 사용자의 대기시간(응답시간)을 줄이고 시스템의 수행성능을 최적화시켜준다. (그림 9)은 이러한 서비스 매니저의 동작을 보여준다.



(그림 9) 서비스 매니저 동작

서비스 매니저는 각 서비스 모듈들의 프로세스 풀 관리 및 서비스 모듈 추가/삭제 등의 작업을 수행한다.

서비스 매니저에 관리되는 서비스 모듈들은 다음과 같다.

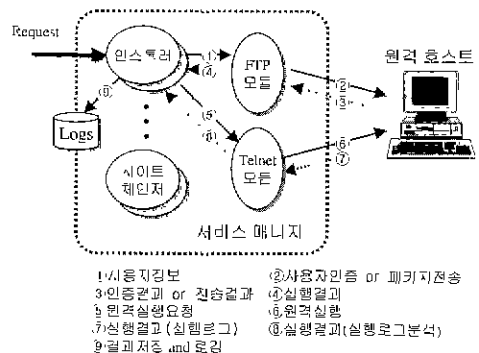
(가) Telnet 모듈

Telnet[12] 모듈과 FTP[13] 모듈은 다른 서비스 모듈을 지원하는 일종의 백그라운드 서비스 모듈이다. 이 중에서 Telnet 모듈은 주로 원격실행 명령을 수행한다. 인스톨러 등의 서비스 모듈이 FTP 모듈을 이용해 원격 호스트로 패키지를 전송한 후, 이 패키지를 인스톨하기 위해 Telnet 모듈로써 인스톨 에이전트를 실행하게 된다

기존의 원격실행기술인 rexec, rsh, RPC 및 CORBA 등을 이용하려면 반드시 서비스 양단에 같은 환경이 요구된다. rexec와 rsh는 시스템 보안상 주로 서버측의 데몬 프로세스를 실행시키지 않는 경우가 많으며, RPC나 CORBA는 대부분의 기본적인 유닉스 사용자 환경에 포함되지 않는다. 하지만, Telnet이나 FTP는 기본 인터넷 표준 서비스로서 대부분의 기본 유닉스 환경에는 반드시 포함된다. 그리하여 RASIS의 원격실행명령은 Telnet과 FTP를 이용한다. 현재 RASIS의 인스톨러는 Telnet 모듈이외에 rsh 모듈도 지원하고 있다. 차후의 버전에는 RPC와 CORBA와의 커넥션도 지원할 예정이다

(나) FTP 모듈

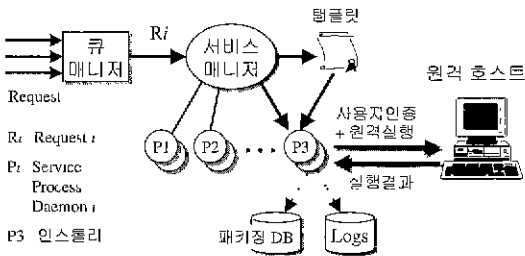
인스톨러 등의 서비스 모듈에서 FTP 모듈은 실행환경 탬플릿 내의 사용자 정보를 이용하여 실제 대상 호스트에 대한 적절한 사용자 인증을 수행한 후, 사용자의 호스트에 RASIS의 서비스 패키지를 실제 전송한다. 이렇게 전송된 패키지는 Telnet 모듈에 의해 해당 서비스 모듈의 에이전트를 구동함으로써 설치된다. (그림 10)은 이와 같은 Telnet 모듈과 FTP 모듈의 동작을 보여준다.



