웹 사이트 컨텐츠 변경 모니터링 시스템

김원중* · 조이기* · 손철수*

The Monitoring System for Informing the Change of Contents on the Web Sites

Won-jung Kim* · Lee-Gi Cho* · Cheol-Su Son*

이 논문은 2002년도 순천대학교 공과대학 학술재단 연구비 지원에 의해 수행되었음.

요 약

웹의 급속한 보급은 전세계에 흩어져 있는 무한한 정보를 손쉽게 얻을 수 있도록 하였지만, 인터넷 공간의 엄청난 정보의 양은 사용자들이 관심을 가지고 있는 정보의 변경을 곧 바로 인식하는데 많은 어려움을 주고 있다. 즉, 사용자가 수시로 변하는 웹 문서의 변경을 탐지하기 위해서는 해당 사이트에 접속하여 일일이 검사하여야 한다. 따라서 웹에서 변화된 정보를 자동적으로 감지하여 사용자에게 알려주는 정보변화 감시(Information change monitoring)기능을 수행하는 로봇의 개발이 필요하다.

본 논문에서는 웹 사이트 문서의 변경을 모니터링 하기 위하여 모니터링할 대상 URL, 모니터링 조건, 모니터링 주기 등을 사용자가 정의하면, 변경이 발생할 경우 사용자에게 알람이나 E-mail를 통하여 자동으로 통지하여 주는 웹 사이트 컨텐츠 변경 모니터링 시스템을 설계 및 구현하였다.

본 연구를 통하여 웹 문서를 의미있는 단위로 구조화시키는 변경 방법과 HTML 태그를 이용하여 의미있는 단위로 분류시키는 방법을 제시하였다.

ABSTRACT

Fast spreading of web made we get easily the vast amount of information all over the world, but quantity of great information on the Internet space is giving much troubles to recognize change of information that users are interested soon justly. That is, users must connect and examine one by one to relevant site to detect change of web documents that changes from time to time. Therefore, the development of Robot which accomplish Information change monitoring function that sense automatically changed contents and inform to user is required.

In this paper, we designed and implemented Web site contents change monitoring system, which notify automatically the change of Web documents to users through alarm or E-mail if user defines target URL to do monitoring, monitoring condition, monitoring period etc. And we presented the method that structure and classify Web Documents to semantic units using HTML Tag. Also, we introduced the concept of virtual key to manage position of word to watch some change efficiently.

키워드

Information change monitoring, Web robot, Virtual key, HTML

* 순천대학교 컴퓨터과학과

접수일자 : 2002. 6. 18

Ⅰ. 서 론

인터넷이 처음 등장 한 뒤 네트워크의 대역폭이 커지고 컴퓨터의 성능이 개선되는 등 하드웨어의 발전으로 인터넷을 기반으로 한 전 세계적인 정보공유가 가능해졌다. 여기에 크게 기여한 기술 중 하나는 HTML(Hypertext Markup Language)이다. 비교적 간단한 HTML 규약을 구현해 주는 웹 브라우저만 있으면 HTML로 된 문서를 전 세계 어느 곳에서나 시간과 장소에 제약 없이 볼 수가 있다. 이러한 대중성과 범용성을 기반으로, 웹 상의 HTML 문서가 급속히 많아짐에 따라 필요한 정보를 찾기 위한 검색엔진이나 디렉터리 서비스와 같은 정보 검색 기술이 발전하였다[1,5,6].

그러나 이러한 서비스는 웹상의 문서 검색과 분류가 목적이기 때문에 획일적이고 정적인 특징을 가지고 있다. 좋은 웹 사이트의 요건 중 하나는 그 사이트 데이터의 정확성이 유지되기 위하여 제 때에 최신 데이터로 변경되는 것이다. 그런데 이러한 최신 데이터로의 변경은 웹 사이트 자체에서 발생하지만 변경된 데이터에 대한 정보를 필요로 하는 곳에는 통지가 되지 않기 때문에 변경 여부를 알아야 하는 사용자는 주기적으로 사이트를 방문하여서 변경 사항을 모니터링하고 감지하여야 한다. 예를 들어, 경쟁사들의 신상품에 대해서 모니터링이 필요한 사용자의 경우 매일 경쟁사들의 신상품 카타로그가 있는 웹 페이지를 방문하고 수작업으로 그 목록을 관리하여야 한다. 이때 검색 엔진을 사용할 수도 있지만 최신 데이터나 잘 정리된 데이터를 얻을 수 없는 것이 일반적이다.

