A Design and Implementation of HTML Filter for Wireless Internet Service

Sam-Jin Jeong

Abstract In this paper, we propose a design and implementation of HTML Filter to provide wireless internet service efficiently. The HTML Filter translates the Web documents written by HTML into the WML documents for portable wireless equipments automatically that are suitable for the WAP environment. The HTML Filter consists of "the HTML lexical analyzer" which gathers HTML source into token units, "the HTML syntax analyzer" which takes the lexical units from the lexical analyzer and analyses grammar for the source, "the HTML abstract parser tree" which can represent meaning of HTML sources more effectively called an intermediate code, and "the target code generator" which can convert the abstract parser tree into WML documents. The proposed translating method using ordinary compiler technology provides more easy and effective than the existing converters, and will be applied to other wireless language translating methods on wireless internet.

Key Words : 무선 인터넷, HTML Filter, WML, WAP, 컴파일러

1. 서론

최근 무선 단말기 사용자의 급속한 증가와 네트워크 인프라의 구축 및 발전으로 인하여 기존 유선망에서의 함께 무선망에서 데이터리를 서비스 받고자 하는 요구가 급증하고 있다.

그러나 일반 데스크탑 컴퓨터나 워크스테이션을 기준으로 제작된 기존의 웹 콘텐츠를 무선 단말기 사용자에 게 제공하는 것은 무선 단말기에 적합하지 않다.

또한 무선 인터넷 콘텐츠를 새롭게 재작하는 것은 추 가로 업무시간 시간과 비용이 들게 된다.

따라서 기존의 웹 콘텐츠를 무선 인터넷 환경에 적합

하도록 변환해서 무선 단말기 사용자에게 제공하는 변환 기가 필요하다.

즉 변환기는 기존 유선망에서 HTML로 되어 있는 웹 콘텐츠를 WML과 같이 무선 단말기에서 지원하는 마크 업 언어로 변환하는 것이다.


위와 같은 WAP라고 하는 무선 인터넷 환경에서 기존의 HTML로 되어 있는 웹 콘텐츠를 WML과 같은 무
원 마크업 언어로 변환해서 무선 단말기 사용자에게 제공하는 것이 HTML Filter이다. 본 연구는 기존의 웹파일러 개발 기술을 이용하여 HTML Filter를 개발하였고, 다음과 같이 네 단계로 개발하였다.

첫째는 HTML 프로그래머를 토대로 구성하는 "HTML 이외 분석기" 개발이고, 두 번째는 이외 분석 단계에서 만들어진 토큰들을 받아들여서 프로그래머의 문법적 구조를 분석하는 "HTML 구문 분석기" 개발이며, 세 번째는 HTML 프로그래머의 의도를 보다 효율적으로 표현할 수 있는 중간언어 "HTML 수상구문들"의 설계 및 개발이며, 네 번째는 중간언어인 수상구문들로 WML 프로그램으로 변환하는 "목적 코드 WML 생성기" 개발이다.

2. 관련 연구

2.1 WAP의 개요

많은 대역폭, 큰 지연시간, 높은 손실율과 같은 특성을 가지는 무선 환경에서는 기존의 HTTP와 HTML으로 서비스하기에는 부적합하다. WAP은 이러한 무선 환경을 고려한 프로토콜로서 모든 콘텐츠를 바이너리형태로 압축하여 전송하며, 무선 환경에 부적합한 HTML과 JavaScript대신 WML과 WMLScript를 사용한다. 따라서 WAP을 사용하기 위해서는 WAP 프락시가 필요하게 된다. WAP 프락시는 WWW 기반의 프로토콜 스택을 WAP 기반의 프로토콜 스택으로 변환하는 프로토콜 게이트웨이 역할과 콘텐츠를 인코딩하고 디코딩하는 역할을 담당한다.

[그림 1]은 WAP에서 제안한 무선 인터넷 서비스 모델을 보여준다.

[그림 2]는 WAP을 이용한 빌의 예를 보여준다. WAP 서비르는 클라이언트로부터의 WAP 요청으로 변환하거나 웹서버로부터의 응답을 클라이언트가 이해할 수 있는 바이어리 형태로 인코딩한다. 클라이언트가 요청한 문서는 WML 문서와 HTML 문서 모두 가능하다. HTML 문서일 경우, HTML Filter를 거쳐 WML 문서로 변환된다.

