XML을 이용한 웹 문서 수집기 설계 및 구현

이새벽*, 임희석**
*한신대학교 컴퓨터공학과
**고려대학교 컴퓨터교육과
e-mail: marsturn@gmail.com

Design and implementation of web-robot using XML

Sae-Byuk Lee*, Heui-Seok Lim**
*Dept of Computer Engineering, Hanshin University
**Dept of Computer Science Education, Korea University

요 약

웹2.0, RIA(Rich Internet Application)의 발전으로 웹 기반 서비스가 다양해지고 기존의 응용프로그램 역시 웹 기반 인터페이스로 제공되면서 사용자 또한 단순 사용자가 아닌 서비스를 제공하는 컨슈머(Consumer)의 형태가 되었다. 따라서 웹 문서는 더욱 방대해 지고 검색, 분류, 색인 등을 위해서 웹 문서의 수집이 새로운 형태로 필요하게 되었다. 그러나 기존의 데이터베이스 사용 방법이나, 문서의 전문을 파일형식으로 저장하는 방법은 웹문서를 이용하여 다양한 컨텐츠를 제공하기에 적합하지 않다. 그러므로 본 연구는 웹 문서를 파싱(Parsing)하여 필요한 부분을 XML파일 형태로 저장하여, 재사용성을 높이는데 초점을 맞추어 HTML을 파싱하고 자동으로 임의의 파일을 수집하는 문서수집기를 구현하게 되었다.

1. 서 론

국내의 인터넷 이용률은 꾸준히 증가 추세에 있으며, 현재 국내 인터넷 사용자 수는 약 34,570(천명)을 넘고 있다. 그리고 인터넷 사용자를 대상으로 한 조사에서 전체 사용자의 73.2%가 하루 1회 이상 인터넷을 사용하고 있는 추세이다.

인터넷상의 정보는 매일 방대한 양이 생성과 소멸을 반복하고, 수시로 변경된다. 이러한 인터넷 정보의 유동성 때문에 사용자에게 효율적인 정보제공을 위한 웹로봇 개발 및 관련연구는 WWW(World Wide Web)가 보편화된 1990년대 중반부터 지속적으로 이루어지고 있다.[1]

최근에는 대부분의 포털에서 지원하는 각종 검색 서비스를 통해 지식을 얻는 수단이 오프라인에서 온라인으로 변화 하였다. 하지만, "구글해킹"이라는 말이 생겨날 만큼 검색을 제공하기 위해 웹문서를 자동으로 색인하는 웹로봇의 성능이 좋아지고 있지만, 그에 따른 부작용이 생겨나고 있다. 개인의 메일이나, 웹사이트의 관리자 페이지까지 수집하기 때문이다.

본 논문에서는 현재까지 웹로봇이 검색서비스에서 사용하기 위해 문서를 수집하였다면, 앞으로는 수집한 문서를 구조화 하여 다양한 서비스를 제공할 수 있도록 하기 위한 웹로봇(문서수집기)의 설계 및 구현에 관한 방법을 연구하였다.

2. 관련 연구

2.1 웹로봇의 개념

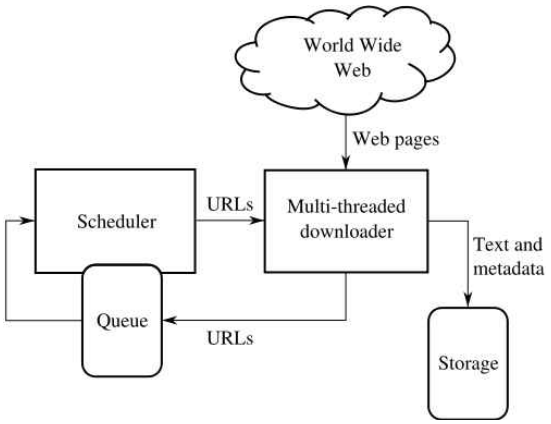
웹로봇은 1993년 전 세계 웹서버의 숫자를 파악하기 위해 만들어진 "World-Wide Wanderer"를 시작으로 초기의 웹로봇은 정보검색을 위해 연구되었다. 당시의 웹로봇은 인터넷 정보를 보다 빠르고 정확히 찾을 수 있는 정보검색엔진의 한 부분으로서 '관리자의 개입 없이 중복되지 않은 URL을 자동으로 찾아가 인터넷 페이지 정보를 축적하여 사용자가 검색할 수 있도록 하는 소프트웨어'라 정의한다.[2]

본 논문에서 다루고자 하는 웹로봇은 '불특정 다

수의 웹사이트에서 웹문서를 수집하고 확장이 가능한 XML형태로 재가공하여, 차후 개발될 컨텐츠에 널리 쓰일 수 있도록 하는 소프트웨어'로 정의하고, 설계 및 구현에 초점을 맞추었다.

