무선인터넷 단말기 에뮬레이터의 사용자 인터페이스에 대한 설계 및 구현

유승범°, 김성찬, 장지산, 신동규, 신동일
세종대학교 컴퓨터공학과
{bummy, kimschan, jsjang, shindk, dshin@gce.sejong.ac.kr}

An Design and Implement of Interface Of Wireless Internet Terminal Emulator

Seung-Bum Yoo°, Seong-Chan Kim, Ji-San Chang, Dong-Kyoo Shin, Dong-il Shin
Department of Computer Engineering, Sejong University

요 약

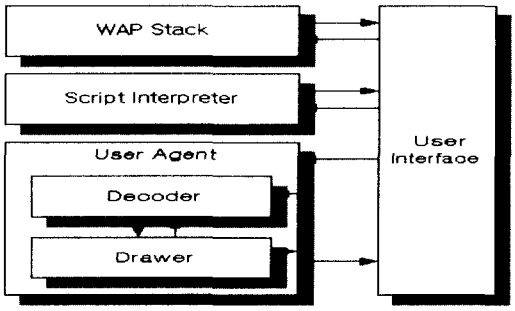
인터넷 기술의 발달과 사용자들은 언제 어디서나 원하는 정보를 얻기를 원하게 되었다. 이러한 요구를 배경으로 무선 인터넷 시장은 급속한 발전을 이루고 있다. 그리고 사용자들의 요구도 점차 다양화되고 있는 실정이다. 그로 인해 대표적인 무선 인터넷 기반 언어인 WML(Wireless Markup Language)과 WMLScript를 개발자가 좀 더 실제와 같은 상황에서 편리하게 개발할 수 있는 환경이 필요하게 되었다. 본 논문에서는 CP(Content Provider)들이 현실감 있게 무선 컨텐츠를 개발, 브라우징 할 수 있도록 하는 WAP Emulator에서 브라우저의 구조와 WMLScript Interpreter를 포함하고 있는 인터페이스 설계와 구현에 대하여 기술한다.

1. 서 론

최근 몇 년간 엄청난 발전을 이룬 인터넷 기술은 사용자로 하여금 언제 어디서나 네트워크에 연결된 컴퓨터만 있으면 자신이 원하는 정보를 네트워크를 통해 손쉽게 취득하는 것을 가능하게 하였다. 따라서 시시각각 변하는 형태의 정보를 얻기 원하거나, 데스크탑 컴퓨터가 설치되지 않은 장소에서도 인터넷에 접근하고자 할 때, 그 효용성이 극대화 될 수 있는 소형 휴대 장비를 이용하여 네트워크를 통한 정보를 수집하고자 하는 요구가 증가하는 추세이다. 이에 WAP(Wireless Application Protocol)[1] 기술을 이용한 휴대용 전화기나 PDA(Personal Digital Assistant)와 같은 무선 터미널을 이용한 웹 정보의 접근이 대중화되고 있다. 따라서 상용화 서비스를 위한 WML(Wireless Markup Language)[2] 응용 프로그램의 개발이 국내외적으로 활발히 이루어지고 있는 실정이다. 이러한 상황에서 WAP 사이트를 브라우징하고, 무선 컨텐츠를 제작, 컨텐츠 문서 내부의 오류를 검출, 수정하기 위한 PC용 WAP브라우저의 개발이 절실하게 요구되고 있다. 본 논문의 구성은 다음과 같다. 2장에서는 WAP 브라우저의 구조에 대해서 기술하고, 3장에서는 WMLScript 설계 및 구현, 4장에서는 WAP Emulator의 전체구조와 User Interface의 설계 및 구현에 대해 기술하고, 마지막 5장 및 6장에서는 결론 및 향후 연구방향에 대해 기술하겠다.

2. WAP 브라우저의 구성

WAP 브라우저의 전체적인 구조는 [그림 1]과 같이 WAP Stack, Script Interpreter, User Agent(UA)와 각 모듈간을 연결해주고 사용자와 브라우저 사이의 인터페이스 기능을 가진 User Interface(UI)로 이루어져 있다.



[그림 1] WAP 브라우저 모듈 구성도

2.1 WAP Stack

W3C의 WAP Specification 1.2에 따라 구현되었으며 사용자의 요청이나 추가 요청이 필요한 문서를 파싱했을 경우 WAP 게이트웨이로 요청을 보내게 되고, 그에 따른 응답을 받게 된다. 또, 데이터의 송수신에 대한 에러, 예외 처리를 담당한다.