그래서 Web 서버에서 HTML로 된 콘텐츠들은 WML로 자동 변환하여, 이동 전화기 등 소형 무선 단말기 상에서 인터넷을 이용할 수 있도록 해주는 HTML Filter의 개발이 필요하다.

2.2 WML의 개요

이동전화로 전달된 콘텐츠를 이동전화의 디스플레이 상에서 표시하기 위해서는 전용 브라우저가 필요하다. 콘텐츠 내비게이션을 위해 이동전화 단말기 상에 몇 가지 전용 바른을 마련하고 간단하게 페이지 간 이동이 가능하게 한다. 또 문자 입력 시에는 전화기 버튼을 이용하여 손자나 영문, 한글 등을 입력하는 것이 가능하다.

WAP에서 중요한 또 다른 한 가지는 바로 인터넷상의 HTML 언어와는 다른 HTML(Handhel Device Markup Language)을 사용한다는 것이다. 이를 WAP에서는 WML로 부른다.

WML은 다음과 같은 기능을 지원할 수 있는 요소들을 정의한다[3].

첫째, 텍스트와 이미지를 화면에 다양하게 보여줄 수 있는 방법을 제공한다. 둘째, 화면의 크기를 고려하여 정보를 여러 개의 단위(card)로 나누고, 여러 개의 카드로 구성된 전송단위를 데이라 한다[4]. 셋째, 뒷에 있는 카드를 링크시켜 카드와 뒷 사이의 이동을 명시적으로 관리할 수 있다. 넷째, 하나의 변수를 여러 개의 카드에서 계속 사용할 수 있다.

2.3 WML 변환기

현재 개발된 WML 변환기는 Amaro 연구소의 TransWAP, ArgoGroup의 WAPTool이 있고, IBM의 WebSphere Transcoding과 Microsoft의 HTML 편집기인 FrontPage 에도 WML 변환기 기능이 추가 되었다. 그 외에 엠넷소
프트의 M-converter 2.5, 엠프리지의 M-이네이블러 등이 있다[5].

국내에서 개발된 HTML Filter는 로그 처리기, HTML 분석기, 태그 조정기와 WML 재구성기를 이용한 개발 방법[5], SGML Parser Toolkit을 이용한 개발방법[6], 그리고 Parsing Engine을 통해 HTML Tag를 분석하고, 미리 정의된 변환 규칙 Rule Set Database를 참조하여 Markup Language Translator를 통하여 WML 문서를 변환하는 방법[16]등이 있다.

3. HTML Filter의 설계 및 구현

3.1 HTML 어휘 분석기 개발

어휘는 정규문법(regular grammar)에 의해서 기술될 수 있고, 정규문법에 따라 생성되는 언어는 유한 오토파타(finite automaton, FA)에 의해서 인식된다. 프로그램에서 사용되는 어휘를 인식하는 FA를 어휘 분석기(lexical analyzer) 혹은 스캐너(scanner)라고 부르며, 프로그램을 문법적으로 가장 작은 의미를 갖는 단위인 토큰(token)으로 분쇄하는 작업을 수행한다[8]. 어휘 분석기는 어휘 분석기 생성기를 사용함으로써 쉽게 만드는 수가 존재한다. 현재 널리 사용되고 있는 어휘 분석기 생성기는 lex, scanner 등이 있는데, 본 연구에서는 Linux에서 기본으로 제공하고 있으며, 가장 널리 사용되고 있는 lex를 사용하여 어휘 분석기를 생성하였다[9].

HTML 구문에서 사용되는 토큰(token) 일부는 [그림 3]과 같이 정의하였다.

```
<html>  { yyval.integer = Thtml; return yyval.integer; }
<head>  { yyval.integer = Thead; return yyval.integer; }
<body>  { yyval.integer = Tboby; return yyval.integer; }
<title> { yyval.integer = Ttitle; return yyval.integer; }
<a>    { yyval.integer = Tao; return yyval.integer; }

left    { yyval.integer = Tleft; return yyval.integer; }
align   { yyval.integer = Taalign; return yyval.integer; }
href    { yyval.integer = Thref; return yyval.integer; }
(id)    { yyval.string = yrtext; return ID; }

[그림 3] HTML 구문에서 사용되는 토큰(token)들의 정의
```