따라서 본 논문에서는 웹사이트 문서의 변경을 모니터링하기 위하여 변경사항을 모니터링할 대상 웹 페이지와 변경의 조건을 지정한 뒤, 모니터링할 주기를 설정하고, 모니터링을 실시한 뒤 변경이 발생할 경우 그 내용을 사용자에게 통지하는 일련의 처리를 자동으로 할 수 있는 웹사이트 컨텐츠 변경 모니터링 시스템 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 2장은 관련연구로 웹 로봇에 대해 기술하고, 3장에서 HTML 문서의 변경을 의미적 그리고 외형적 변경으로 분리하여 정의하고, HTML 문서를 의미적 항목과 물리적 항목으로 구분하여 모니터링하는 방법을 제시하였다. 4장에서는 변경 모니터링 시스템 구현을 위한 시스템 구조와 각 모듈의 기능들에 대해 설명하였다. 그리고 5장에서는 결론 및 향후 연구 과제를 제시하였다.

Ⅱ. 관련 연구

전 세계적으로 헤아릴 수 없을 정도의 많은 정보들이 인터넷을 통하여 만들어지고 유통되고 있다. 이들 정보들은 사용자에게 통보 없이 수시로 변하고 있지만, 사용자가 개별적으로 일일이 검사하는 것은 매우 어려운 일이다. 이러한 문제점의 근본적인 이유는 웹상의 문서가 언제, 어떻게 그리고 무엇이 변경될지 웹사이트 관리자외에는 알 수 없다는 것이다. 따라서 웹상의 문서의 내용이 변경되었을 경우 이것을 사용자에게 자동으로 통보하는 Mind-it, News Index, URLy Warning, TRACERLOCK 등의 개인 웹 로봇에 대한 연구가 진행되어 왔다[2,3,6].

2.1 웹 로봇

웹 로봇은 웹의 하이퍼텍스트 구조의 문서를 찾고, 링크가 지정하는 모든 문서를 다시 재귀적으로 찾는 것에 의해 웹상에 흩어져 있는 문서들을 획득하는 프로그램이다. 웹 브라우저는 로봇이 아니다. 그 이유는 웹 브라우저는 인간에 의해 동작되고 참조된 문서를 자동으로 되찾아 주지 않기 때문이다. 웹 로봇의 사용 목적은 웹 사이트 인덱싱, HTML 문서의 링크 검증, 새로운 정보를 찾기 위한 모니터링 등이다. 잘 알려진 웹 로봇만도 수 백 개가 넘는데 대표적인 웹 로봇들을 기능에 따라 표 1과 같이 분류할 수 있다

Table. 1 Category of web robot

목 적	대표적인 웹 로봇
인 덱 싱	GoogleBot, Scooter, InfoSeek Robot, IntelliAgent, Acme.Spider
링크검증	CyberSpyder, GetURL, Inspector Web, JoeBot, Link Validator
통 계	ArchitextSpider, BackRub, BlackWidow, Direct Hit Grabber
변경통지	Mind-it, News Index, URLy Warning, TRACERLOCK

2.2 웹 문서 변경 통지 로봇

웹 문서 변경 통지 로봇들은 주로 웹 문서의 변경을 모니터링하고 전자우편, 웹 브라우저 플러그인 또는 클라이언트 프로그램을 이용하여 사용자에게 통지한다. 전자우편을 이용하는 경우 하루 단위로 변경 사항을 통보하고 클라이언트 프로그램을 이용할 경우는 분 단위로 변경 사항을 통보한다.

2.2.1 Mind-it

Mind-it은 두 가지 방법으로 웹 페이지의 변경을 모니터링할 것인지를 지정할 수 있다. 첫째, 사용자는 변경을 모니터링할 URL을 직접 입력하고, 그림 1과 같이 모니터링할 대상을 페이지, 텍스트, 이미지, 링크 또는 키워드 등에서 선택하고 통지 받을 전자우편 주소를 입력한다. 둘째, 사용자는 금융, 연예, 컴퓨터, 건강, 스포츠 등과 같이 미리 분류된 영역을 선택하고 키워드를 입력하여 통지 받을 전자 우편 주소를 입력한다. 사용자가 변경을 모니터링할 URL을 직접 입력하고 모니터링할 대상을 페이지를 선택하였을 경우 페이지 전체를 대상으로 변경 유무를 판단하는데 웹 페이지의 폰트 크기가 조금만 변경되어도 사용자에게 변경 내역을 알린다[7].

