Web Assistant

What

Web assistant is a system which provides digested content and content based alerts to users. To explain, lets consider an average internet user: He/she uses internet for a certain amount of time, and usually does what's expected from him/her: browsing the web. Why do we browse the web? We certainly have different jobs, interests, hoobies etc. And we act accordingly. Some of us, usually check 5-10 websites everyday, to check whether a news item appeared on them, and if something attracts our attention, we check the details about the item.

These sites usually include news sites like news.com, msnbc.com, specific news sites like theregister.co.uk, slashdot.org, mozillazine.org, linuxtoday.com; security alert and bug news sites like net-security.org; forum sites like catoftheday.com, tomshardware.com or anandtech.com; weather forecast sites whether next day will be rainy or not and even peoples blogs. People keeps checking these sites several times everyday, to see whether something new has happened.

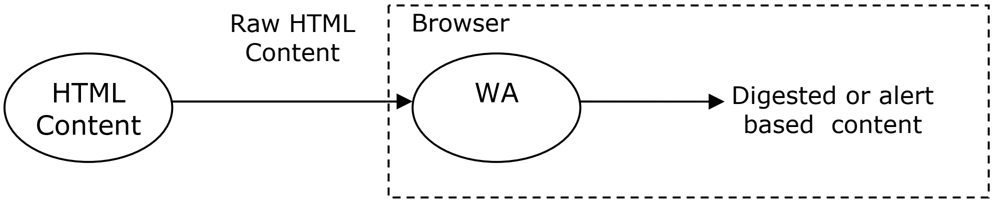
Some sites provide on-line news alerts (msnbc.com, excite.com, yahoo.com) some sites provide mail notifications on a daily or weekly basis. There are some ticker programs (infogate, kticker) which read information from content URLs of sites, generally in some form of XML or RDF.

Web Assistant provides a general solution for an obvious need of internet users: They need to know when a web site is updated; whether the update is of interest to them and what is the content update, without continuously checking those sites and wasting time.

The WA is therefore acting like an **intelligent bot** , which automatically checks sites of your interest, and informs you about the content update, and if necessary, stores the data for future use.

How ?

There are several technical issues that must be resolved before implementing this system. Systems outline can be seen below.



Here, HTML Content means sites with dynamic content. WA checks, filters and produces appropriate digest from HTML ocean.

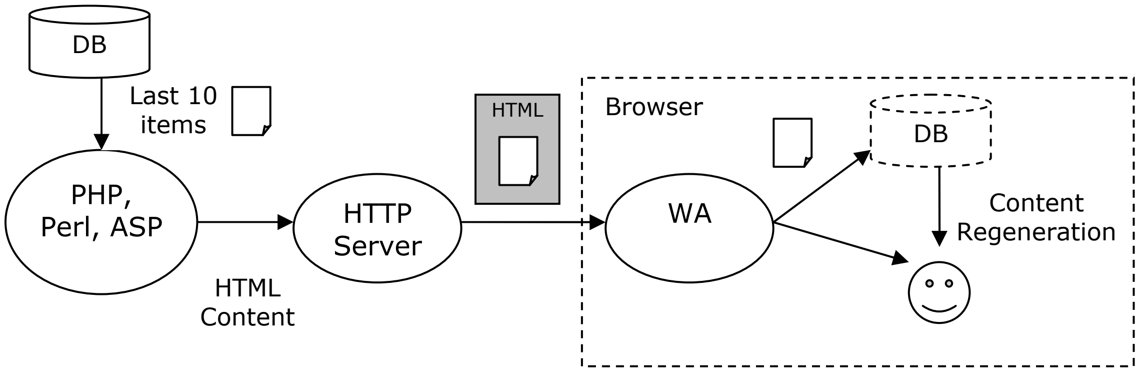
One can tell, that there are now services for this kind of "intelligent content proxy systems" (myYahoo and Microsoft's attempts) But what WA offer is, complete independence for such services!.

The system relies on the fact that all of the sites which have some form of dynamic content , are produced using;

- A selected set of records from a database.
- A server side script to make queries and produce html output. (like ASP, PHP, Perl CGI, JSP, PL-SQL, Servlet, Cold Fusion ec)
- HTML elements to make raw data presentable.

These dynamical content changes every time the database is updated, but other parts (HTML) of page do not change at all. They only change when a general design change occurs, which is very very unlikely for big sites.

And this means, **it is almost always possible to generate the data which reside in a database, from the HTML content that is presented to the user**. Unlike XML, HTML is supposed to be only human readable, but by using some methods, it can be automatically read and reformatted by an application, like WA.