HTML 구문에서 사용되는 시작 태그와 종료 태그 사이에 사용되는 내용들은 다음과 같이 정의하였다.

char0   "*[a-zA-Z]+*[a-zA-Z]+*
char1   "<*[a-zA-Z]+*
char2   
;

id   {char0}|{char1}|{char2}

위에서 char2는 문자와 숫자가 복합적인 일반적인 문장을 뜻하고, char0은 "test.html"와 같이 <A> 태그의 HREF 속성의 값을 뜻하며, char1은 <B>과 같은 태그의 경우 필요한 텍스트를 표시하는 문자열을 받아들이기 위한 것이다.

3.2 HTML 구문 분석기 개발

프로그램밍 언어의 문법 구조는 일반적으로 Context Free Grammar(CFG)이다. CFG에 따라서 작성된 프로그램은 Push Down Automaton(PDA)에 의해서 인식된다. 프로그램을 인식하는 PDA를 구문 분석기(syntax analyzer) 또는 파서(parser)라고 부르며, 프로그램을 문법적으로 분석한다[10]. 구문 분석기의 어휘 분석기와 마찬가지로, 구문 분석기를 자동으로 만들어 주는 구문 분석기 생성기(parser generator)를 이용하면, 쉽게 구문 분석기를 만들 수 있다. 구문 분석기 생성기는 CFG 문법을 입력으로 받아서 그 문법에 따라 생성된 언어를 인식하는 구문 분석기를 출력으로 만들어 낼 수 있다[11].

현재 널리 사용되고 있는 구문 분석기 생성기는 yacc, bison 등이 있는데, 본 연구에서는 UNIX에서 기본으로 제공하고 있으며, 가장 널리 사용되고 있는 yacc를 사용하여 구문 분석기를 생성하였다.

일반적으로 프로그래밍 언어는 BNF로 기술되어 있으나, yacc이 입력으로 받아들이는 문법도 BNF가 가까운 표현이다[12]. 따라서 yacc을 이용하기 위해서는 HTML 문서의 문법을 먼저 BNF형태로 바꾸어야 한다.

BNF로 표현된 HTML 문서의 문법 일부를 보면 [그림 4]와 같다.

```
Program ::= Block
Block ::= 'html' <head> 'HeadList'  
          '/<head>' 'body' 'BodyList' '/<body>
$html' 

HeadList ::= Head | HeadList Head
Head ::= '#Itsmt
TitleStmt ::= '/'<title>' NameList

[그림 4] HTML 문서의 구문 요소들
```

1661
3.3 HTML 추상 구문 트리의 설계 및 개발

추상 구문 트리는 원시 프로그램의 다른 표현으로서, 프로그램의 구문적 구조를 그대로 유지하고 있다. 원시 프로그램의 트리 구조 표현은 프로그램의 의미를 보다 효율적으로 표현할 수 있는 방법이 되며, 트리의 특성상 손 쉽게 재구성이 가능하므로 최적화 컴퓨터 비어 중간 언어로 가장 적합한 표현이다[13].

추상 구문 트리의 설계 시 다음과 같은 사항을 고려해야 한다[14].

1) 노드의 종류의 가능성을 한 정적 한다. 노드의 종류가 많으면, 트리 순회시 각 노드 종류별 구분이 필요하며, 그에 따라 다른 처리를 필요로 하게 되며, 많은 시간을 소모하게 되고 프로그램이 복잡하게 된다.
   2) HTML의 프로그램이 가진 문법적 구조를 그대로 유지하도록 한다. 이는 트리에서 추상 구문 트리로부터 원시 프로그램을 생성할 때, 원래의 프로그램을 정확히 생성하기 위해서이다.

   세째, 추상 구문 트리의 구조는 일반적인 프로그래밍 언어에서 많이 사용되지만 HTML 언어를 대상으로 용이 적용될 수 있도록 설계했다.

   넷째, 각 노드는 추상 구문 트리의 구성, 데이터 종류와 속성, 프로그램의 변환과 사용자의 편리한 환경을 위해 필요한 여러 가지 정보를 포함할 수 있도록 하였다.