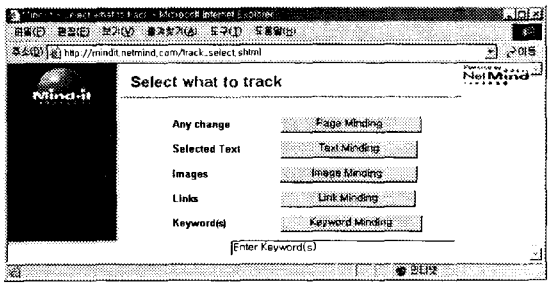


Fig. 1 Selection of monitoring target in Mind-it

2.2.2 TRACERLOCK

TRACERLOCK도 그림 2와 같이 사용자가 뉴스, 금융, 검색엔진, 뉴스그룹, 구인과 경매 등과 같이 미리 분류된 영역을 선택하고 키워드를 입력하면 전자우편으로 변경 내역을 통지 받는다. 뉴스의 경우 15분마다 변경 내역을 통지하지만 유료로 서비스되고 있다. 모니터링 하고자 하는 URL을 직접 입력할 경우 페이지의 내용에 변화가 있으면 사용자에게 변경 내역이 전자 우편으로 통지된다[8].

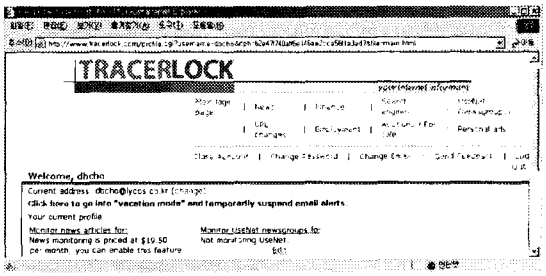


Fig. 2 Monitoring of TRACERLOCK in pre-defined area

2.2.3 URLy Warning

Mind-it와 TRACERLOCK는 서버에서 모니터링이 수행되지만 URLy Warning의 경우 클라이언트에서 모니터링이 실행된다. 사용자는 모니터링하기 위한 URL, 통지 주기와 모니터링할 키워드를 그림 3과 같이 입력하면, URLy Warning은 변경을 감지하고 그 결과를 클라이언트 화면에 출력한다.

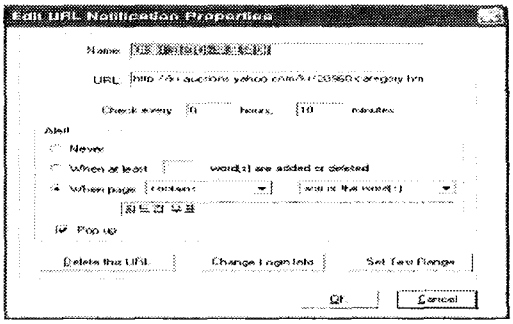


Fig. 3 Monitoring condition define of URLy warning

III. 웹 문서 변경 및 모니터링

3.1 HTML 문서의 변경

초기 웹 사이트의 문서는 단순히 텍스트와 이미지로 구성되었고, 대부분이 HTML 문서였다. 그러나 현재의 웹상의 문서는 HTML 문서뿐만 아니라 여러 가지 목적을 위한 DHTML, XHTML, XML, VRML 등 많은 종류의 문서가 존재한다. 이렇게 여러 종류의 문서가 웹상에 존재하지만 아직까지는 유용한 문

서의 대부분이 HTML 문서로 존재하고 지금도 HTML 문서로 만들어지고 있다. 따라서 본 논문에서 구현한 시스템 역시 현실성 있는 HTML 문서 변경을 모니터링 하기 위하여 설계되었다. 그런데 HTML 문서는 인간이 읽을 수 있게 표현되어지도록 만들어진 것이다. 즉, 응용프로그램이 분석하고 사용하기 쉽도록 구조적인 형태를 가지고 있지 않기 때문에 문서의 내용이 변경되었다는 것을 응용프로그램에 의해 쉽게 판단할 수 없다. 따라서 HTML 문서의 변경은 다음 2가지 분류로 나누어야 한다[1,2].