(그림 10) Telnet 모듈과 FTP 모듈

(다) 인스톨러

인스톨러는 RASIS에서 서비스되는 응용 S/W를 직접 사용자 호스트에 새로이 인스톨하는 모듈이다 (그림 11)은 인스톨러의 기본동작을 보여준다. 실행환경 템플릿 내의 사용자정보를 이용하여 FTP 모듈의 도움으로 대상 호스트에 대한 적합한 사용자 인증절차를 거친 후, 패키지 매니저에게 패키징 요구를 보낸다. 결과로 얻은 패키지를 FTP 모듈로 대상 호스트로 전송하고, Telnet 모듈을 이용하여 패키지에 포함된 인스톨 에이전트를 수행한다. 인스톨이 성공적이면, 패키지 매니저에 의해 관리되는 각 사용자의 패키징 DB에 수행 결과를 저장한다. 만일 인스톨이 실패하면, 로그 매니저에 의해 관리되는 log DB에 실패 원인과 실패 시점의 수행 단계 등을 저장한다. 이는 서비스 매니저 또는 사용자에 의한 재실행 명령시에 참조된다.



(그림 11) 인스톨러 동작과정

패키지 설치 작업외에도 인스톨러는 각 사용자 사이트마다 하나의 사이트 관리 에이전트, 큐잉 에이전트를 구동시킨다.

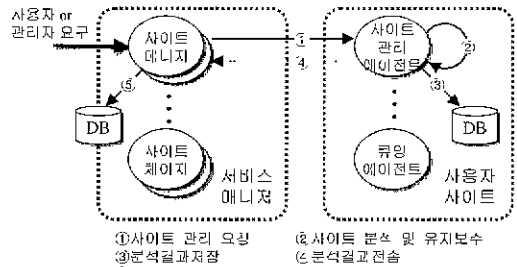
(라) 사이트 체인저

사이트 체인저는 사용자가 RASIS에서 제공하는 서비스 응용 S/W 사용시 사이트의 디자인 패턴 변경, 업그레이드 또는 패치 등과 같은 요구를 발생하였을 때 수행된다. 기본적인 동작은 인스톨러의 동작과정과 같지만, 기존의 사용자 패키징 DB와 인스톨 정보를 바탕으로 요구에 따라 새로이 변경 및 업그레이드될 자료들만 패키징해서 전송하는 것이 다르다. 이와 같이 패키지 매니저에서 패키징 DB 내의 이전 설치정보와 현재의 사용자요구에 따른 설치정보를 비교 분석하여 중복되지 않은 데이터만을 패키징하는 것을 차이분석 방법(deferencing analysis method)[14]이라 한다. 이것은 불필요한 데이터의 전송을 줄이고, 시스템 전체의

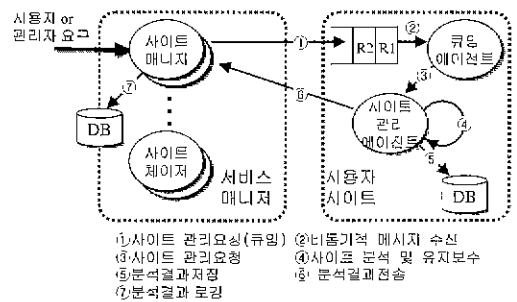
수행 성능을 높여준다.

(마) 사이트 매니저

사이트 매니저는 RASIS에 등록된 사용자의 사이트를 대상으로 사용자 옵션에 따라 동기적/비동기적으로 사이트 관리기능을 수행한다. 동기적인 관리는 원격 호스트에서 동작하는 사이트 관리 에이전트와의 상호작용을 통해 실시간으로 사이트 분석 및 유지보수를 수행하고, 그 결과를 분석하게 된다. 이것은 주로 RASIS의 관리자에 의해 각 사용자 사이트에 대한 통계, 서비스 유지보수 차원에서 수행할 수도 있다. 비동기적인 관리는 주로 사용자 사이트의 큐잉 에이전트를 이용하는데, RASIS의 사용자 사이트 분석 등의 관리 작업 요청이 사용자 메시지 큐에 입력되면, 이 큐를 관리하는 큐잉 에이전트가 비동기적으로 사이트 관리 작업을 사이트 관리 에이전트에게 요청하게 되고, 그 결과를 RASIS에 전송하여 관리하게 된다. (그림 12-a,b)는 이러한 사이트 매니저의 동작과정을 보여준다.



