2002년도 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집(2002.11.30)

XML 이메일 시스템의 필터링 에이전트 인터페이스 설계

정 옥 란, 조 동 섭
이화여자대학교 컴퓨터학과
전화 : 02-3277-2309 / 핸드폰 : 011-651-5054

Design of Filtering Agent Interface using XML E-Mail System

Ok-Ran Jeong, Dong-Sub Cho
Dept. of Computer Science, Ewha Womans University
E-mail : orchung@ewha.ac.kr

Abstract

인터넷의 발달로 인하여 웹을 통한 문서 송수신이 많아지면서 종래의 인쇄 매체 상에 기술된 문서들은 점차 전자문서화 되기 시작했다. 이러한 문서들을 서로 다른 시스템 사이에서 상호 교환하기 위해서는 사용자가 원하는 논리적 구조를 태그로 구현할 수 있는 정형화된 문서 형태가 필요하다. 또한 이메일을 통한 개인적 정보를 얻고 또한 메일의 양이 갈수록 늘어나는 상황에서 카테고리별 자동 분류를 할 수 있는 에이전트가 현안이 되고 있다.

본 논문에서는 XML 형식의 메일에 XSL 문서를 임베디드하여 보내는 XML 이메일 시스템을 설계하여, 본 시스템을 이용하여 본문 내용을 카테고리별 자동 분류해주는 필터링 에이전트 인터페이스(Filterng Agent Interface)를 제안하고자 한다. XML 메일 서버를 통하여 수신된 메일은 XML과 XSL 형식에 따라 XML 메일 데이터베이스에 따로 저장되기 때문에 분석이 매우 용이하다는 장점을 이용하였다.

I. 서론

인터넷의 정보 공간은 전 세계 네티즌들에게 지역적으로 분산된 정보의 공유와 정보 교류의 장을 제공하고 있다. ARPanet을 기반으로 발전한 인터넷은 하이퍼링크(hyperlink)에 의해 서로 다른 문서 및 미디어를 연결하는 웹 시스템 형태로 발전되었다. 따라서 네티즌들은 웹에서 단순하고 간편한 HTML(Hyper Text Markup Language) 언어로 작성된 하이퍼텍스트 문서들을 이용하여 정보공간의 여행이 가능하다. HTML은 SGML(Standard Generalized Markup Language)에 기반을 두고 있어서 하이퍼텍스트 문서는 물론 이미지를 삽입할 수 있는 기능을 가짐으로써 사용자에게 시각정보를 쉽게 접근할 수 있는 통로를 제공하였다. 그러나 HTML은 화면상에 보여지는 기능 외에는 별다른 기능을 제공하지 않을 뿐만 아니라 태그와 논리적 구조가 고정되어 있어서 만들고자 하는 문서의 논리적 구조를 정확히 표현할 수 없다는 단점이 있다. 웹 발전의 중요한 이유였던 HTML의 단순함이 사용자의 다양한 요구에 의해 현재의 시점에서는 단점이 되고 있는 것이다[1]. 이러한 HTML 문서의 단점을 보완할 수 있는 방법으로 XML과 XSL을 이용한 이메일 시스템을 기반으로 하여 필터링 에이전트 인터페이스(FAI: Filtering Agent Interface)를 설계하고자 한다. 지금까지는 이메일을 통해 문서를 교환할 때에 주로 HTML 문서 형식만을 사용해왔다. 이미 고정되어 있는 태그로 논리적 구조를 표현하는 HTML 형식이 아니라 사용자가 원하는 논리적 구조를 태그로 표현할 수 있다면 인터넷이 제공하는 풍부한 정보의 이점을 더욱 향상시킬 수 있을 것이다.

XML 메일 서버를 이용한 이메일 시스템은 도착한 메일로부터 XML 문서와 XSL 문서를 각각 추출해 내고 메일 데이터베이스에 따로 저장하기 때문에, FAI의 전처리 작업인 메일 분석에 매우 용이하다.

