ABSTRACT

Web change Monitoring and Tracking tools are automated tools that monitors changes to a web page or website. Website tracking refers to the act of archiving existing websites and tracking changes to the website over time. Website monitoring allows interested parties to track the health of a website or web application. A software program can periodically check to see if a website is down, if there are broken links or if errors have occurred on particular pages. Website tracking refers to the act of archiving existing websites and tracking changes to the website over time. There are many applications to website tracking which can be applied to many different businesses. This Paper is an overview of various types of Web Change Monitoring and Tracking tools, the techniques and algorithms used by them are studied.

Keywords: Web change Monitoring and Tracking tools, Information Extraction, Intelligent crawling.

1. INTRODUCTION

Web surfing, reading, blogging, trading, publishing and many others web activities have become a part of our daily life, after a decade of information booming letting many individuals and organization have their own Web site. In order to stay ahead, some corporations has fascinating web site for information servicing, such as announcing company’s latest movements or make a place for easy information exchanging. Hence, many new web pages will appear at the same time and many others will be updated frequently. As a result, there are many information portals constantly providing dynamically changing “fresh” information on the Web. This fresh information can be valuable for the users, so there is need for the users to know this newest information as soon as the change happened, by checking the particular web page constantly and frequently. This kind of activity of checking the web page for changes is called “Web Monitoring”. However, monitoring web pages manually is not efficient. This is because human tempts to make errors and may forget or miss important changes, especially when checking pages manually is rather impossible. Some intelligent Web monitoring agents or software robots, Web monitoring tools [Acke97, Dedi98, Doug98, Fish97, Glan01, NetM, Spy, WebS] have played an important role since the early days of the WWW.

2. MONITORING AND TRACKING TOOLS

A few Web monitoring and Tracking tools are discussed below. The first tool namely ChangeDetector is AI based machine learning system that reports changes efficiently. Similarly, the second tool ChangeDetect sends mail for identification of changes in webpage, TopBlend notifies the change by comparison, WebSpector time based monitoring, Webbeholder services through Agents etc.

2.1 ChangeDetector

ChangeDetector [boyapati02] is a system not only tracks the changes but designed to monitor a whole website. Whenever changes happen in a website, the ChangeDetector detects and reports to the user. In this way, it can inform changes which is difficult for user to find manually. For example company group organization restructuring or the end of some production lines can be reported by ChangeDetector.

![Change Detector Architecture](image)
and categorization techniques. ChangeDetector would automatically generate a sitemap, which is useful for intelligent crawling, so that it would give priorities to some important pages and crawl the website more efficiently next time. Therefore, ChangeDetector could monitor the whole website for changes and report it to the users via email or web browser. Figure 2 shows ChangeDetector detecting changes in the site.

2.2 ChangeDetect

ChangeDetect is a free but efficient web change monitoring service. It is a replacement of a few earlier commercial tracking tools which are Netmind, Mindit and Syponit. It saves user’s favorite web pages, monitors content for changes and sends an automatic email notification to the user whenever the web pages are updated. What has actually changed on web page text will be marked with color-coded highlights for easy understanding. Web page change notifications are delivered to the user via email, ICQ, text message or even pager. ChangeDetect provides a registration “shortcut”, a small utility script in the form of a browser’s toolbar icon or a bookmark in Favorites Menu which when clicked it will register the page being surfed automatically. This is convenient because the user has no longer needed to visit the server’s homepage and register the page to be monitored by keying in its URL. This is one of the good features that no other monitoring tools support. Besides, ChangeDetect allows user to set the notification trigger, by specifying some keywords combination with AND/OR rules. This allows user to decide whether to be notified when the changes on the monitored pages comprise or without a certain phrase. The figure below shows how to request for any website changes in ChangeDetection.com.

2.3 TopBlend

TopBlend [chen00] is a HTML differencing tool, which can detect the changes happened on a page by doing page comparison by HCS (Heaviest Common Subsequence). TopBlend has two major components, a HTML parser and an application of the Jacobson-Vo algorithm for the HCS (Heaviest Increasing/Common Subsequence) problem [Jacobson92]. The parser breaks down each HTML file into a set of blocks according to HTML tags and emits a token sequence. The HCS algorithm compares the two token sequences generated by the parser and outputs the difference information in either a single merged HTML file or a frame view for side-by-side comparison.