3.1.1 의미적 변경

앞에서 기술한 바와 같이 HTML 문서는 인간에게 표현되어지기 위한 문서이기 때문에 인간이 그 문서를 읽어서 문서의 내용이 변경되었다고 판단되어져야만 문서가 실질적으로 변경된 것이다. 즉, 의미전달이외에 문자의 효과나 장식은 문서가 지니고 있는 의미에는 영향을 주질 않는다. 그러나 이중 취소선과 같은 효과는 의미의 변경을 일으킬 수 있으므로 본 논문에서 구현한 시스템은 이러한 효과 등을 변경의 의미로 사용할 것인지를 사용자에게서 입력을 받아서 처리한다. 구체적인 변경의 정의는 본 장의 3절에서 기술한다.

3.2.2 물리적 항목

물리적 항목이란 문서의 길이, 문자의 크기, 문자의 글꼴, 주석, 프레임 등 의미와 관련 없는 항목이다. 즉 웹 문서의 외형적인 부분만 변경된 것이다. 실제적으로 이러한 외형적인 변경을 모니터링할 경우가 없으므로 이러한 물리적 항목의 변경에 대한 모니터링은 본 논문에는 포함되지 않는다.

IV. 시스템 설계 및 구현

4.1 모니터링 시스템의 목적

정보변화 감시 기능은 하루가 다르게 변화하는 정보들을 사용자가 개별적으로 검사하는 수고를 덜고 자동으로 변화된 사실을 감지하여 사용자에게 통보하는 역할을 한다. 즉, 사용자가 감시를 원하는 URL(Uniform Resource Locator)을 지정하고 모니터링할 항목을 설정하면 시스템은 그 HTML 문서를 사용자가 정의한 주기마다 점검하여 감시 대상 단어의 추가, 삭제, 빈도, 갱신의 변경을 모니터링 한다.

본 논문의 목적은 모니터링에 의해서 위에서 정의한 문서 내용의 변경이 감지되면 감시 대상 단어의 변경 내역을 사용자의 컴퓨터에 전송하고 사용자의 컴퓨터에 설치된 프로그램은 사용자에게 변경된 내역이 도착했다는 사실을 알람 기능으로 인식할 수 있도록 하는 웹 컨텐츠 변경 모니터링 시스템을 개발하는 것이다.

4.2 모니터링 시스템 운영 환경

모니터링 시스템은 인터넷에서 클라이언트/서버 환경으로 구성된다. 클라이언트에서 변경 항목을 모니터링하기 위한 URL, 변경 모니터링 항목, 주기 등을 웹 서버에 브라우저로 설정하고 서버는 주기적으로 변경사항이 발생하였는지를 점검하여 그 결과를 서버에 보관하고 클라이언트에 TCP/IP 소켓 통신을 이용하여 통지한다. 모니터링 시스템은 서버, 클라이언트 그리고 서버와 클라이언트간의 통신으로 구성된다.

4.2.1 서 버

서버의 운영체계는 Red Hat Linux로 선택하였다. 이것은 인터넷에서 무료로 다운 받을 수 있고 성능이 뛰어나다. 특히, 시스템 개발, 운영과 관리에 필요한 소프트웨어가 무료로 제공되는 특징이 있다. 프로그램 언어는 자바를 사용하였다. 자바는 플랫폼에 독립적인 특성을 가지고 있어서 향후 서버와 프로그램을 개인용 프로그램으로 전환하여 사용할 경우 이식성을 높이기 위해서이다. 웹 서버는 아파치 소프트웨어 파운데이션에서 무료로 제공하는 TOMCAT이라는 웹 서버를 사용하였다. 그 이유는 웹 서버에서 제공하는 화면을 서블릿으로 구성하였기 때문이다. 즉, TOMCAT은 별도의 서블릿 엔진을 설치할 필요 없이 웹 서버에 서블릿 엔진을 갖추고 있기 때문이다. 데이터베이스는 MYSQL을 사용하였는데, 소프트웨어가 무료로 제공되지만 유저관리, 트랜잭션관리 등이 용이하고 원격 접속이 가능하기 때문이다[9,10,11,12].