II. XML과 XSL를 이용한 이메일 시스템

2.1 XML

XML은 1996년 W3C(World Wide Web Consortium)에서 제안한 것으로써 웹상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이다. 구조화된 데이터를 나타내고 교환하는 일반적인 방법을 제공함으로써 HTML의 문제점을 보완한다.

XML에는 여러 가지 장점이 있다. SGML의 측면에서 보면 SGML의 서브셋(subset)이면서 SGML에서 XML로의 변환이 용이하고 XML의 일부 수정으로 SGML의 응용에도 사용할 수 있다. 즉, 웹 문서에 쉽게 응용하기 위해 SGML의 특정 부분을 요약 발췌하여 배우기 쉽게 어플리케이션에서도 구현될 수 있다[1,2].

HTML의 측면에서 보면 기존의 HTML을 확장·보완하였기 때문에 HTML을 그대로 사용할 수 있고 지금보다 더욱 복잡한 문서의 생성이 가능하며 구조적인 정보도 포함할 수 있다. 또한 XML을 네트웍으로 전송할 때에 HTTP를 사용하는데 이 때 데이터가 클라이언트에 전달되면 다시 서버로 가지 않고도 이 데이터를 조작· 편집하여 다양한 형태로의 디스플레이가 가능하다.

XML은 서로 다른 소스로부터 추출된 데이터를 비슷하게 통합시켜 구조화한 데이터를 사용자 인터페이스와 분리시킬 수 있다. 예를 들면, 주문 관리, 배송 관리, 상품 문의, 입금 확인, 검색 결과 및 기타 정보가 XML로 변환되어 HTML 페이지에 데이터를 기록하듯이 쉽게 데이터를 온라인상에서 교환할 수 있는 것이다[6].

2.2 XSL

XSL은 stylesheet language로서 XML 문서와 데이터의 포맷팅 정보를 기술하기 위하여 개발되었다. XML이 SGML의 서브셋이라면, XSL은 DSSSL (Document Style Semantics and Specification Language)의 서브셋이라고 할 수 있다. SGML이 너무 복잡하여 웹에 그대로 적용하기 어려웠듯이 DSSSL 또한 너무 복잡하여 웹 상에서의 적용이 용이하지 않다. 따라서 XSL은 DSSSL을 기반으로 설계되었지만 웹에서의 활발한 이용과 이를 지원할 수 있는 형태로 설계되었다.

XSL은 웹 상에서 DSSSL과 동일한 기능을 지원함과 동시에 CSS에서 지원하는 모든 기능 지원을 목적으로 한다. CSS보다 강력하고 전문화된 포맷팅 기능을 요구하는 환경이나 XML문서와 같이 상세히 구조화된 문서를 포맷팅할 경우에 사용하는 것이 적합하다[3].

2.3 XML 이메일 서버 시스템

기존의 메일 시스템에서는 보통 HTTP(Hyper Text Transfer Protocol)를 통해서 HTML 문서만을 교환해왔다. 이미 고정되어 있는 태그로 사용자가 원하는 모든 것을 표현하기에 HTML은 그 역할을 다 할 수 없다. 따라서 표현하고자 하는 논리적 구조와 그것의 데이터 타입을 사용자가 직접 정의할 수 있는 XML 문서를 이용하여 메일을 교환한다면 송신자나 수신자 모두에게 매우 유용할 것이다.

현재 사용하고 있는 메일 서버들은 XML과 XSL 문서 송수신에 적합하지 않다. 기존의 메일 송신 방법으로 두 문서를 보낼 경우 서버가 XML과 XSL 문서들을 처리할 수 없기 때문에 XML 문서의 유용한 특징을 이용할 수 없게 된다. 따라서 메일을 보낼 때 XML 문서에 이어 XSL 문서를 임베디드 하여 보내는 XSL 메일 전송 기법을 제안한다. 메일을 보낼 때 XML 문서 뒤에 임베디드 되는 XSL 문서는 주석 처리로 표시되기 때문에 메일 전송에 있어서 문제가 발생하지 않는다. 이 때 송신자는 HTML 형식의 문서를 보냄을 알리는 라디오버튼에 반드시 표기를 해주어야 한다.