2.2 Script Interpreter

사용자의 각 동작 이벤트에 따른 WMLScript[3]로 처리될 작업이 있을 경우 Script Interpreter에서는 해당 작업

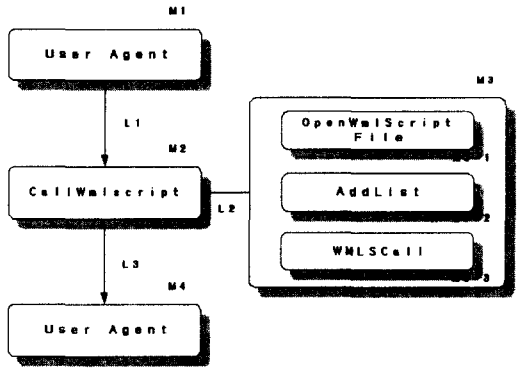
을 수행하여 User Agent로 값을 돌린다. WMLScript 내부의 에러가 있을 경우 에러를 검출해 낸다.

2.3 User Agent

WAP 브라우저에서 가장 핵심적인 부분으로 WML문서의 파싱 및 브라우저 화면상의 디스플레이를 담당한다.

2.4 User Interface

전체적으로 모듈간을 연결해주고, 모듈간의 메시지 전송을 대신하는 역할을 한다. 또한 메모리 관리와 캐쉬,



[그림 2] WMLScrpit Interpreter 전체 구조도

히스토리 기능을 담당한다.

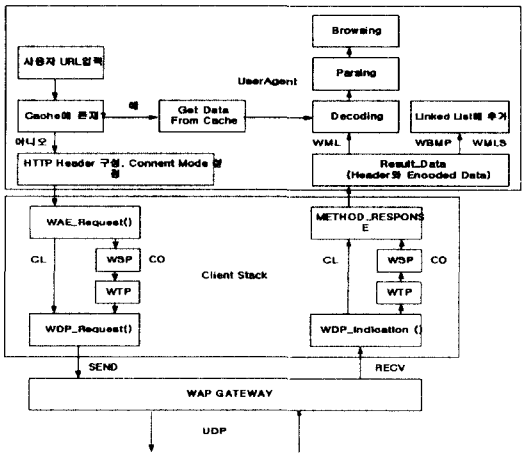
3. WMLScript Interpreter의 설계 및 구현

WMLSript interpreter에서의 내부 데이터 및 Function의 실행은 [그림 2]의 구조를 전체적인 설계구조로 하였다. 전체적인 구조에서 연산 수행은 다음의 순서를 따른다.

- UserAgent M1이 WMLScript interpreter에게 연산에 필요한 파라미터를 전달한다.
- 파라미터들은 WMLScript interpreter의 함수 호출과 관련되는 전반부 처리기 M2에 전달된다.
- M2로 전달되는 파라미터들은 바이트 코드에 대한 연산을 수행하는 M3로 전달된다.
- M3-1은 전역 메모리 포인터에 M1이 전달하는 파라미터 중 WMLScript 문서 전체와 크기를 저장한다. 여기서 저장을 하는 것은 M2에서 데이터를 분석하여 OperandStack에 저장하기 위함이다.
- M3-2는 Variable을 List 스택에 저장하는 역할을 수행한다. 이 데이터들은 이 후에 연산에서 리스트 목록을 포인터로 전달받음으로써 동작을 처리할 수 있게 한다.
- M3-3은 연산이 수행되는 단계이다.
- M4에서는 다시 UserAgent에게 연산 후의 결과 데이터를 전달한다. 데이터의 전달은 저장되어 있는 포인터의 주소 값을 전달하는 것과 WML문서 정보를 갖고 있는 UserAgent 구조체에 데이터를 갱신 혹은 추가한다.