(그림 12-a) 동기적 사이트 관리



(그림 12-b) 비동기적 사이트 관리

3.1.4 큐 매니저

메시지 큐잉은 메시지를 이용하여 분산된 응용 프로

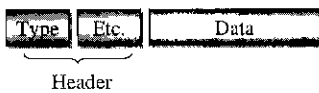
그럼 간에 비동기적(asynchronous)인 통신을 가능하게 하는 방법으로서, 서로 다른 상이한 응용 프로그램들 사이의 상호 연동을 가능하게 한다[15].

큐 매니저는 자동생성된 웹사이트들과 RASIS 간의 메시지 큐를 관리한다. 메시지는 사용자 요구 및 기타 작업에 의해 발생되어 지역(local) 큐에 들어가고, 각 메시지 타입에 따라 해당 서비스 모듈에 전송된다. 사용자 사이트의 큐잉 에이전트와 RASIS 시스템 내의 큐 매니저와의 근본적인 차이점은 메시지 타입의 정의에 있다. 큐 매니저는 제공하는 각 서비스에 따른 메시지 타입을 정의할 수 있으며, 또한 RASIS 시스템과 타 시스템의 응용과의 통신을 위해 새로운 메시지 타입을 정의할 수 있다. 일단 새로운 메시지 타입이 정의되면, RASIS 서비스 메시지 테이블이 다시 작성되고, RASIS내에 등록된 사용자 사이트로 이 정보가 전송된다. 그리하여 사용자 사이트 상에서 동작하는 큐잉 에이전트는 RASIS 서비스 메시지 테이블에 정의된 메시지 타입에 대해서 지정된 동작방식으로 동작하게 된다.

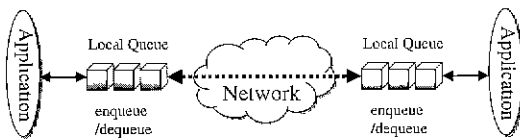
메시지는 다음과 같은 속성을 갖는 정보단위로 정의될 수 있다

- ① 목적지와 수신자
- ② 미리 정의된 전달정보
- ③ 우선 순위와 순서
- ④ 소멸 시간

이 외에도 메시지 데이터의 크기 등이 포함된다. (그림 13-a)는 이러한 RASIS의 메시지 포맷을 나타내며, (그림 13-b)는 메시지 큐잉 동작을 보여준다.



(그림 13-a) 메시지 포맷



(그림 13-b) 메시지 큐잉

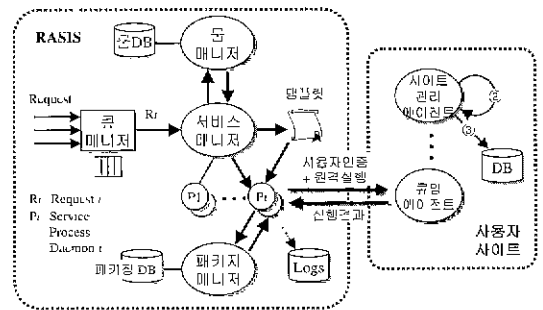
3.1.5 로그 매니저

RASIS에서는 매 사용자 요청에 의해 수행되는 작업

의 처리결과에 대한 로그를 기록하게 된다. 이러한 로그는 로그 매니저에 의해 관리되며, 수행결과 로그, 이벤트 로그, 시스템 에러 로그 등이 있다. 수행결과 로그는 수행한 작업에 대한 성공/실패에 대한 로그이며, 이벤트 로그는 주로 각 사용자 사이트에서의 이벤트 발생에 대한 로그로서, 사이트 디자인 변경 및 업그레이드 등의 사용자 요구에 의해 발생하게 된다. 시스템 에러 로그는 시스템의 오류, 네트워크 단절 등의 원인으로 수행 중지된 작업에 대한 로그이다. RASIS는 매 작업단위별로 서비스 양단 간에 발생된 로그를 체계함으로써, 작업의 동기화, 에러 복구 등의 기능을 수행하게 된다. 대부분 작업에 대한 처리 및 결과통보 등은 메시지 큐잉으로 해결할 수 있지만, 로그 매니저는 작업 단위별 트랜잭션 처리를 가능하게 한다.

