

인트라넷 구축을 위한 WEB-TO-HOST 게이트웨이

박나연*, 유남현, 윤보열, 이성재, 김원중
 cookie@cs.sunchon.ac.kr, 순천대학교 컴퓨터학과

The WEB-TO-HOST Gateway For Intranet

Na yeon Park*, Nam Hyun Yoo, Bo Yeol Yoon, Sung Jae Lee, Wong Jung Kim
 Dept. of Computer Science, Suncheon National University

요 약

현재의 정보처리시스템은 대형 컴퓨터 중심의 중앙 집중형 처리 환경, 클라이언트/서버의 분산처리시스템에서 웹 환경의 인트라넷 환경으로 빠르게 변화하고 있다. 인트라넷 환경은 웹 브라우저라는 사용자 중심의 통합된 환경, TCP/IP, HTTP, HTML, XML 등의 표준화 된 기술을 사용한 플랫폼 독립적인 사용환경, 비용의 효율성, 확장 및 유지보수의 용이성 등을 제공한다. 그러나 현재 사용중인 호스트 환경을 완전히 무시하고, 인트라넷 환경을 구축한다는 것은 매우 비현실적이다. 따라서 기존 환경은 그대로 유지하면서 인트라넷을 구축하기 위한 방법이 필요하다. 본 논문에서는 사용중인 기존 시스템은 전혀 변경하지 않고 호스트의 터미널 환경을 그대로 인트라넷 환경으로 바꾸기 위한 WEB-TO-HOST 게이트웨이에 대한 연구이다.

1. 서 론

정보처리시스템은 대형 컴퓨터 중심의 중앙 집중형 처리 환경, 클라이언트/서버 환경의 분산처리시스템으로, 그리고 지금의 인트라넷 환경으로 발전하여 왔다. 중앙집중형 시스템은 호스트 시스템의 과중한 부하, 느린 처리속도, 폐쇄적 아키텍처, 신뢰성, 업그레이드 비용 및 유지보수 비용, 환경의 변화에 대한 신속한 대응, 다양한 응용프로그램 개발 등에 있어서 많은 문제점을 가지고 있지만, 통합적인 업무 환경의 제공, 정보의 효율적 관리, 시스템 운영의 안정성, 보안성 등에서는 매우 뛰어나다.

중소형 컴퓨터와 네트워크를 기반으로 하는 클라이언트/서버의 분산처리 환경은 정보이용의 효율성, 자료의 무결성, 네트워크의 안정성 및 처리속도, 수 많은 클라이언트 프로그램 관리 등의 문제를 가지고 있으나, 화려한 그래픽 환경의 사용자 인터페이스(GUI), 사용의 편의성, 오픈 아키텍처, 다양한 애플리케이션의 통합, 초기 투자비용의 절감과 같은 많은 장점을 가지고 있다.

인트라넷은 중앙집중형 시스템과 클라이언트/서버 환경의 분산시스템이 지니고 있는 장점들을 수용한 정보시스템 환경이다. 즉, 웹 브라우저라는 사용자 중심의 통합된 환경, TCP/IP, HTTP, HTML, XML 등의 표준화된 기술을 사용한 플랫폼 독립적인 사용환경, 비용의 효율성, 확장 및 유지보수의 용이성 등을 제공한다. 따라서 효율적인 인트라넷 구축에 대한 전략적인 연구와 접근이 필요하다[2,3,4].

II. 연구의 필요성

처음부터 인트라넷을 이용하여 정보시스템을 구축하면 별 문제가 없으나, 수년 동안 사용된 메인프레임 중심의 정보처리 환경을 인트라넷 환경으로 바꾸기 위해서는 기존에 개발된 모든 어플리케이션을 새로 개발하여야 하는 문제에 봉착하게 된다. 따라서, 중앙집중형 환경에서 개발된 모든 어플리케이션을 전혀 변경하지 않으면서, 기존의 호스트 터미널 사용자들에게 웹 브라우저와 같은 편리한 인터페이스를 제공하여야 한다.