4.2.2 클라이언트

클라이언트는 하드웨어 플랫폼이나 운영체계에 제

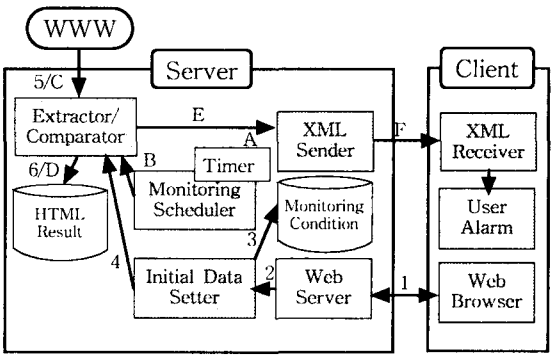
한없이 웹 브라우저가 동작하고 자바 응용프로그램을 실행할 수 있는 자바 가상머신이 설치된 환경이면 충분하다. 즉, 사용자가 브라우저로 서버에 접속하여 URL을 지정하거나 모니터링 항목에 대한 변경 감시 주기를 설정한다. 그리고 변경 내역을 통지 받기 위해서는 자바 응용프로그램이 사용자 컴퓨터에서 상주하여야 하기 때문에 자바 가상머신이 사용자 컴퓨터에 설치되어 있어야 한다. 대부분의 웹 브라우저나 자바 가상머신은 무료로 제공되기 때문에 클라이언트 프로그램 설치에 별도 비용이 들어가지 않는다.

4.2.3 서버와 클라이언트 통신

웹 컨텐츠 변경 모니터링 시스템에서 서버와 클라이언트는 웹 서버와 웹 브라우저의 통신이기 때문에 기본적으로 HTTP(HyperText Transfer Protocol)를 사용한다. 그런데 서버는 사용자가 지정한 URL의 변경을 모니터링하고 변경 사항을 감지하였을 경우 클라이언트에 통지하여야 하는데, 웹 브라우저를 통한 통지는 세션 유지 등에 제약 사항이 많다. 따라서, 클라이언트에 소켓 통신 기능이 있는 자바 어플리케이션을 설치한다. 클라이언트는 항상 서버로부터 변경 내역을 통지를 받아야 하므로 소켓 서버형으로 프로그래밍 한다. 이때, 전송되는 전문의 데이터를 클라이언트가 쉽게 사용할 수 있도록 하였다. 사용자에게 변경 모니터링 항목이 변경되었음을 알람 처리하고 그 데이터를 XML형태의 파일로 보관되도록 한다. 변경된 내용을 클라이언트에 XML 파일로 보관할 때, 변경 내역을 누적하여 통보 받은 이력을 알 수 있도록 한다.

4.3 시스템 구성도

개발 시스템은 변경 모니터링을 하기 위한 조건을 설정하는 Initial Data Setter, 문서를 다운로드하고 변경 여부를 점검하는 Extractor/Comparator, 주기적으로 Extractor/Comparator을 실행시키는 Monitoring Scheduler, 변경된 내역을 송신하는 XML Sender로 구성되어 있고, 클라이언트에는 XML Receiver가 설치된다.



범 례
- 1,2,3,4,5,6 : 최초 모니터링 조건 설정 및 기준 문서 다운로드
- A,B,C,D,E,F : 모니터링 시작, 문서 다운로드, 비교 및 통지

Fig. 4 Structure of web contents monitoring system

4.3.1 Initial Data Setter

Initial Data Setter는 웹사이트의 HTML 문서의 변경을 모니터링하고 통지하기 위해서는 가장 기본이 되는 정보를 Monitoring Condition 테이블에 설정한다. 이 정보에 저장된 URL을 기준으로 Extractor/Comparator는 웹사이트 HTML 문서를 모니터링하고, 저장된 주기를 기준으로 Monitoring Scheduler는 Extractor/Comparator에게 HTML 문서를 지정된 웹사이트로부터 가져와서 변경 여부를 판단하고, 클라이언트 정보를 기준으로 XML Sender는 변경 내역을 클라이언트에 보낸다.

4.3.2 Monitoring Scheduler

Monitoring Scheduler는 Monitoring Condition 테이블에 저장된 주기에 따라 Extractor/ Comparator를 실행시켜 해당 URL을 웹사이트에서 다운로드하여 변경여부를 비교하도록 한다. 실제 구현에 있어서 Extractor/Comparator를 특성시점에 실행시켜야 하는데, 기준 데이터에는 변경 주기만 있으므로 스케줄에 따라 정확히 실행되도록 하려면 실행할 특정 시점을 순차적인 리스트로 구성하고 그 리스트에 정의된 순서대로 읽어서 처리한다.