3.2 RASIS의 실행구조 설계 및 응용 예

본 절에서는 RASIS의 각 컴포넌트들의 전체적인 상호작용 및 동작과정을 기술한다. (그림 14)는 RASIS의 전체동작에 대해 보여준다.



(그림 14) RASIS의 전체동작

본 논문에서는 앞에서 제안한 RASIS의 응용 예로서 강의지원 웹사이트 자동생성시스템을 구현하였다. 강의지원 웹사이트와 같은 특수한 목적의 홈페이지를 개설하기 위해서는 웹 저작 툴, HTML, CGI 프로그래밍 등에 관한 기술들을 교수 본인이 잘 알고 사용할 수 있어야 한다. 그리하여 대부분이 이러한 웹사이트를 구축하기에는 상당한 기술적/시간적 비용이 요구된다. 본 논문에서 구현된 강의지원 웹사이트 자동구축시스템은 이러한 어려움을 해결하기 위하여 원격지에서 원격실행을 통하여 응용 S/W를 자동설치/관리하는 RASIS를 이용하였고, 사용자는 단지 웹 브라우저에서

강의과목에 대한 파라미터들만 입력하면, 사용자의 웹사이트에 강의지원용 웹사이트를 자동생성할 수 있다.

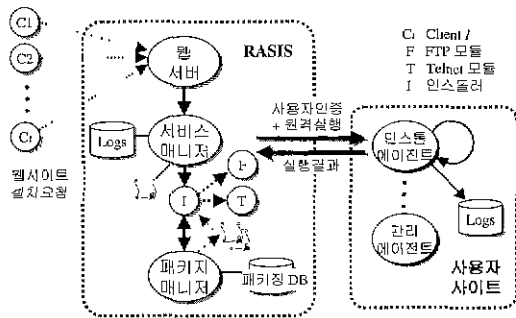
4. 강의지원 웹사이트 자동생성시스템

4.1 시스템 설계

본 시스템은 앞에서 제안된 RASIS의 주요 부분만을 적용해 보았다. 강의지원 웹사이트 설치요구는 RASIS의 서비스 매니저에 전달되고, 서비스 매니저는 새로운 웹사이트 구축요구인 경우 인스톨리가 이를 처리하게 한다. 현재 버전에는 메시지 큐잉을 지원하지 않으므로 사용자 요구에 대한 실행은 실시간으로 이루어진다. 웹 상의 HTML 폼을 통한 사용자 입력정보와 해당 서비스마다 미리 정의된 룰들을 기반으로 템플릿들이 생성되고, 이 템플릿들은 인스톨러의 패키징 요구시에 사용된다.

인스톨러는 패키지 매니저로부터 받은 패키지를 적절한 사용자 인증길치를 거친 후, FTP 모듈을 이용해 사용자 사이트로 전송하고, Telnet 모듈로서 패키지 내의 인스톨 에이전트를 구동시킨다. 인스톨 에이전트는 함께 전송되어온 템플릿 정보를 이용하여 자동으로 웹사이트 설치작업을 수행하며, 그 수행결과를 로그에 저장함과 동시에 RASIS의 인스톨러로 전송한다. 결과를 전송받은 인스톨러는 수행결과를 자체 로컬 로그 DB에 저장하고, 적절한 사용자 응답정보를 생성한 후, 웹 서버에 이를 리턴한다.

(그림 15)는 이와 같은 강의지원 웹사이트 자동생성 시스템의 동작을 보여준다.