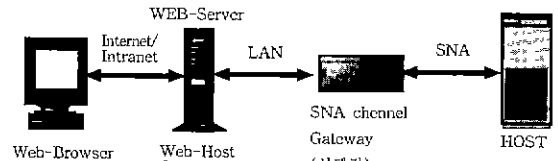
또한, 기존 호스트와 터미널 사용자 사이에는 전용선으로 연결하여 각자 독자적인 통신 프로토콜을 사용하였기 때문에 많은 유지비용이 소요되었다. 그러나 호스트의 정보처리환경을 전혀 변경하지 않고, 호스트 터미널의 정보내용을 HTML 환경의 웹 페이지로 변환하면 전용선 대신 TCP/IP 프로토콜의 인터넷망을 통하여 소비비용을 대폭 절감 할 수 있다 [표1]은 호스트의 터미널과 인터넷의 웹 브라우저 사용환경을 비교한 것이다[1,5,6].

	Terminal	Web Browser
사용자인터페이스	사용 호스트에 따라 각기 다름	웹 브라우저로 통일
정보검색	Host의 Application에 의존 (Function Key 등 사용)	하이퍼 링크로 편리함
프로토콜	Host에 따라 다름	TCP/IP, HTML로 통일
연결성	세선에 의해 연결성 유지	동적인 즉, 연결성 지인 못함
성보재검색	Host와 지속적인 상호 동기화	로컬 캐싱 및 정보검색기능 (푸워드/백워드 버튼 등 사용)

[표 1] Host Terminal 과 Web Browser의 차이점

III. WEB-TO-HOST 시스템

WEB-TO-HOST 접속 시스템은 웹 브라우저, 웹 서버, 웹-호스트 게이트웨이, 호스트의 4가지 구성요소들간의 상호작용에 의해 수행된다. 웹-호스트 게이트웨이는 시스템에서 가장 중요한 요소로서 웹 브라우저와 호스트에 연결된 터미널 사이의 정보를 상호 연동시키는 역할을 수행한다. 즉, 호스트의 표준 터미널 데이터를 HTML, 자바 코드로 변환하거나, 웹 브라우저로 정보를 전송할 수 있는 새로운 웹 어플리케이션을 생성하는 역할을 한다. 또한, 웹 브라우저에서 사용자의 입력이나 어플리케이션 수행요구를 호스트에서 처리할 수 있도록 표준 터미널의 형태로 변환하는 작업을 수행한다. 따라서 기존의 웹 브라우저, 웹 서버 그리고 호스트의 사용환경을 전혀 변경하지 않으면서도 두 환경을 상호연동 시킬 수 있다. [그림 1]은 WEB-TO-HOST 시스템의 구조이다.



[그림 1] WEB-TO-HOST 시스템 구조

WEB-TO-HOST 시스템의 구축을 위해서는 다음과 같은 문제점에 대한 해결이 선행되어야 한다.

3.1 프로토콜 매핑과 데이터 스트림의 변환

웹 서버와 웹 브라우저간에는 HTTP 프로토콜을 사용하고, 웹 서버간의 연결에는 TCP/IP 프로토콜이 사용되고 있으며, 웹 브라우저는 HTML를 사용하여 정보를 표현한다. 호스트와 터미널 사이는 호스트의 각종 특정한 프로토콜을 사용하기 때문에 각기 다른 에뮬레이터를 사용하여 지원한다. 따라서 이들을 상호연동 시키기 위해서는 프로토콜의 매핑과 데이터 스트림의 변환이 이루어져야 한다.

3.2 웹과 호스트의 연결상 차이점

네트워킹 기술에서 값이 싸고 기능이 뛰어난 단말을 사용하여 호스트에 접근하고자 하는 것은 모두의 희망이고, 이것을 성취하기 위해 많은 게이 트웨이와 프로토콜 변환 소프트웨어가 탄생했다. 한 예로 IBM의 SNA 초창기에 ASCII 단말은 아주 값이 싸고 널리 사용됐다. 그 결과로 ASCII 단 말은 IBM 호스트의 3270 디스플레이 스테이션 기능을 지원하기 위해 프로 토콜 변환기를 탄생시켰으며, 이것은 오늘날 소프트웨어 업체들이 값싼 웹 브라우저로 호스트에 접속하려는 상상과 매우 유사하다. 그러나 게이 트웨 이 또는 프로토콜 변환기를 사용할 때, 어떤 환경에서는 사용 가능한 기능 과 특징이 다른 환경에서는 적용되지 않을 수 있으며, 이것은 웹 브라우저

와 호스트 인터페이스에서도 적용된다. 즉, 웹 인터랙션은 동적(Dynamic)이고, 정보 검색을 위해서는 하이퍼 링크(포인트&클릭) 기능을 이용하고 있으나, 호스트의 터미널은 끊임없는 세션을 불러오는 형태이고, 정보 검색도 키보드(keyboard Interaction)와 호스트에 있는 애플리케이션에 크게 의존한다.