추상 구문 트리의 크기 두 부분으로 구성되어 있다. 프로그램의 문법적 구조를 트리형태로 표현하는 프로그램 트리, 프로그램에서 프로그램의 구조에 자연스럽게 사용하고 있는 변수 이름과 같은 심볼들에 관한 정보를 갖는 심볼 테이블(symbol table)이 있다. 이 두 부분을 구성하고 있는 노드들이 서로간의 관계에 따라 포인터로 연결되어 있는 것이 추상 구문 트리이다.

프로그램 트리는 테그의 속성을 표현하는 테그 노드와, 문장을 표현하는 문장 노드를 중심으로 구성되어 있다. 심볼 테이블은 프로그램에서 사용된 심볼마다 하나의 심볼 노드가 할당되어, 모든 심볼들에 관한 정보를 가지 고 있다.

추상 구문 트리는 문장 노드, 테그 노드와 같이 두 가지 종류의 노드를 사용하고 있다.

3.3.1 문장 노드

문장에는 하나 이상의 문자열들로 이루어져 있다. 문자열은 문자, 숫자 및 기타 특수 문자들도 포함할 수도 있어야 한다. 문장 노드의 구조는 [그림 5]와 같다.

```
struct ExprNode {
    char *Text;
    struct ExprNode *next;
}
```

[그림 5] 문장 노드의 구조

Text는 하나의 문자열을 저장하는 주소 값을 나타내며. 포인터 next는 문장 노드간의 트리 구조가 아닌 링크드 리스트 구조를 정의한다. 예를 들어서,"a bc def"와 같이 여러 개의 문자열들을 연속적으로 표현하기 위함이다.

3.3.2 테그 노드

테그 노드는 모든 종류의 테그들에 대하여, 테그의 속성을 표현할 수 있어야 한다. 테그노드의 구조는 [그림 6]과 같다.

```
struct StmtNode {
    int NodeVar;
    struct ExprNode *expr;
    struct ExprNode *stmt;
    struct StmtNode *next;
}
```

[그림 6] 테그 노드의 구조

NodeVar은 테그의 종류를 구별하기 위한 것으로, 테그의 종류마다 고유한 번호를 갖는다. 테그는 하나 이상의 문장으로 이루어져 있으며, expr은 그 테그에 속한 문장 노드를 가리키는 포인터이다. 포인터 stmt는 <a> 테그와 같이 속성과 문장이 함께 있는 테그들을 위해서 정의
3.3.3 추상 구문 트리의 검증

원시 프로그램으로부터 만들어진 추상 구문 트리가 원래의 구문적 구조를 그대로 나타내고 있는지를 검증하기 위해서는, 변환된 추상 구문 트리로부터 원래의 원시 프로그램을 재생성할 수 있어야 한다. [그림 7]은 HTML로 기술한 원시 프로그램을 보여준다.

```html
<!-- WML Test -->
<html>
  <head>
    <title>
      Title
    </title>
  </head>
  <body>
    <p>WML TEST</p>
    <p align="left">test left</p>
    <p align="center">test center</p>
    <p align="right">test right</p>
  </body>
</html>
```

[그림 7] 원시 프로그램

3.4 목적 코드 WML 생성

HTML Filter 개발의 마지막 단계가 목적 코드 생성 단계이며, HTML 프로그램에서 중간 언어를 생성한 후 중간 언어를 목적 코드인 WML 프로그램으로 변환하는 작업을 말한다.

HTML 문서를 WML 문서로 변환하는 과정에 대해서는, HTML 문서는 기본적으로 90 개 이상의 태그로 구성되며, 각각의 태그의 사용에도 예의 상황이 많아 완벽하게 XML에서 제공하는 문서의 특성을 WML로 재생할 수 없다. 그러므로 WML로 표현 가능한 HTML의 태그를 찾아내기 위해서 HTML 문서의 재구성이 필요하며 이렇게 변형 가능한 HTML 문서를 WML로 변환한다. HTML에서 제공되는 태그를 아래의 [표 1]과 같이 논리적 요소로 분류한다.