XML, XSL 문서가 서버에 도착하면 XSL 메일 서버 시스템은 POP3(Post Office Protocol version 3) 서비스를 이용하여 메일을 가져온다[5]. 아무런 제약 없이 한꺼번에 가져오는 것이 아니라 먼저 메일의 헤더 부분, 바디 부분으로부터 정보를 얻도록 한다. 헤더에서는 이름, 이메일, 도착한 날짜, 제목 등의 송신자 정보를 추출하고 메일의 본문에서 XML 문서를 가져와 XML 데이터베이스에 파일로 저장한다. XSL 문서도 마찬가지로 송신자 정보를 추출한 후 주석 처리로 되어 있는 부분을 가져와 XSL 데이터베이스에 파일로 저장한다. 즉 메일의 내용을 XML, XSL 두 부분으로 나누어 각각의 데이터베이스에 저장하는 것이다. 이렇게 두 문서를 따로 저장하는 이유는 첫째, 사용자가 원하는 요소를 얻고자 할 때에 XML 데이터베이스에

저장되어 있는 XML 파일만을 열고 엘레멘트들을 처리할 수 있는 용이성 때문이고 둘째, 수신된 메일의 문서 스타일과 데이터 정의 방법 등의 검색은 XSL 데이터베이스에서만 이루어진다는 장점 때문이다. 도착한 메일의 본문을 태그로 구분하여 데이터베이스에 따로 저장된 문서들은 XML 내에 정의되어 있는 DTD와 XSL이 제공하는 stylesheet을 제공받아 완벽한 XML 문서로 구성된다. 이 문서는 브라우저에서 제공하는 파서에 의해 XML 문서의 유효성을 검증받은 후 브라우저를 통해 그 결과 파일을 볼 수 있다. 도착한 메일을 구분하여 XML, XSL 데이터베이스에 따로 저장하기 위해서는 메일 내용을 태그로 파싱하여야 한다. 먼저 메일의 정보를 얻기 위하여 헤더 부분에서 송신자의 이름, 이메일 주소, 도착한 날짜, 제목을 추출하고 쿼리문을 사용하여 필드별로 저장한다. 현재는 서버마다 이메일을 송신할 때 헤더가 각각 다르기 때문에 이 작업을 하는 것이 쉽지 않다[4]. 헤더의 정보를 추출한 후에는 본문의 내용에서 <?xml version="1.0" encoding="euc-kr"?>로 시작하는 XML 문서와 XSL 문서를 따로 파일로 만들어서 각각의 데이터베이스에 저장하도록 한다. 이 때 생성되는 XSL 파일의 이름은 XML 파일의 stylesheet 선언부에 있는 이름과 같도록 저장한다[3]. XML 메일 서버 시스템의 전체적인 구조는 그림 1과 같다.

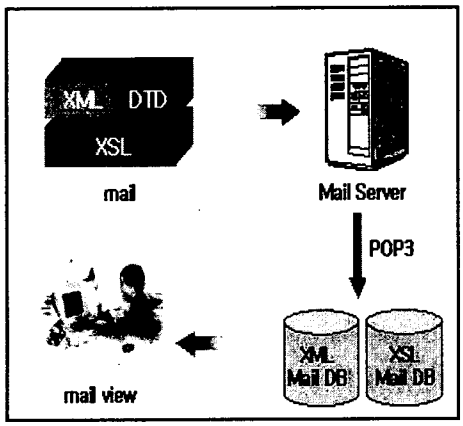

그림 1 XML 이메일 서버 시스템의 구조

현재 사용하고 있는 메일 서버는 메일을 보내기 위해 내용 작성을 하는 부분을 body로 저장한다. 따라서 실제 본문인 XML 문서 안에 존재하는 body는 HTML 문서 형식을 따르기 위해 자동으로 table 태그로 바뀌게 된다. 이런 경우 따로 데이터베이스에 저장되는 XML 문서는 body 태그가 없는 불완전한 XML 문서로 저장되고 브라우저로 XML 파일을 실행시키면 에러 메시지를 띄우게 된다. 이러한 문제점을 해결하기 위하여 XML 문서 내에 가장 먼저 나오는 table 태그와 마지막 table 태그를 body 태그로 자동으로 바꿔주도록 설계한다.