3.3 웹 브라우저와 터미널의 키보드 매핑

웹 브라우저와 호스트 터미널 환경간의 근본적인 차이점은 웹 사용자는 하이퍼링크를 사용하여 정보 검색을 할 수 있는 반면, 호스트의 단말은 기능을 사용해 호스트의 어플리케이션 범위 안에서 정보를 검색한다. 웹 브라우저는 터미널에서 제공하는 Enter,Tab,Esc, PF(Program Function Keys), PA(Program Access Kcys), Clear, Sysreq와 같은 기능키 등을 지원하지 않기 때문에 이것을 지원하는 방안을 강구하여야 한다.

3.4 브라우저와 호스트 어플리케이션의 동기화

웹 브라우저를 호스트의 터미널처럼 보여줄 때, 중요한 문제점 중의 하나가 사용자와 브라우저 사이의 작동을 호스트의 어플리케이션과 동기화시켜 주는 것이다. 호스트의 터미널은 어플리케이션과 긴밀한 동기화를 유지하지만, 웹 브라우저는 웹 서버와 HTTP 프로토콜로 느슨하게 결합되어 있다. 즉, 웹 브라우저의 웹 서버 사이에서의 통신은 각각 독립된 TCP 접속이다. 따라서 브라우저와 웹 서버 간의 통신을 동기화해 이것을 호스트의 세션(Session)으로 매핑하여야 한다.

3.5 웹 브라우저와 호스트 에뮬레이터의 차이

호스트의 단말은 근본적으로 터미 터미널이다 따라서 터미널은 진적으로 접속한 호스트의 어플리케이션에 의해 통제를 받는다. 호스트의 어플리케이션은 할당 소속 단말들을 제어하기 때문에 음동성 없이 정해진 규칙대로 상호 동기성을 유지한다. 그러나 웹 브라우저는 성능을 향상시키기 위해 로컬 캐시 기능을 이용한다. 웹 서버로부터 정보를 가져오면 일단 요속으로 로컬 디스크에 저장(캐시)한다. 이러한 방법을 통해 같은 정보를 요청하면 웹 서버와 무관하게 캐시된 정보를 보여준다. 또한 대부분의 브라우저는 로컬 정보검색 기능을 가지고 있다 사용자가 포워드/백워드 버튼을 사용하면서 웹 페이지를 검색하는 것은 전통적인 호스트와 터미 터미널 간의 통신방법과는 많은 차이가 있다. 이러한 특징이 브라우저와 호스트 사이의 동기화를 유지하는데 문제를 발생시킨다.

IV. 시스템 구조 및 각 구성요소의 기능

시스템은 크게 Session Monitor, CT-Emulator(Collector & Translator Emulator), Refiner, GS-Browser(Gather & Scatter Browser)등의 4가지 시스템으로 구성되어진다. Session Monitor는 사용자의 웹 브라우저와 그에 대응하는 Host의 터미널 화면의 동기화를 위한 것이다. CT-Emulator는 Host와의 인터페이스를 담당하며, Refiner는 호스트와 사용자 웹 브라우저 간의 자료 교환을 위한 화면 구성을 주로 담당하며, GS-Browser는 사용자 웹 브라우저와의 인터페이스를 담당한다.

4.1 Session Monitor

웹에서의 작업과 터미널에서의 가장 큰 차이점은 사용자 영역에 대한 관점이다. Web을 구현하기 위한 HTTP 프로토콜은 사용자 영역(Session) 정보를 유지하지 않는다. 그러나, 터미널 환경은 접속에서부터 종료될까지 계속 호스트와 연결되어 사용자 영역(Session) 정보를 유지한다. 이러한 차이점을 위하여 웹 브라우저로 전송되어지는 HTML문서에 Session ID와 Sequence ID를 Input Field의 Hidden Filed 형태로 전송한다.