2.2 웹로봇의 동작원리

일반적으로 웹로봇은 [그림 1]에서와 같이 스케줄러, 다운로더로 나눌 수 있다. 처음 URL을 지정해 주면 해당 URL로 접속하여, 문서를 수집하고, 문서에서 링크를 얻어서 스케줄러에 등록하는 형식으로 이루어져 있다.



[그림 1] 웹로봇의 구조

스케줄러에 등록된 URL은 우선순위를 정하여, 멀티쓰레드 다운로더를 생성하게 된다. 멀티쓰레드란 하나의 프로세스를 쪼개어 여러 개의 일을 동시에 처리할 수 있도록 하는 것이다. 멀티쓰레드 다운로더는 해당 URL에 네트워크에 연결하여, 웹 문서를 요청하게 된다. 서버에서 응답이 오면, 그 내용을 파싱하여 데이터를 스토리지에 넣고, 해당 문서에서 발견된 URL은 Queue에 넣어서 위에 스케줄러에 등록하고 위의 작업을 반복하게 된다.

2.3 웹로봇의 연구동향

정보검색의 발달로 현재의 웹로봇은 HTML 문서뿐만 아니라 XML, MS오피스문서, Acrobat문서 등 다양한 문서들의 파싱을 지원한다. 또한, 동영상과 이미지를 처리하여 태그를 달아 문서처럼 색인하는 것도 현재 연구가 이루어지고 있다. 또한 불필요한 자료를 제거하고, 필요한 자료를 추출해 내는 것에 대한 연구가 이루어지고 있다.

2.4 XML의 개념

XML(eXtensible Markup Language)은 1996년 W3C(World Wide Web Consortium)의 후원으로 형성된 XML Working Group에 의해 개발되었다. XML은 SGML(Standard Generalized Markup Language)를 기반으로 만들어진 간단하고 매우 융통성 있는 텍스트 포맷으로, 대규모의 전자 출판이나 웹에서 구조화된 폭넓고 다양한 문서들을 상호 교환 가능하도록 설계된 표준화된 마크업(mark-up) 언어이다.

XML의 특징 중 하나가 바로 구조화가 가능하다는 것이다. 구조화란, 사용자가 직접 문서의 구조를 정의 할 수 있다는 것이다. 따라서 문서의 필요한 부분을 빠르게 알 수 있으며 검색할 수 있다.

2.5 XML의 활용분야

XML의 활용분야는 무한하다. 가장 흔하게 XML을 접할 수 있는 것이 RSS인데, 이 RSS는 데이터베이스로부터 문서를 정해진 폼에 입력하여 보여주는 것으로 현재 각종 언론사의 홈페이지나, 게시판, 블로그 등이 새로 올라온 문서를 홈페이지에 접속하지 않고 RSS리더를 통해서 빠르고 정확하게 볼 수 있도록 제공하고 있다.

또한, EDI(Electronic Data Interchange)에서도 새로운 대안으로 떠오르고 있다. 기존의 EDI를 사용하는 것보다. 사용이 용이하고, 유연한 확장성을 가지고 있기 때문이다.

그 외에도 XML의 기술은 최근에 각종 문서편집기나 데이터베이스에서도 지원하며, 데이터베이스 혹은 기존 문서 포맷의 대안으로 떠오르고 있다.