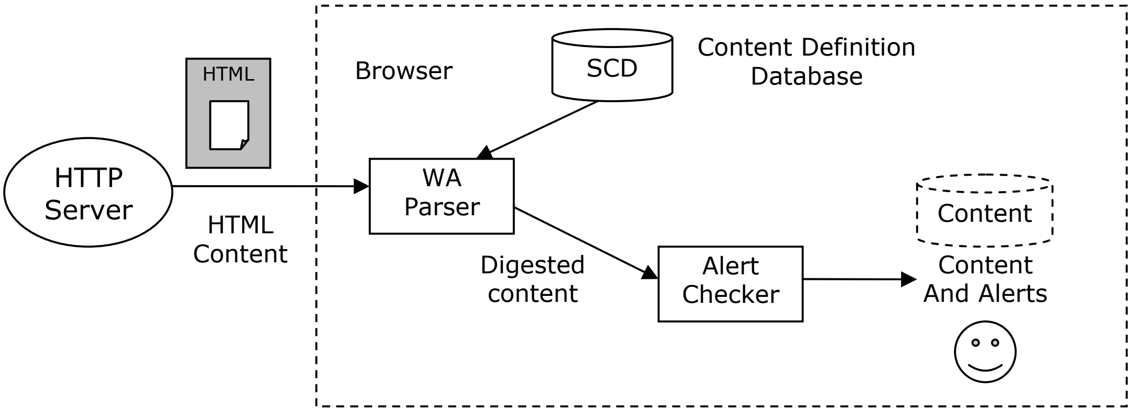


Therefore, the system can check sites with dynamic content, inform the user if there are new items, alert the user, if something important to him has happened. (according to the alerts and filters which user defined previously) Without visiting those sites, seeing boring advertisements all the time you check them.

How to extract content from HTML?

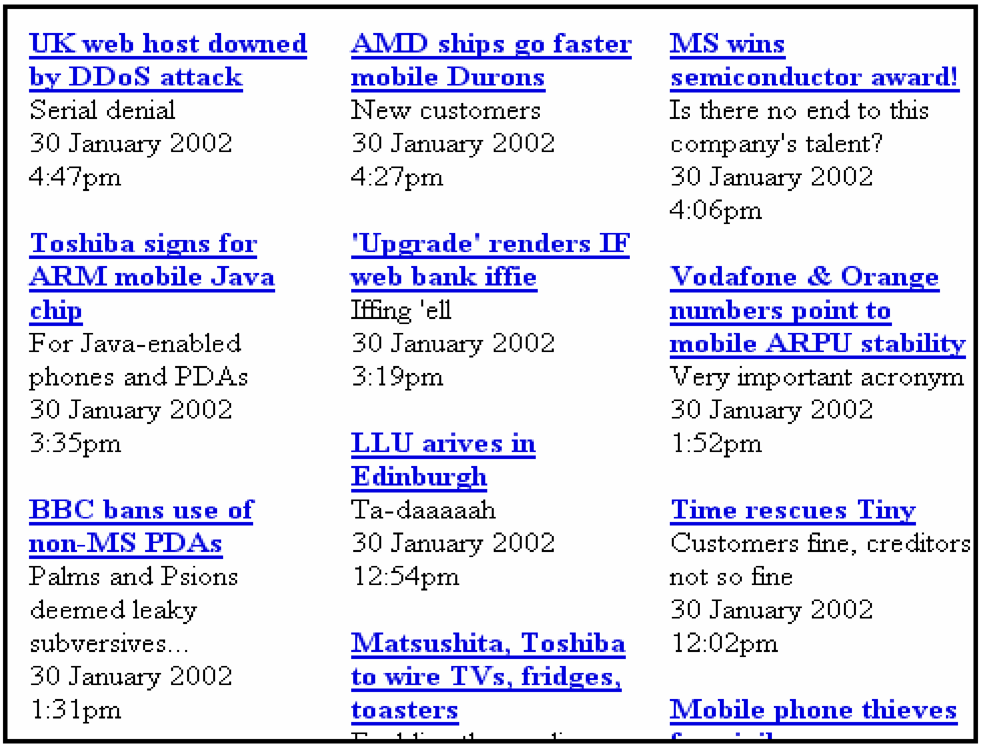
One of the key points of WA is that, it analyses the HTML document and extracts the content fragments according to some template matching mechanism. Of course WA cannot know what's meaningful and what's not,

by just checking the HTML code, it must be taught to do so, and this is done by using **Site Content Definition Files (SCD)**.
The idea is simple: We first analyse the html code of the site, and produce a file which defines the content fragment locations on the HTML code. And use this SCD files to parse the HTML data.



The MWA parser is actually an intelligent pattern searcher. Content definition files consist of logical structure of data items in the page (usually in tree form) and fragments of HTML code which signs the beginnings and the ends of the real content. Parser first reads HTML code and content definition, then searches for the HTML code fragments on HTML page and generates the content tree.