Session ID는 사용자의 사용자 영역 정보를 유지하기 위한 것으로서 터미널 화면과 사용자 브라우저간의 동기화를 유지한다. Session Monitor는 각 Session ID별로 해당 터미널 화면의 정보를 Refiner의 버퍼에 저장하고 있다. 웹 브라우저에서 전달되어지는 정보 중 Session ID가 일치하는 정보와 매칭 작업을 함으로써 웹브라우저와 터미널간의 사용자 영역(Session) 정보를 유지한다.

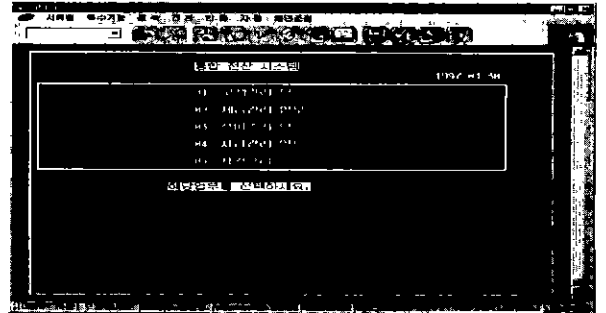
웹 환경에서의 사용자 인터페이스 중 가장 많이 사용되는 것이 'Back', 'Forward', 'Reload'의 클릭 및 URL 직접 입력 등이다. 이러한 웹 환경에 익숙한 사용자들은 WEB-TO-HOST 시스템에서 변환된 웹 화면에서도 습관적으로 Back, Forward 기능 등을 사용하게 된다. 이러한 행동은 오직 작동 프로그램에 정의된 키 값을 사용하는 터미널 화면과 사용자의 웹 브라우저간의 정보의 일관성 및 정확성에 영향을 미치게 된다. 즉, Session Monitor에 저장되어 있는 Session ID별로 정의된 화면과 사용자의 Back, Forward 기능 사용으로 변형된 웹브라우저 정보와의 결합작업은 잘못된 연산이나 결과를 유발할 수 있다. 이러한 문제점을 해결하기 위하여 사용될 수 있는 첫 번째 방법은 사용자의 웹 브라우저 화면에서 'Back', 'Forward', 'Reload' 클릭을 사용하지 못하게 하는 교육 및 직접 URL을 입력하지 못하게 하지만, 브라우저에서 해당 메뉴바를 아예 제거하는 방법이 있다. 두 번째 방법은 첫 번째 방법을 보완하는 것으로서 프로그램의 수행 순서 정보를 저장하고 있는 Sequence ID를 사용자에게 전달되는 HTML문서에 Session ID와 같이 포함시키는 것이다. 그래서, 사용자에게 전송하는 웹 화면

에 Session ID와 Sequence ID를 전송하여 시스템에서 유지하는 않는 Session ID와 Sequence ID를 가지는 화면이 WEB-TO-HOST 시스템으로 전송될 때는 호스트와의 연결을 강제 종료하고 사용자에게 여러 화면을 전송한다

4.2 CT-Emulator(Collector & Translator Emulator)

호스트의 터미널은 지정된 프로토콜을 사용하여 호스트와 직접 데이터를 주고 받는다. CT-Emulator는 사용자가 웹 브라우저를 사용할 때, 터미널처럼 접속하여 데이터를 주고 받는 일종의 Agent이다. CT-Emulator의 역할은 두 가지로 나타낼 수 있다. 첫 번째로는 Refine Script에서 넘어온 터미널 화면을 사용자에게 전달해주는 역할이며, 두 번째는 호스트에서 넘어오는 터미널 화면을 Attribute들을 제거한 Text형태의 터미 화면과 속성 등을 검색하여 구축한 테이블을 Refiner에게 넘겨주는 역할을 한다.