3. 웹로봇 시스템

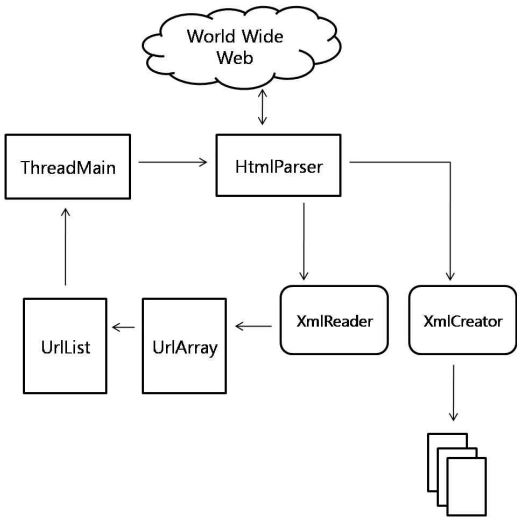
본 연구를 통해 구현된 웹로봇 시스템은 2장의 개념을 적용하여 작성 되었다. 불특정 다수의 웹페이지에서 타이틀, 텍스트, 링크 등 필요한 정보를 XML형태로 저장하고 있으며, 추가적으로 필요한 정보에 대해서는 확장이 가능하도록 설계해 놓았다.

3.1 설계

본 시스템의 설계는 클라이언트 프로그램 형식으로 작성이 되었으며 어느 환경에서나 사용이 가능하도록 JAVA로 작성 하였다. 데이터의 처리는 앞에서 언급했듯, 데이터베이스를 전혀 사용하지 않고 XML형태로 저장하며 URL의 디렉터리를 기준으로 파일시스템에 저장된다.

전체적인 시스템의 구상도는 [그림 2]와 같이

ThreadMain에서 시작하여, HtmlParser를 통해 해당 사이트에 접속하여 Html문서를 파싱하고 그 것을 XmlCreator를 이용해서 XML형식의 문서로 생성하고, 문서에서 얻은 URL을 XmlReader로 기존 문서와 비교하여, 중복 제거를 한 다음 UrlArray에 추가하여 다음 작업리스트로 등록한다.

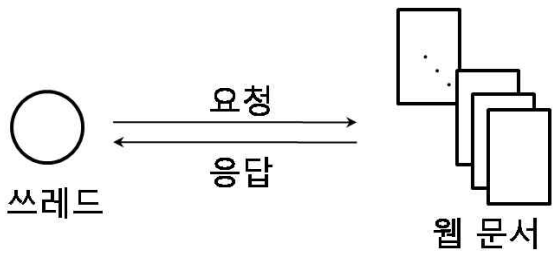


[그림 2] 웹로봇 구조

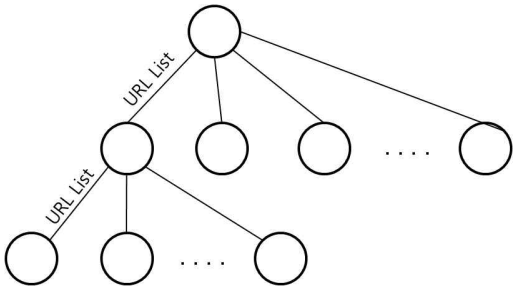
3.1.1 Multi-Thread

본 시스템에서는 프로세스를 최대한으로 활용하고, 성능을 높이기 위해 Multi-Thread 기법을 사용하여 구현하였다. Multi-Thread를 URL별로 무작위로 생성하지 않고, [그림 3]에서처럼 한 사이트의 URL을 리스트로 만들어 한 Thread에서 리스트 들을 순회한다. 여러 개의 리스트가 동시에 생성되므로, [그림 4]와 같이 리스트 들을 순회하는 Thread가 여러 개가 생성되는 것이다.

쓰레드를 무한대로 생성하면, 네트워크와 시스템에 부하가 걸리므로, 한 쓰레드 내에서 반복적인 방법으로 여러 사이트를 탐색하고 동시에 생성되는 쓰레드의 최대 수를 256개로 한정하였다.



[그림 3] 하나의 쓰레드가 하는 일



[그림 4] 쓰레드 생성

3.1.2 Html Parser

Html Parser는 해당 웹사이트에 접속하여 URL에 대한 문서를 요청하고 응답을 받아 HTML구문을 분석하는 것이다. 받은 문서를 스트림으로 읽어 파일이 끝날 때까지 태그가 발생할 때 마다, 각 태그에 적합한 명령을 처리하게 된다. 현재 프로그램에서는 일단, 해당 문서에서 다른 페이지로의 링크들을 추출하고, HTML Tag를 제거한 텍스트를 추출하며, 문서의 타이틀 등을 추출한다.

3.1.3 XML Process

[그림 3]의 XmlCreator는 Html Parser를 통해서 얻은 정보를 XML파일로 작성하고 파일 시스템에 저장하는 역할을 한다. XmlReader는 생성된 XML 파일을 읽어와 중복된 URL 리스트를 제거하고, 파일이 같을 경우 텍스트 내용을 MD5형태로 변환한 뒤 비교하여, 중복 데이터를 최소화 시킨다.