이러한 과정을 거쳐서 새로이 생성되는 XML 파일은 브라우저를 통하여 유효성이 검증되고 이 파일을 실행시키면 XSL이나 DTD에 대한 에러 없이 완벽한 XML 문서로 보여진다.

다음 그림2는 XML과 XSL을 추출해 내는 과정을 보여준다. <?xmlversion="1.0" encoding="euc-kr" ?>로 시작하는 XML 문서와 주석 처리 기호 아래의 XSL 문서를 도착한 메일로부터 분리하여 각각의 파일로 만든다. 수신된 메일의 stylesheet 부분에 명시되어 있는 XSL 파일의 이름을 변수 Test1에 저장하여 같은 이름의 XSL 파일을 결과물로 생성하도록 한다.

```
//xml
 . . .
sprintf(FileName, "d:\\mail-%04d.xml",iMessageId+1);
. . .
while (file_check)
 {
   if(strBuf.Find("?xml")==1) {
     fputs(strBuf, pFile);
     file_check=s.ReadString(strBuf);
     if(strBuf.Find("href")!=-1) {
          int y;
          y=strBuf.Find("href");
          if(strBuf.Find(".xsl")!=-1){
               int a;
               a=strBuf.Find(".xsl");
               Test1 = strBuf.Mid (y+6, a-y-6);
          }
   }
  . . .
 sprintf(FileName_xsl, "d:\\%s.xsl",Test1);
 FILE    *pFile_xsl = fopen(FileName_xsl, "wb");

//xsl
 . . .
if(strBuf.Find("/xsl:stylesheet")==1) {
      int x;
      x=strBuf.Find("/xsl:stylesheet");
      Test5 = strBuf.Left(x+16);
      fputs(Test5, pFile_xsl);
      ch1=1;
      fclose(pFile_xsl);
      break;
   }
   fputs(strBuf, pFile_xsl);
 }
. . .
s.Close();
```

그림 2 XML과 XSL의 추출 과정

III. FAI(Filtering Agent Interface)

3.1 모듈별 설계

XML 데이터베이스에 일단 저장된 내용을 카테고리별 분류를 위하여 FAI(Filtering Agent Interface)를 설계한다.

FAI(Filtering Agent Interface)는 다음과 같이 크게 세가지 모듈로 구성되어 있다.

①**The Mail Interface**: 새로운 메시지가 도착하면 먼저 사용자의 메시지 처리과정을 관찰하여, 특징 추출 및 규칙(rule) 형성에 도움을 준다.

②**The Rule Generation module**: 메시지 처리 과정에서 특징 추출하여 룰(Rule)을 형성한다.

③**The Classification engine**: 형성된 룰(Rule)을 기반으로 새로운 메시지가 도착하면 자동 분류 및 관리를 한다.

각 모듈의 설명은 다음 표[1]와 같다.

Modularity of the agent	Description
The Mail Interface	- observe the user handling mail - user actions on mail messages are recorded by the mail interface for later rule generation - negotiate with the user which automated action to perform
The Rule Generation module	- the user profile can reflect the user over a given time : the rule generation module is responsible for handling this
The Classification engine	- this module tests each new mail message against the rule base and analysis the results

표[1] 이메일 필터링 에이전트의 모듈별 설계

FAI의 모듈은 위에서 제시한 것과 같이 세 부분으로 되어 있으며, 메일 인터페이스의 한 모듈로 XML 메일 서버를 사용한다. 다음으로 데이터의 분류에 있어서, 메일의 내용을 분석하여 그룹화 하도록 한다. 서버에 도착한 메일의 내용을 분석하여 해당하는 데이터베이스의 매칭되는 테이블에 저장한다. 본 연구에서는 그룹화를 위하여 특정단어(Keyword) 검색 방법을 이용하기 때문에 매칭 그룹을 위한 키워드를 미리 지정해 놓아야 한다. 이러한 분류과정을 거쳐서 수신된 메일은 그룹화되어 저장된다.