Table. 2 Monitoring cycle of monitoring condition table

기 준 일 시	주기	URL
2001-08-15 10:00:10	10분	U_1
2001-09-09 11:00:20	5분	U_2
2001-10-01 12:00:30	30분	U_3

Table. 3 Schedule list for monitoring scheduler

순번	URL	기 동 일 시
1	U_3	2001-10-01 12:30:30
2	U_2	2001-10-01 12:35:20
3	U_1	2001-10-01 12:40:10
4	U_2	2001-10-01 12:40:20
5	U_2	2001-10-01 12:45:20
6	U_1	2001-10-01 12:50:10

예를 들면, 표 2와 같이 Monitoring Condition 테이블에 모니터링 주기가 설정되어 있고, 2001년 10월 1일 12시 30분 29초일 때 스케줄 리스트는 표 3과 같이 구성되어 있어 Monitoring Scheduler는 순번 '1' 부터 차례대로 기동시킨다. 이러한 스케줄 리스트는 일정한 개수를 갖는 환형 리스트 형태로 구성한다. 만약, Monitoring Condition 테이블의 주기가 변경되면 스케줄 리스트는 재구성되어야 한다.

4.3.3 Extractor/Comparator

Extractor/Comparator는 Monitoring Condition 테이블에 지정된 URL로부터 HTML 문서를 다운로드하고 최초로 다운로드한 문서가 아니면 Monitoring Condition 테이블의 기준에 의하여 변경 여부를 점검하여 변경되었으면 변경내역을 XML Sender에게 전송한다. 그리고 다운로드한 문서를 HTML 테이블에 저장한다.

만약 최초로 다운로드되는 경우에는 HTML 문서가 문법에 맞지 않으면 이 문서의 항목을 분석하고 비교할 경우 오류가 발생할 수 있다. 이러한 오류를 사전에 검출하고 최대한 정정한 다음 분석할 수 있도록 사전처리가 필요하다. 그리고 HTML 형태의 문서는 응용프로그램이 쉽게 다룰 수 있을 만큼 구조적이지 않기 때문에 문서를 구조화시킬 필요가 있다. 따라서, HTML Tidy의 JAVA 버전 API를 이용하여 다운로드 받은 HTML 문서를 정정하고 XML 문서로 변화하여 분석, 비교하고 저장한다.

4.3.4 XML Sender

Extractor/Comparator로부터 통보 받은 변경 내역을 XML 형태의 문서로 변환하여 클라이언트의 XML Receiver에 전송하며, 클라이언트에 접속 요구를 하여 접속되지 않으면 일정 시간동안 재전송을 시도하고 실패하면 E-MAIL로 변경내역을 통지한다.

서버에서 변경 내역을 단순한 TEXT나 HTML 형태로 보내지 않고 XML로 보내는 이유는 클라이언트가 그 데이터를 받아서 다시 가공하여 사용할 수 있도록 하기 위해서이다.

4.3.5 XML Receiver와 User Alarm

XML Sender로부터 변경 내역을 수신하여 파일로 저장하고 그 이벤트를 User Alarm으로 통지한다. 파일 저장 선택 사항에 따라 덮어쓰기를 할 것이지 이어 쓰기를 할 것인지 달라진다.

XML Receiver로부터 파일명을 수신을 받아서 클라이언트 화면에 변경사항이 발생했음을 알람 처리하고, 변경 발생 내역을 통지 받았을 경우 실행될 프로그램이 지정되어 있으면 그 프로그램을 기동시킨다.

4.4 시스템 주요 처리 흐름

웹 컨텐츠 변경 모니터링 시스템은 모니터링 조건 설정, 변경항목 모니터링 그리고 변경 내역을 통지하는 순서로 작동된다. 다음은 모니터링 시스템이 작동되는 주요 기능의 처리 흐름을 기술한다.

4.4.1 모니터링 조건 설정

클라이언트는 웹 브라우저를 이용하여 서버에 접속한 다음 모니터링할 웹 사이트 문서의 URL, 모니터링 조건과 주기를 입력하고 그 데이터를 웹 서버에 전달하면 Initial Data Setter는 그 정보를 웹 서버로부터 전달을 받아서 Monitoring Condition 테이블에 저장한다. Extractor/Comparator는 해당 URL에서 문서를 가져와 문서의 정확성을 확보할 수 있도록 교정하고 HTML Result 테이블에 웹 문서 다운로드 일자와 가상키 리스트를 저장한다.

4.4.2 변경 항목 모니터링

Monitoring Scheduler는 내부 Timer에 의해서 Monitoring Condition 테이블을 읽고 테이블에 저장된 주기로 Extractor/Comparator가 웹 사이트의 페이지를 읽어서 다운로드 하도록 하고 이전에 HTML Result 테이블에 저장한 문서와 비교하도록 한다. 만약 변경되었으면 변경 내역을 XML Sender로 보낸다.