3.2 구현

본 연구를 통해 구현된 웹로봇은 간단하게 콘솔에서 사용할 수 있으며, 어느 환경에서나 JRE가 설치되어 있으면, 사용이 가능하다.

실행방법은 [표 1]에서와 같이 프롬프트에서 실행파일명과 처음에 시작할 URL을 적어주면 된다.

>javac ThreadMain [URL1] [URL2] ..

[표 1] 실행 커맨드

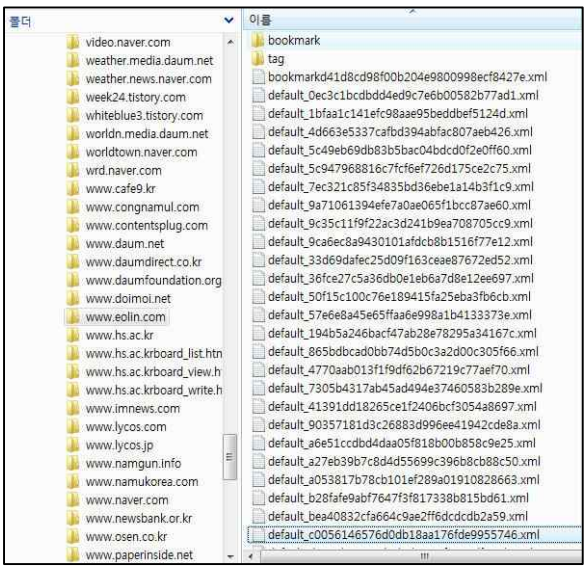
프로그램을 실행 시키면, [그림 5]과 같이 시작 URL부터 시작하여 자동으로 문서를 수집해 온다.

수집해온 문서들은 프로그램이 있는 폴더안의 /web_document/라는 폴더 안에 URL의 디렉터리를 참조하여 생성되고, 파일명 또한 "[URL의 파일명]+[QueryString의 MD5Hex 값].xml" 형태로 저장

되게 된다.

수집된 파일을 보기위해 /web_document/ 폴더를 열어보면 <그림 6>와 같이 각 호스트의 디렉터리별로 저장되어 있다. 그리고 파일을 웹브라우저로 열어보면 <그림 7>와 같이 XML형태로 저장되어 있는 것을 볼 수 있다.

[그림 5] 문서가 수집되고 있는 실행화면



[그림 6] 수집된 파일을 파일은 브라우저를 통해서 볼 수 있다.

4. 결론

하루가 다르게 변해가는 현대시대에서는 누가 얼마나 많은 지식을 가지고 있는가에 의해 그 사람의 능력이 평가된다. 이미 서론에서 언급한 것처럼 지식을 얻는 방법이 오프라인에서 온라인으로, 특히 검색으로 변화하고 있다.

좋은 검색 서비스를 제공하기 위해서 검색엔진의 모델을 연구하는 것도 중요하지만 더 많은 정보를 수집하는 웹로봇을 연구하는 것도 좋은 방법이 될 수 있다.

본 연구를 통해 구현된 웹로봇은 웹문서를 XML 형식의 문서로 생성하므로 기존의 복잡한 HTML형식의 파일을 구조화 시킬 수 있다. 하지만 문서의 중요 컨텐츠 부분을 식별할 수 없다는 단점이 있다. 따라서 문서에 중요한 내용이 양이 적고 다량의 메타데이터로 구성된 문서에서 키워드 식별의 어려움을 가질 수 있다. 이러한 문제점을 극복하기 위해선 같은 도메인 상에 여러 번 출현하는 단어를 도메인에 대한 키워드로 하고, 각 페이지 별로 특별하게 출현하거나 빈도가 잦은 키워드를 따로 분리해서 저장해야 한다.

참고문헌

[1] 김광명, 백상규, 김선호 "업종별 포탈 사이트의 효율적 정보제공을 위한 웹로봇 시스템 개발에 관한 연구", 한국경영과학회 학술대회논문집, 한국경영과학회, 2003

[2] 박규석, 이충석, 김 성, "서버 부하를 고려한 동적 로봇에이전트 시스템의 설계 및 구현", 한국정보처리학회 논문지, 7권 11호, 2000