3.2 필터링 에이전트 인터페이스 [Filtering Agent Interface] 구조

다음 그림 3은 XML 메일 서버 기반 메일 필터링 에이전트 인터페이스(FAI) 흐름도 이다.

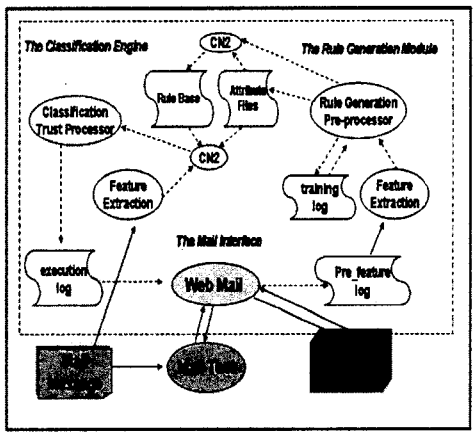

그림 3 메일 필터링 에이전트 인터페이스

이메일 필터링 에이전트의 모듈은 위에서 제시한 것과 같이 세 부분으로 되어 있는데, 이로 인하여 서버에 도착한 메일의 내용을 분석하여 해당하는 데이터베이스의 테이블에 저장하고 그에 대한 확인 및 전달까지 가능한 FAI [Filtering Agent Interface]가 구현된다.

IV. 결론

이메일은 웹의 발전과 함께 가장 널리 이용되고 있는 웹 소프트웨어이다. 현재 이메일을 통해 많은 양의 정보들이 오고가고 있고, 사용자들 또한 이중에서 본인에게 중요한 정보만을 얻고자 한다. 이를 위해 본 논문에서 제안하는 메일 필터링 에이전트 인터페이스(FAI: Filtering Agent Interface)는 XML 메일 서버에 도착한 메일의 내용을 개개인의 상황에 맞게 자동분류 및 관리를 가능하게 해 주는 것이다.

FAI(Filtering Agent Interface)의 설계를 위한 XML 메일 서버는 HTML 형식의 문서와는 달리 얻고자 하

는 element를 메일로부터 쉽게 분리해 낼 수 있으므로 대량의 메일을 수신한 경우 메일들을 효과적으로 관리할 수 있으며, XML 부분과 XSL 부분을 따로 저장하여 사용자가 원하는 요소들을 쉽게 추출할 수 있는 장점을 가지고 있다.

엄청난 지식더미 속에서 개인에게 가장 직접적이고 개인적인 정보의 송수신이 이메일을 통해서 이루어지는 현 상황에서 FAI(Filtering Agent Interface)는 유용하게 활용될 수 있을 것이다.

참고문헌

[1] 이종석, 황대훈, "Internet과 XML", 한국멀티미디어학회, 2권 1호, pp. 21-24, 1998.

[2] 김성동, 김성환, 최기호, 김수훈, "XML(SMIL) 기반의 멀티미디어 메일 저작도구의 설계 및 구현", 한국멀티미디어학회, 2권 2호, pp. 445-449, 1999.

[3] 최영인,곽미라,조동섭, "XSL 메일 서버를 이용한 문서 관리 시스템", 2002년도 대한전기학회 하계학술대회논문집, D권, pp.2837-2839, 2002.

[4] System & Network Administration, "Administering Electronic Mail", SUN PRESS, 1990.

[5] Jonathan B. Postel, "SMTP(Simple Mail Transfer Protocol)", RFC 821, Network Workong Group, 1992.

[6] Jae-Young Kim, James Won-Ki Hong, "Design and Implementation of a Web-based Internet/Intranet Mail Server Management System", Proceedings of the IEEE International Conference on Communications, Volume 1, pp. 641-645, 1999.