그림 5의 화면은 변경 항목 모니터링을 위해 해당 사용자의 URL 정보와 모니터링 조건을 입력하

는 예이다.

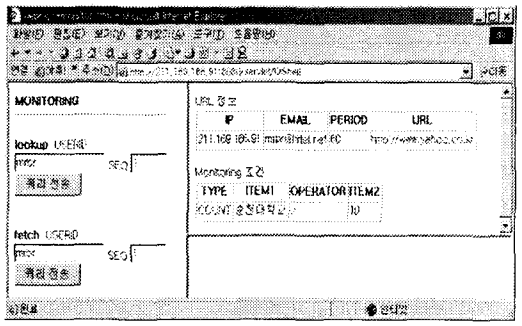


Fig. 5 Screen layout for monitoring

4.4.3 변경 내역 통지

XML Sender는 클라이언트측의 XML Receiver에게 XML로 가공된 데이터를 전송한다. 만약, 클라이언트로 초기 연결이 실패하면 일정한 주기 동안 반복적으로 재연결을 시도하고 지정된 시간을 초과하여 클라이언트에 접속이 불가능하다고 판단될 경우에는 클라이언트의 등록된 메일 주소로 그 변경 결과를 보낸다.

4.5 감시 대상 단어 갱신 검사를 위한 가상 키 생성

앞에서 제시한 감시 대상 단어 존재 유무나 감시 대상 단어 존재 발견 빈도 점검은 매우 간단하게 해결할 수 있다. 즉, 감시 대상 단어 그 자체를 문서에서 찾고, 찾은 것을 계산해서 그 값을 비교하기만 하면 된다.

Table. 4 Method of virtual key selection

종류	내용	정확성	실용성
문서전체	지정어를 제외한 문서 전체	매우높음	매우낮음
문장전체	지정어를 제외한 문장 전체	높음	낮음
문장앞쪽	지정어 기준 문장 앞쪽	높음	높음
문장뒷쪽	지정어 기준 문장 뒤쪽	낮음	낮음
앞 단어	지정어 기준 앞 단어	중간	중간
뒤 단어	지정어 기준 뒤 단어	낮음	낮음
앞뒤단어	지정어 기준 앞뒤 단어	높음	중간
절대위치	문서 시작부터 물리적 위치	높음	없음
상대위치	문서 구조의 상대적 위치	중간	중간

즉, 감시 대상 단어가 검색 단어가 되는 것이다. 그러나 문서에서 특정한 지정어가 다른 내용으로 바뀐 것을 찾아내는 것은 쉽지 않다. 그 이유는 기준으로 삼을 감시 대상 단어 자체가 변경되었기 때문이다.

그러므로 감시 대상 단어의 갱신을 검사하기 위한 기준으로 삼을 대상을 선정해야 한다. 본 논문에서는 이것을 가상키(Virtual Key)라는 용어로 사용하겠다. 가상키의 역할은 감시 대상 단어 갱신을 점검하기 위한 기준이 된다. 앞에서 제시한 감시 대상 단어 존재 유무나 감시 대상 단어 발견 빈도수 점검은 변경되는 문서만 분석해도 그 해답을 얻을 수 있었으나 감시 대상 단어 갱신 검사를 위한 가상키 검출은 기준 문서에서 행해져야 한다. 즉, 변경되기 전의 원본 HTML 문서에서 감시 대상 단어를 찾을 수 있는 가상키를 선정해야 한다. 이때 가상키 선정에는 다음과 같은 여러 가지 방법이 있을 수 있다.

본 시스템에서는 구조적 가상키 선정 방법을 제외하였다. 그 이유는 서론에서 언급한 것처럼 HTML 문서는 구조적이지 못하고 문서의 표현을 강조한 특성을 가지고 있기 때문이다. 따라서 의미적인 가상키 선정 방법을 사용하였다. 표 4에 나타난 것처럼 가상키 선정 기준 단위가 문서, 문장, 단어 등이 있는데, 정확성 측면에서 보면 문서전체, 문장전체 등을 가상키로 사용하면 좋지만 HTML 문서가 반드시 갱신 변경을 검사하려는 감시 대상 단어만 변경되는 것이 아니기 때문에 실용성 측면에서 고려하지 않았다. 그리고 지정어의 앞 단어와 뒤 단어를 가상키로 선정하는 것은 정확성은 높지만, 테이블과 같은 구조에서는 실용성이 떨어지기 때문에 역시 고려하지 않았다. 따라서 본 연구에서는 표 4의 평가에 따라 감시 대상 단어를 기준으로 문장 앞쪽을 가상키로 선정하였다. 문장 앞쪽을 가상키로 선정하는 순서는 그림 6과 같다.