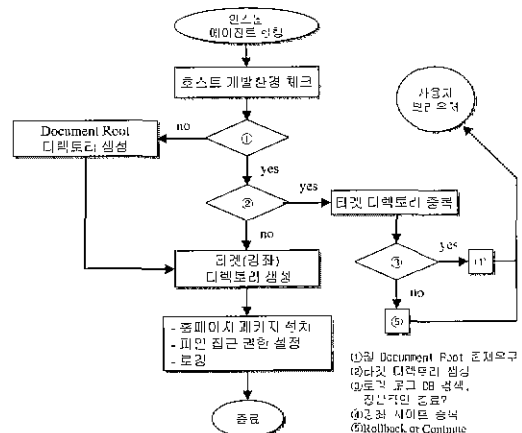
(그림 15) 강의지원 웹사이트 자동생성시스템

사용자가 자신이 속해 있는 로컬 호스트에 강의지원 웹사이트를 설치하기 위해 본 시스템의 URL로 접속하

면, 현재 제공하고 있는 서비스를 사용자에게 보여준다 여기서 사용자는 웹 브라우저 상에서 홈페이지 설치 요청을 한다. 이 때 사용자가 입력한 사용자 정보(파라미터)로는 사용자가 속해 있는 로컬 호스트의 계정과 패스워드, 그리고 강의정보를 제공하고자 하는 과목에 대한 정보(학과, 수강대상, 과목코드 등)들을 포함한다. 강의지원 웹사이트 자동생성시스템은 사용자 입력 오류를 검사한 후 사용자 계정과 패스워드로 사용자 인증을 수행하고, 액세스가 수락되면 홈페이지를 설치하는 데 필요한 사용자 정보가 있는 템플릿들, 인스톨 에이전트 및 기타 에이전트, 배포판 응용 S/W 등을 패키지로 하여 FTP 모듈을 통해 전송한다 전송이 완료되면, 본 시스템은 Telnet 모듈을 이용해 인스톨 에이전트를 실행하여 자동으로 홈페이지를 구성하고 자동설치된 홈페이지 URL을 본 시스템의 게시판에 연결한다. 이러한 일련의 과정으로 설치가 완료되면 설치 결과 정보가 웹 브라우저에 전달된다

4.2 에이전트 모듈

에이전트 모듈은 RASIS의 매니저 모듈과 유사한 작업을 수행하는데 크게 2가지로 구분된다. 인스톨 에이전트와 같은 1회성의 단순실행 에이전트와 사이트 관리 에이전트, 큐잉 에이전트 등과 같이 데몬 형태로 동작하는 데몬 에이전트가 있다. 단순실행 에이전트는 주로 셸 스크립트로 구현이 되며, 데몬 에이전트는 C나 C++로 구현이 되어 사용자 사이트에서 컴파일되고 실행된다 (그림 16)은 인스톨 에이전트의 동작을 나타낸다



(그림 16) 인스톨 에이전트의 동작

인스톨 에이전트는 크게 호스트 개발환경 검사, 디렉토리 검사, 홈페이지 설치 및 권한 설정의 3부분으로 구성되어 있다. 홈페이지가 설치될 원격 호스트에 배포판 S/W의 전송이 완료되면, 인스톨리에 의해서 인스톨 에이전트가 실행된다. 인스톨 에이전트가 실행되면 제일 먼저 하는 작업이 호스트 개발환경 체크로서, OS의 종류와 버전, 컴파일러 등의 여러 정보들을 파일로 생성한다. 이 정보는 기타 다른 에이전트의 설치 및 구동시에 이용된다. 이 작업이 완료되면 디렉토리 검사를 수행하는데, 웹 서버의 Document Root 디렉토리가 각 사용자의 홈 디렉토리에 존재하는가를 검사한다. Document Root 디렉토리는 사용자의 홈페이지 서비스를 위해 반드시 필요한 디렉토리로서 웹서버에 의해 설정된다. 다음은 Document Root 디렉토리(예, /htdocs) 내에 사용자 강의과목 코드(예, km307)와 일치하는 디렉토리(예, /htdocs/km307/)가 존재하는지 여부를 검사한다. 사용자 강의과목 코드와 같은 디렉토리가 이미 있다면 2가지를 고려할 수 있다. 사용자 호스트에 이미 그 강좌 사이트가 개설이 되어있던가 아니면 장애(시스템 에러, 네트워크 단절 등)로 인해 설치 작업이 비정상적으로 종료된 경우이다. 후자의 경우는 로컬의 Log DB를 검색하여, 분석함으로써 에러의 원인 및 작업 진행 선택(Rollback or Continue) 등을 사용자에게 브라우징한다.