터미널 환경에서 사용되는 Attribute는 Alpha-Numeric, Auto-Skip, Hidden, Protected, Blink, Hidden, Intersified, Reverse, Underscore, Extended Color 등이 있다. 이러한 Attribute들의 특징을 웹으로 반영하기 위하여 각 해당 Attribute들을 원래 데이터 화면과 먼저 분리해내야 한다. [그림 2]는 기존의 터미널 화면이며, [표 2]는 Attribute가 제거된 터미 화면, [표 3]은 터미널 화면에서 분리한 Attribute들을 저장한 Table의 구조를 나타낸 것이다. 이 Attribute들은 각 터미널의 특성에 따라 지정된다. 즉, VT100, VT220, 3270, 5250 터미널 등에서 사용되는 각각의 Attribute들을 저장하는 것이다



[그림 2] VT100 터미널 화면

통합 생산 시스템		1997/04/30
01	고객관리 업무	
02	제품관리 업무	
03	장비관리 업무	
04	자금관리 업무	
05	사원관리 업무	
해당업무를 선택하십시오		

[표 2] Attribute가 제거된 Dummy 화면

내용	행	열	길이	속성
통합	1	28	4	Reverse
생산	1	33	4	Reverse
시스템	1	38	6	Reverse
1999/07/40	2	63	10	Under score
01	4	25	2	Bold
고객관리	4	29	8	Bold
업무	4	38	4	Bold

[표 3] Attribute Table

4.3 Refiner

Refiner는 호스트의 터미널 화면과 HTML문서를 상호 연결하여 변환하는 작업을 한다. 첫 번째로 CT-Emulator에서 넘겨준 Attribute 정보를 HTML문서의 해당 태그로 변환하여 터미널 화면에서의 각 문자들의 위치 정보를 저장하고 있는 터미 화면과 통합하여 웹브라우저에 전송할

HTML문서를 만든다. 두 번째는 GS-Browser에서 넘겨준 Input Filed 정보와 해당 태그들을 Attribute 정보와 더미 화면으로 변경시킨다. 이 변경되어진 화면은 GS-Browser에게 넘겨지지만 Attribute 정보와 더미 화면은 Session ID 및 Sequence ID와 함께 Refiner의 Session Pool에 저장된다. GS-Browser를 통해 사용자에게 전달될 후 다시 사용자 처리를 거쳐 GS-Browser에 되돌아 오는 정보는 화면 전체에 관한 데이터가 전달되어져 오는 것이 아니라 CGI 형태로 돌아온다. 다시 말해, Hidden Field로 사용자에게 넘겨진 Sequence ID, SessionID와 사용자의 데이터 입력이 가능한 Input Field로 정의된 것과 해당 값, 키보드 핸들러의 이름과 해당 값만이 넘어오기 때문이다. 그러므로 각 넘겨진 CGI 형태의 정보와 Session Pool에 저장된 호스트의 터미널 화면과의 매칭 작업을 통해 변환되는 것이다.

4.4 GS-Browser(Gather & Scatter Browser)

GS-Browser는 사용자 웹브라우저와의 사용자 인터페이스를 담당한다. GS-Browser는 크게 키보드 이벤트 처리를 위한 Key-Handler와 CGI-Decoder로 구성된다. Key-Handler는 Key 처리 중심의 터미널 환경의 PF1~PF24, PA1, PA2 등의 기능키 기능을 웹 브라우저 상에서 구현하게 하는 것이다. [그림3]에서 화면 하단에 표현된 버튼이 바로 Keyboard Event 처리기인 Key-Handler를 통하여 구현된 화면으로서 "

Key-Handler는 넘어온 데이터 중 키보드 이벤트 값을 추출, 분석하여 호스트의 해당하는 신호로 변경하며, CGI-Decoder는 특수문자와 각 입력 필드의 값을 디코딩하여 Text 형태로 변환한 후 정제 작업을 거쳐 [표2]와 비슷한 형태의 테이블 정보로 저장하여 refiner에게 넘겨준다. [표3]에서 <input Type=Text Name="input.1.15.49" Value="" Size="3" Maxlength="3">은 터미널 화면에서 사용자가 데이터를 입력하는 곳으로서 Name의 input.1.15.49의 15는 15행 49는 49열을 나타낸다. Size의 3은 input field의 크기를 나타내며, maxlength의 3은 입력 가능한 데이터의 최대 크기를 말한다. 일반적으로 size와 maxlength의 값은 동일하다.