TopBlend was implemented in Java and used the fast Jacobson-Vo [jacobson92] algorithm to solve the HCS problem for page changes comparison. One of the merits of TopBlend is it allows comparisons to be performed either on the server or client side, which could offload busy servers duty in performing heavy computations. TopBlend has evolved from and been integrated with the AT&T Internet Difference Engine (AIDE) [douglis98] from AT&T research lab. AIDE adapted the LCS (Longest Common Subsequence) [hirschberg77], which is used in the well-known Unix diff program. TopBlend can be considered a re-implementation of AIDE after introducing Java applet, Jacobson-Vo algorithm and side-by-side comparison display methodology. The heart of the TopBlend system is a Java implementation of the Jacobson-Vo algorithm that solves a generalization of the longest common subsequence (LCS) problem. The line-matching algorithm of the well-known Unix command diff is based on computing an LCS between the sequences of lines of text of two given files. Unmatched lines are insertions and deletions. Since LCS is based only on matching symbols, the diff’s line matching algorithm cannot take advantage of line lengths or approximately matched lines when determining line insertions and deletions. [Jacobson92] discussed how to include weights when matching symbols. In the most general case, weights can be associated to both symbols and their locations in the
sequences being matched. Therefore, the Jacobson-Vo algorithm is capable of solving the heaviest common subsequence (HCS) problem. In TopBlend, there are three stages in applying the HCS algorithm. At every stage, the checksum of each token (computed from its string content), rather than the string itself, is used for comparison to improve performance.

2.4 Webspector

Webspector [webspector] is another tool useful for tracking web pages for content changes. Different from other server-based monitoring tools, Webspector is a client application that can be installed in user’s personal computer that allows unlimited number of web pages monitoring depends on the computational power of the PC. Besides being able to send notification emails, Webspector also provides an interface to registering web pages URL and displaying the changes results. In addition, with Webspector user can set the time schedule for the checking task to be executed, and also have the changes or special keywords highlighted for intuitive visualization. The figure below depicts the output display of Webspector.

2.5 WebBeholder

WebBeholder [santi98] is a multi agent web monitoring system developed in Ishizuka-lab from Tokyo University. System structure of WebBeholder is shown below. It consists of Service Provider Agent, a number of mobile agents and Mediator. The Service Provider Agent is responsible for monitoring the changes on the Web, while the mobile personal agent will be useful for customizing user requests. Next, Mediator will be responsible for dealing and negotiating the services between mobile agents and Service Provider.

Mediator contains three service modules. They are Request Broker, Navigator and Facilitator. The Request Broker is needed to convey the service requests from mobile agents to the Service Provider, at the same time coordinating the common requests among the mobile agents. Then, the task of the Navigator is to deliver the information about other WebBeholder communities to the mobile agents. Finally, the Facilitator is purposed to provide various capabilities to the newly added mobile agents. WebBeholder bases on LOCTAGS (Longest Common Tag Sequence) algorithm to retrieve the changes on HTML pages. Different tags in the changes will have different weights depends on their importance, say a change in the URL will gain more points than a change on some text. When total points of changes on a page exceed the trigger threshold, the user will be notified.

2.6 WebCQ

WebCQ [webcq] is Web information change detecting and delivering system developed in Georgia Tech. It is capable of monitoring and tracking the changes happening on both the static and dynamic web pages. Type of changes on web pages is classified in sentinels and this possible the more granular detecting and result
presentation. Therefore, this system can deliver user with personalized change results, because the users are allowed to decide the kind of sentinels for monitoring. WebCQ explores some object extraction algorithms to locate and identify the object of interests on web pages. Algorithms used can detect changes to arbitrary objects and compute how a page has changed. In addition, proxy cache service is provided to minimize access latency, reduce the workload of remote information servers, and achieve higher robustness of its change monitoring service. Lastly, Notification service is offered with a set of enhanced capabilities to provide both server-initiated push delivery and client-initiated pull delivery of changes, at the same time presenting rich and pleasant display to the user.

2.7 Infosphere Project

Infosphere project is part of DARPA Information Technology Expedition program. This project has been developed at Georgia Tech led by Calton Pu. Infosphere project focuses on bringing fresh information from a variety of sensors to the user as personalized fresh information delivery. They develop concepts, techniques and tools to support end-to-end quality of service (QoS), in terms of freshness, performance, availability and maintainability. Part of the concepts and techniques were implemented in the WebCQ introduced the previous session. In contrast to the well known of remote procedure recall (RPC), the concept of Infopipe was proposed. In addition to their basic function of transporting information, Infopipes can control the delivery properties of information such as freshness and performance mentioned above. They call the software architecture implementing this information flows in Infopipes as producer/consumer architecture, in contrast to the traditional client/server architecture. Information generated by a producer is carried to the consumers by Infopipes.

3. CONCLUSION

It is concluded that the automatic tracking changes in web pages help the user to keep up to date dynamic content. With the advent of the mobile Web, monitoring tools have found another niche: notification via mobile phone or PDA of changes in stock prices, weather alerts, News about specific companies or markets, the availability of a product, and much more. Some tools are also able to selectively monitor sections of a page, changes that include certain keywords, and structured changes.

REFERENCES

[18] [webcq] http://www.cc.gatech.edu/projects/disl/WebCQ/