되면 그 결과가 파일로 인스톨리에 전송된다. 이 부분은 현재 구현중인 메시지 큐잉 시스템이 완성되면, 메시지 큐로 대체하게 될 것이다. (그림 17)은 웹사이트 자동생성시스템의 프로시저 다이어그램을 나타낸다.

4.3 시스템 구현 환경

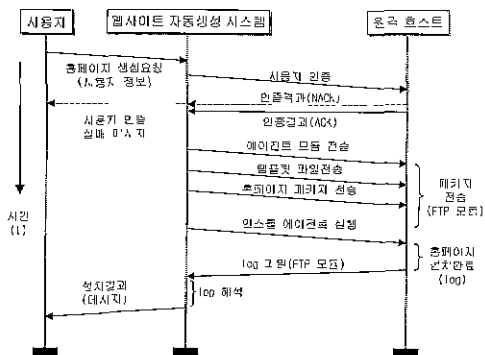
본 논문의 웹사이트 자동생성시스템의 동작 플랫폼은 UNIX 환경이며, 웹 상의 서비스를 주로 처리한다. 설치된 하드웨어는 UltraSparc 워크스테이션 2대를 사용하였다. 하나는 이 웹사이트 자동생성시스템(nety)으로, 다른 하나는 홈페이지가 자동설치될 원격 호스트(ynucc)로 사용하였다. 운영체제는 Solaris 2.5.1이고, 웹 서버는 CERN 서버와 Apache 서버를 사용하였다. 사용된 C 컴파일러는 SunSoft사의 SPARCwork 3.1 버전을 이용하였다.

본 시스템에서 구현된 인스톨 에이전트는 rsh 이라는 원격 실행명령도 지원하는데, rsh 명령이 실행되기 위해서는 원격 호스트에 별도의 환경설정이 필요하므로 불특정 다수를 대상으로 서비스하기 위해서 주로 Telnet 모듈을 이용하여 원격실행 명령을 수행한다.

5. 결과 및 고찰

본 논문에서 제안한 RASIS는 기존의 rsh, RPC, CORBA와 같은 원격실행기술의 문제점을 해결하고, 인터넷을 통한 Unix 기반의 어플리케이션을 원격에서 자동설치/관리하는 하부구조를 제공한다. RASIS를 이용하게 되면 손쉽게 다양한 웹 상의 서비스를 구현할 수 있다. 이에 대한 적용 사례로서 웹 상에서 사용자와 상호작용을 통한 강의지원 웹사이트 자동생성시스템을 설계 및 구현하였다.

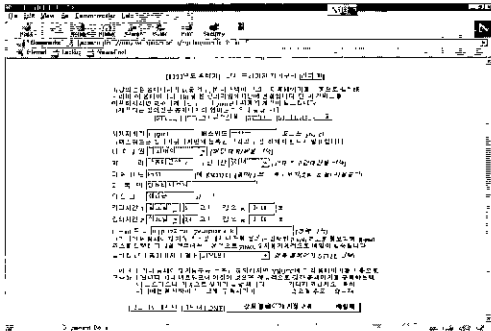
본 논문에서 RASIS를 적용하여 구현된 강의지원 웹사이트 자동생성시스템은 현재 영남대학교(www.yeungnam.ac.kr) '강의지원'을 통해서 시험적으로 서비스되고 있으며, 강의지원 홈페이지를 개설하고자 하는 사용자들이 HTML, CGI 프로그래밍 등과 같은 웹 서비스 관련 기술을 전혀 습득하고 있지 않더라도 쉽게 본인의 원격호스트에 강의지원 웹사이트를 구축할 수 있다. 현재 제공되는 강의지원 웹사이트(강의지원 웹사이트 패키지)는 4종류로서, 기본적으로 '강의계획서', '강의자료', '강의게시판', '강좌관련사이트' 기능 모듈들로 구성된다. 추가적인 기능 모듈로서 실시간 채팅 등을 제공할 예정이다.