4.1 Wap Emulator의 전체적인 구조

설계, 구현한 Wap Emulator의 전체 구조 및 동작방법에 대해 살펴보자. 사용자가 테스트 하고자 하는 URL을 입력하면 그 URL로 Requst를 보내기 위해 프로그램에 포함된 Client Stack을 호출하여 Wap Gateway[4]로 UDP로 소켓을 생성하여 Request를 보내게 된다. Requst를 함에 있어서 Connection Oriente(CO)방식과 ConnectionLess(CL)방식 두 가지가 있는데 해당 방식에 따라 Client Stack의 단계를 거쳐 Wap Gateway로 Request를 보내게 된다. WAP Gateway는 WAP 기반 무선 인터넷을 구성하는 핵심 요소로써 프로토콜변환 및 WML Content의 암호화 및 복호화 등의 작업을 담당한다. 이 WAP Gateway에서 해당 URL의 Content를 가져와 다시 Wap Browser로 Response를 주게 된다. Wap Gateway에서 Response가 들어오면 먼저 Client Stack이 받아 UserAgent로 전달한다. UserAgent는 Encoding된 Data를 Decoding작업을 수행하고 이를 화면상에 Display를 하도록 Data를 가공한다. 이때 가장 중요한 작업이 화면상에 Display를 하기 위해 데이터를 가공하



[그림 3] Wap Browser의 전체 구조도

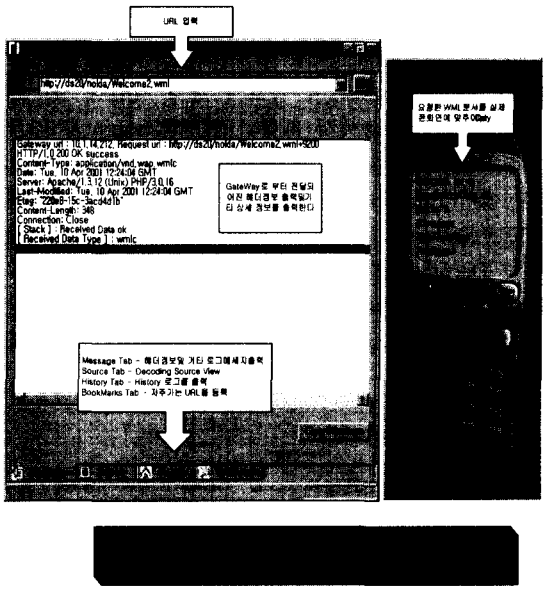
는 작업인데 이 작업을 어떻게 하는가에 따라 화면에 Display하기가 수월해 진다.

이 모든 작업이 One Cycle (URL입력->ClientStack->WapGateway->ClientStack->UserAgent->UserInterfac

e) 로 수행되지만 필요한 동작을 수행 하도록 필요한 함수를 호출 하도록 하고 관리를 할 수 있는 커다란 프로그램이 존재하는데 이것이 User Interface부분이다.

4.2 User Interface의 설계 및 구현

WAP Emulator의 실행화면은 [그림 4]와 같다. 사용자로부터 URL을 입력받을 수 있도록 URL을 입력받는 부분과 메시지를 출력하는 부분을 가지고 있고 또한 Wap Gateway 및 Cache의 사용여부를 설정 할 수 있도록 하는 메인 다이얼로그와 그리고, WML문서에 대한 화면 출력을 볼 수 있는 실제폰과 같은 폰 이미지를 가지고 있는 폰 다이얼로그로 크게 나누어져있다. 그리고 내부적으로 Cache와 History를 구현 할 수 있도록 하는 부분을 포함하고 있다.



[그림 4] 실제 WML 컨텐츠 브라우징 화면

4.3 구현환경

본 WAP 브라우저는 Microsoft Visual C++ 6.0에서 구현되었다. Windows 2000상에서 테스트되었으며, WAP 브라우저와 통신한 WAP 게이트웨이는 Unix 상에서 실행되었다.

5. 결론

본 논문에서는 무선 인터넷 컨텐츠를 실제 무선 기기와 비슷한 환경을 제공함으로써, CP들이 좀 더 쉽게 WML 및 WMLScript 문서를 제작하고 오류를 수정할 수 있는 WAP 브라우저의 파싱과 디스플레이 기능을 가진 User Agent의 설계와 구현에 대해 기술하였다.

6. 참고 문헌

[1] WAP, "Wireless Application Protocol Architecture Specification, WAP Forum,
http://www.wapforum.org/what/technical.htm

[2] WML, "Wireless Markup Language Specification" ,WAP Forum,
http://www.wapforum.org/what/technical.htm

[3] WMLScript, "WMLScript Language Specification, WAP Forum,
http://www.wapforum.org/what/technical

[4] WAP Gateway, http://www.mosca.co.kr