[그림 3] WEB-TO-HOST기 생성한 화면

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=euc-kr">
</HEAD>
<BODY style="font-family 굴림체; font-size 12" text="#0000ff" bgcolor="#B0B0B0">
<INPUT TYPE="hidden" NAME="SessonId" VALUE="923561287">
<INPUT TYPE="hidden" NAME="SequenceID" VALUE="13602">
<FORM METHOD="POST" ACTION="http://localhost/Trans"><PRE>
<Font Color=White>총합</Font> <Font Color=White>선산</font>
<Font Color=White>시스템</Font>
      <u>1997/04/30</u>
      <B> 01 고객관리 업무</b>
      <B> 02 제품관리 업무
      <B> 03 장비관리 업무</b>
      04 자금관리 업무
      <B> 05 사원관리 업무</B>
<Font Color=White>해당업무분 선택하세요.</Font> <Input Type=Text
Name="input.1.13" Value="" Size="3" Maxlength="1">
</PRE>
<hr width=1>
<INPUT TYPE="Submit" NAME="Button_Action" VALUE="Enter">
<INPUT TYPE="hidden" NAME="GrawScreen" VALUE="Yes">
</FORM>
</BODY>
</HTML>
```

[표 4] WEB-TO-HOST에서 생성된 화면의 HTML

4.5 구현 넷 결론

[그림 3]은 이 WEB-TO-HOST를 통하여 만들어진 [그림 2] 호스트 터미널 화면의 변환된 화면이다. 이 시스템은 실시간으로 변환되는 시스템이기 때문에 각 호스트의 터미널처럼 사용이 가능하다. 즉, 각 호스트에 맞는 환경에 맞는 네트워크 환경을 구축할 필요가 없다. 현재 인터넷은 진 세기에 사용되지 않는 곳이 없고, 인터넷을 말하지 않고는 기업활동 및 개인의 행동을 정의할 수 없을 정도이다.

IBM에서 사용되는 SNA(Sysetem Network Architecture)나, VTxx 환경에서 적용되던 Network 환경은 유지하기 위한 전용 네트워크의 구축은 기업의 활동과 비용에 많은 부담으로 작용해 왔다. 그러나 본 연구의 이 WEB-TO-HOST 시스템을 사용한 3-Tier구조의 미들웨어 시스템을 구축한다면 기업활동을 위한 네트워크 구축을 TCP/IP기반의 인터넷으로 대체할 수 있다. 즉, 해외지사나 지방을 연결하는 비싼 전용선 망을 값싼 인터넷 선으로 대체할 수 있어 경비를 대폭 절감할 수 있다. 또한, WEB-TO-HOST는 기존의 메인 프레임 화면을 웹으로 실시간으로 전환해 주며, 특정한 사용자를 위한 화면을 구성하거나 특정한 몇몇 화면을 조합하여 사용자에게 전혀 새로운 정보를 제공해 줄 수 있는 기능을 제공하고 있다. 이러한 기능은 메인 프레임용 데이터를 전사적 지원관리나 멀티미디어 데이터와 조합하여 클라이언트 서버 형태의 GUI를 구현하도록 해 줄 수 있다.

참고문헌

- [1] A. Luotonen and T. Berners-Lee, CERN httpd Reference Manual - A Guide to a World-Wide Web Hypertext Daemon, CERN, May, 1994.
- [2] D.M Krstol, Proposed HTTP State Management Mechanism, Internet Draft, Jun, 1996.
- [3] D. Robinson, The WWW Common Gateway Interface Version1.1, Internet Draft, Jan, 1996.
- [4] Douglas Comer, The Internet Book, Prentice-Hall, 1995.
- [5] IS GRAHAM, The HTTP Protocol and Common Gateway Interface, HTML SOURCE BOOK, pp. 181-230, 1995.
- [6] P.C Kim, "A Taxonomy of the Architecture of Database Gateways for the Web," Proc of the 13th ICAST97, Schaumburg, IL, pp 226-232, April, 1997.
- [7] T. Berners-Lee, Hypertext Transfer Protocol - HTTP/1.0, Internet draft, Dec. 20, 1994.
- [8] T. Berners-Lee and D. Connolly, Hypertext Markup Language Specification - 2.0, Internet draft, Feb. 8, 1995.
- [9] <http://www.byte.com/art/9707/sec4/art1.html>
- [10] <http://shinan.hongik.ac.kr/~joensae/interanet/chap01.html-chap14.html>
- [11] <http://etlars.etcd.re.kr/EtlarsHome/lecture/intranet.html>
- [12] <http://www.infoage.co.kr>