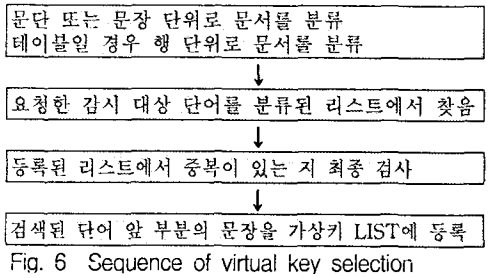


Fig. 6 Sequence of virtual key selection

V. 결론 및 연구 과제

일반적으로 인터넷상의 정보가 변경되는 사실은 과거에 그 웹사이트에서 정보를 획득했던 사용자에게 통지되지 않기 때문에 사용자가 변경된 최신 정보를 얻기 위해서는 그 웹사이트에 재접속하여 정보의 변경을 계속 모니터링하여야 한다. 따라서 본 논문에서는 특정한 사이트의 문서로 국한하여 모니터링하지 않고 일반적인 웹상의 HTML 문서의 의미적 변경을 단순 변경, 집합적 변경 그리고 갱신 변경으로 모니터링하고 그 결과를 사용자에게 즉시 통지해 주는 시스템에 대하여 연구하였다.

본 논문에서 구현한 웹 컨텐츠 변경 모니터링 시스템의 기능만으로도 대부분의 웹 사이트들을 모니터링할 수 있고 인터넷에 접속 가능한 컴퓨터를 가진 사용자에게 변경 내역을 즉시 통보할 수 있다.

본 논문에서 구현한 시스템은 일반 사용자를 대상으로 개발되었기 때문에 미리 구성된 화면에서 제한된 모니터링 조건만 지정할 수 있어, 복잡한 모니터링 조건을 필요로 하는 전문가를 위한 모니터링 질의어를 지원하는 것이 향후 연구 과제이다.

참 고 문 헌

[1] Altavista. http://www.altaista.com/.

[2] Greory Cobena, Serge Abiteboul, Amelie Marian, "Detecting Changes in XML Documents", Verso Report number 194, 2001

[3] Laurent Mignet, Mihai Preda, Serge Abiteboul, Bernd Amann, Amelie Marian," Acquisition and Maintenance of XML Data from the We b", Verso Report number 188, 2001

[4] HTML Tidy. http://www.w3.org/People/Raggett/tidy/.

[5] World Wide Web Consortium. HyperText Markup Language(HTML) 4.1. http://www.w3.org/Markup/.

[6] 김태훈, 최중민, "사용자 편의의 인터넷 정보검색을 위한 지능형 웹 브라우징 에이전트", 정보과학회논문지(B) 제25권 제7호,1998

[7] Mind-it, http://www.netmind.com/index.shtml.

[8] TRACERLOCK, http://www.tracerlock.com/.

[9] TOMCAT, http://jakarta.apache.org/tomcat/index.html.

[10] SERVLET, http://java.sun.com/products/servlet/index.html.

[11] MySQL, http://www.mysql.com/.

[12] Red Hat Linux 7.2, http://www.redhat.com/.

저 자 소 개

김원중(Won-Jung Kim)

1987년 전남대학교 계산통계학과(이학사)

1989년 전남대학교 대학원 전산통계학과(이학석사)

1991년 전남대학교 대학원 전산통계학과(이학박사)

1999년~2000년 Iowa State University 교환교수

1992년~현재 순천대학교 정보통신공학부 부교수

※관심분야 : 소프트웨어공학, 시스템 모델링, 객체지향 시스템, 인터넷 서비스

초이기(Lee-Gi Cho)

1998년 순천대학교 전자계산학과(이학석사)

2000년 순천대학교대학원 컴퓨터과학과(박사수료)

※관심분야 : 데이터웨어하우스, XML응용, 인터넷 서비스

손철수(Cheol-Su Son)

2002년 순천대학교 전자계산학과(이학석사)

2002년 순천대학교대학원 컴퓨터과학과(박사과정)

1994~2002 (주)포스데이타 근무

※관심분야 : 소프트웨어공학, 네트워크, XML응용