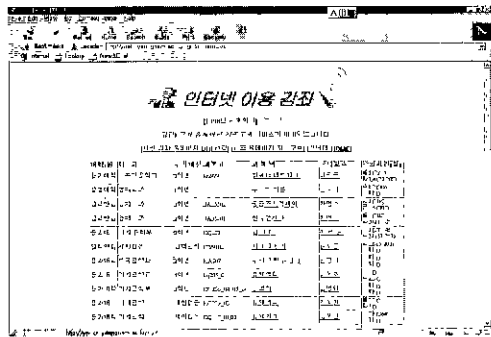
(그림 17) 웹사이트 자동생성시스템의 프로시저 다이어그램

Rollback의 경우는 인스톨 전과정을 새로이 실행하고, Continue를 선택한 경우는 작업이 완료되었던 시점부터 설치 작업을 진행한다. 이 작업이 성공적으로 수행되면, 에러가 발생했던 작업의 로그를 갱신한다. 인스톨 에이전트에 의해 모든 작업이 성공적으로 수행

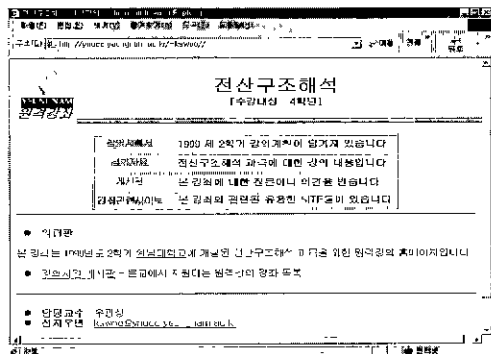
(그림 18-a)는 홈페이지를 설치하기 위해 사용자 및 원격호스트 정보를 입력하는 예이고, (그림 18-b)는 강의용 홈페이지 설치 후 강의리스트에 자동으로 추가된 모습이다. 그리고 (그림 18-c)는 설치된 강의지원 웹사이트를 나타낸다.



(그림 18-a) 설치를 위한 사용자 정보 입력 예



(그림 18-b) 설치 후의 강의리스트
(http://nety.yeungnam.ac.kr/cgi-bin/rem_1ec)



(그림 18-c) 설치된 강의지원 웹사이트

6. 결 론

본 논문에서는 이기종 환경의 분산 컴퓨팅 환경에서 원격실행 기술에 의한 응용 S/W를 원격에서 자동설치/관리하는 하부구조로서 RASIS를 제안하고, 이의 적용사례로써 강의지원 웹사이트 자동생성시스템을 구현해 보았다

본 논문에서 제안한 RASIS를 이용하면 강의지원 웹사이트 자동생성시스템의 사례와 같이 다양한 목적의 홈페이지를 자동으로 구축해 주는 엔진을 개발할 수 있으며, 기존의 다양한 시스템과도 쉽게 통합할 수도 있다.

향후 연구과제로는 메시지 큐잉을 이용한 보다 효율적인 통신 방법 연구와 에이전트 모듈의 이동성을 지원하는 테카니즘 연구가 남아있다.