### [표 1] 논리적 분류

<table>
<thead>
<tr>
<th>논리적 요소</th>
<th>예제</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;HTML&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;BR&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;LI&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;UL&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;OL&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;NL&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;H1&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;H2&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;H3&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;H5&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;I&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;U&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;P&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;A HREF=&quot;값&quot;&gt; 내용 &lt;/A&gt;</td>
<td>좌표</td>
</tr>
<tr>
<td>&lt;INPUT TYPE=&quot;타입&quot; NAME=&quot;이름&quot; VALUE=&quot;값&quot; SIZE=&quot;크기&quot;&gt;</td>
<td>좌표</td>
</tr>
</tbody>
</table>

HTML 문서를 WML 문서로 변환하기 위해서 변환 가능한 코드의 분류에 관하여 아래의 [표 2]와 같이 정리할 수 있다.

### [표 2] 변환 가능한 코드의 분류

<table>
<thead>
<tr>
<th>논리적 요소</th>
<th>해당 Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>&lt;head&gt;, &lt;access&gt;, &lt;meta&gt;, &lt;setvar&gt;, &lt;template&gt;</td>
</tr>
<tr>
<td>Text</td>
<td>&lt;br&gt;, &lt;big&gt;, &lt;em&gt;, &lt;i&gt;, &lt;small&gt;, &lt;strong&gt;, &lt;u&gt;, &lt;b&gt;</td>
</tr>
<tr>
<td>Select</td>
<td>&lt;select&gt;, &lt;option&gt;, &lt;field set&gt;</td>
</tr>
<tr>
<td>Link</td>
<td>&lt;a&gt;, &lt;anchor&gt;</td>
</tr>
<tr>
<td>Card Action</td>
<td>&lt;do&gt;, &lt;go&gt;, &lt;prev&gt;, &lt;noop&gt;, ...</td>
</tr>
<tr>
<td>Input</td>
<td>&lt;input&gt;</td>
</tr>
<tr>
<td>Comment</td>
<td>&lt;!--, --&gt;</td>
</tr>
<tr>
<td>Table</td>
<td>&lt;table&gt;, &lt;td&gt;, &lt;tr&gt;</td>
</tr>
<tr>
<td>Paragraph</td>
<td>&lt;p&gt;</td>
</tr>
</tbody>
</table>

4. 결론 및 향후 과제

본 연구에서는 무선 환경에서는 기존의 HTTP와 HTML으로 서비스하기에는 부적합함으로 Web 서비스에서 HTML로 된 콘텐츠들을 WML로 자동 변환하여, 이동 전화기 등 소형 무선 단말기 상에서 인터넷을 이용할 수도 있도록 해주는 HTML Filter를 개발하였다.

Lex와 Yacc와 같은 기존의 컴파일러 개발 Tool들을 이용하여 개발된 변환 방식은 기존의 다른 HTML Filter 개발 방식보다 개발하기 쉽다.

또한 “HTML Filter” 개발로 개발된 어휘 분석기, 구문 분석기, 추상구문트리, 목록 코드 생성기 등에 관한 개발 기술들을 이용하여 타 언어의 자동변환기 및 타 언어 컴파일러(Compiler) 개발에 활용될 것이다.

향후 계획으로는 WML 텍스트 문서를 바이너리 형태로 인코딩함으로써, 무선 셀로에서 전송되는 데이터의 양을 최소화할 뿐만 아니라, 이동 단말기에서의 문서 처리 부담을 경감하여 보다 빠르게 실행시킬 수 있는 무선 인터넷 서비스를 위한 WAP 게이트웨이용 WML 컴파일러의 개발이며, HTML 콘텐츠의 스마트폰의 처리, 이미지 변환, 웹에서의 CGI 기능처리 등의 연구와 JPEG, GIF, TIFF, BMP 형식의 이미지를 무선 응용 환경에 적합한 WBMP(Wireless Bitmap) 이미지 변환기 개발이 필요하다.

참고문헌


정 삼 진(Sam-Jin Jeong) [정회원]

- 1979년 2월 : 경북대학교 공과대학 공학부 공과학과 (공학사)
- 1987년 2월 : 인디애나대학교 컴퓨터공학과 (컴퓨터학석사)
- 2000년 8월 : 충남대학교 컴퓨터공학과 (이학박사)
- 1997년 3월 ~ 현재 : 백석대학교 정보통신학부 교수

<관심분야>
병렬 컴퓨팅, 프로그래밍 언어