참 고 문 헌

- [1] Stathes P. Hadjicthymades, Drakouhs I. Martakos "Improving the performance of CGI compliant database gateways," The 6th International World Wide Web Conference, 1997.
- [2] Shishir Gundavaram. 'CGI Programming on the World Wide Web', O'Reilly & Associates, Inc., 1996.
- [3] 송규백, 임인택, 정영석, 임경수, 김종근, "원격실행 기술을 이용한 자동 홈페이지 생성 메커니즘", '98 멀티미디어학회 학술대회, 1998 12.
- [4] Alex Homer, 'PROFESSIONAL ASP Techniques for Webmasters', 김보문화사, p 719. 1999.
- [5] D. Flanagan. 'JavaScript The Definitive Guide. O'Reilly & Associates', Inc., 1997.
- [6] K.M. Chandy, A. Rifkin, P.A.G. Sivillotti, J. Mandelson, M. Richardson, W. Tanaka, and L. Weisman. "A World-Wide Distributed System Using Java and the Internet," In IEEE International Symposium on High Performance Distributed Computing, Syracuse, NY, August 1996.
- [7] Tim Bray and C.M. Sperberg-McQueen, "Extensible Markup Language (XML): Part I Syntax," World Wide Web Consortium Recommendations,

1998. 2. 10, <http://www.w3.org/TR/REC-xml>

[8] 조경산, '프로그래머와 사용자를 위한 UNIX 완성', 이한출판사 1994.

[9] Sun Microsystems, Inc. "RPC: Remote Procedure Call Protocol Specification," RFC 1050 April 1988.

[10] John Bloomer, 'Power Programming with RPC', O'Reilly & Associates, Inc 1995

[11] Michi Henning, "Binding, migration, and scalability in CORBA," Commun. ACM 41, 10 (Oct. 1998), pp.62-71.

[12] J. Postel, J. Reynolds "TELNET PROTOCOL SPECIFICATION," RFC 854, May 1983.

[13] J Postel, J. Reynolds, "FILE TRANSFER PROTOCOL(FTP)," RFC 959, ISI, October, 1985.

[14] D.M. Ludlow, "Compare process for quick determination of text changes," IBM Technical Disclosure Bulletin 22(8A) (January 1980).

[15] Eric A. Brewer, Frederic T. Chong, Lok T. Liu, Shamik D. Sharma, and John D. Kubiatowicz. "Remote queues exposing message queues for optimization and atomicity," Papers of the 7th annual ACM symposium on Parallel algorithms and architectures, 1995, p.42.



임 인택

e-mail : ileem@lfc.ac.kr
 1996년 영남대학교 건축공학과 졸업(공학사)
 1998년 영남대학교 대학원 멀티미디어 통신학과 졸업(공학석사)

2000년 영남대학교 대학원 컴퓨터공학과 박사과정 수료
 2000년~현재 대구미래대학 멀티미디어정보과학과 전임강사
 관심분야 : 멀티미디어, 정보통신, 차세대 인터넷 등



송 규백

e-mail : kbsong@yu.ac.kr
 1991년 영남대학교 전산공학과 졸업(공학사)
 1998년 영남대학교 컴퓨터공학과 졸업(공학석사)
 2000년~현재 영남대학교 전산정보원 근무

관심분야 : 웹 프로그래밍, 데이터베이스, 네트워크 등



김 재 일

e-mail : jikim@munkyoung.ac.kr
 1984년 경북대학교 전자공학과 졸업(공학사)
 1984년~1988년 (주) LG 전자 연구원
 1992년 금오공과대학교 산업대학원 전자공학과 졸업(공학석사)

1998년 영남대학교 대학원 멀티미디어 통신공학과 박사 수료

1996년~현재 문경대학 컴퓨터정보과 조교수
 관심분야 : 원격교육, 분산처리, 망관리, 멀티미디어 통신 등



김 종근

e-mail : cgkim@ynucc.yeungnam.ac.kr
 1981년 영남대학교 전자공학과 졸업(공학사)
 1987년 영남대학교 대학원 전자공학과 졸업(공학석사)
 1991년 일본 東京電氣通信大學 정보공학과 졸업(공학박사)

1996년~1997년 Virginia Tech 방문교수
 1991년~현재 영남대학교 컴퓨터공학과 교수
 관심분야 : 분산처리시스템, 컴퓨터네트워크, 멀티미디어 정보통신, 멀티미디어 가상교육 등