Let us consider a dynamic news page (www.theregister.co.uk), and examine its code to proove that the reconstruction of data tree is possible using techniques like pattern searching.



UK web host downed by DDoS attack
Serial denial
30 January 2002
4:47pm

Toshiba signs for ARM mobile Java chip
For Java-enabled phones and PDAs
30 January 2002
3:35pm

BBC bans use of non-MS PDAs
Palms and Psions deemed leaky subversives...
30 January 2002
1:31pm

AMD ships go faster mobile Durons
New customers
30 January 2002
4:27pm

'Upgrade' renders IF web bank iffie
Iffing 'ell
30 January 2002
3:19pm

LLU arives in Edinburgh
Ta-daaaaah
30 January 2002
12:54pm

Matsushita, Toshiba to wire TVs, fridges, toasters

MS wins semiconductor award!
Is there no end to this company's talent?
30 January 2002
4:06pm

Vodafone & Orange numbers point to mobile ARPU stability
Very important acronym
30 January 2002
1:52pm

Time rescues Tiny
Customers fine, creditors not so fine
30 January 2002
12:02pm

Mobile phone thieves

```
....
....
</a></noscript></td></tr></table> <hr></td></tr> <tr><td WIDTH="160"
VALIGN="top">

<div><div CLASS="indexheadlink"><a HREF="/content/55/23888.html"><strong>UK
web host downed by DDoS attack</strong></a></div> <div
CLASS="indexintro">Serial denial</div> <div CLASS="indexposted">30 January
2002 4:47pm</div> <br>

</div><div><div CLASS="indexheadlink"><a HREF="/content/3/23885.html">
<strong>Toshiba signs for ARM mobile Java chip</strong></a></div> <div
CLASS="indexintro">For Java-enabled phones and PDAs</div> <div
CLASS="indexposted">30 January 2002 3:35pm</div> <br>
....
....
```

As can be seen, every news item in the page has four parts,

- A Link (started with a <div><div CLASS="indexheadlink"><a HREF="/ and end with a "><strong>
- A Title (started with a "><strong> and end with a </strong></a></div>
- A Brief (started with a <div CLASS="indexintro"> and ends with a </div>
- A Date-Time (started with a <div CLASS="indexposted"> and ends with a </div> <br>

Obviously these data can be extracted using certain parsing techniques. There are several difficulties in parsing HTML content using string fragments and a content template, but all of them are solvable. The system can also detect a general design change in the site and warn the user.

The SCF file for a site also contains information about site, category of site and average update interval. But its main goal is to define the content locations in the HTML jungle.
A content defining part of SCF file might look like :

```
<Content>
 <Content_Set ID="news" type="set">
  <SP><![CDATA[<div><div CLASS= ]]></SP>
  <EP><![CDATA[ </HTML> ]]></EP>
  <Content_Bundle ID="news_item" type="bundle">
     <Content_Item ID="title" type="string">
      <SP>...</SP>
      <EP>...</EP>
     </Content_Item>

     <Content_Item ID="link" type="string">
      <SP>...</SP>
      <EP>...</EP>
     </Content_Item>

     <Content_Item ID="brief" type="string">
      <SP>...</SP>
      <EP>...</EP>

     </Content_Item>
  </Content_Bundle>
 <Content_Set>
</Content>
```

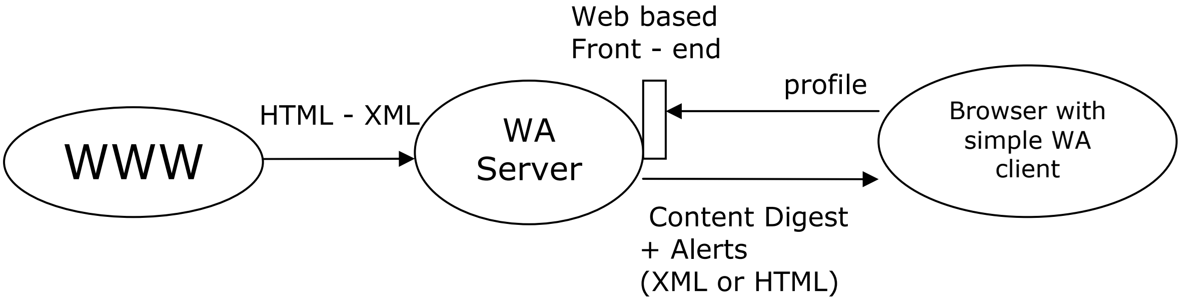
SP and EP means Starting and Ending patterns respectively. This is just an example, not a real representation of SCD file structure. Starting and Ending patterns must be flexible to catch some exceptions. Alternating patterns for one content fragment, or wildcards in patterns may help.

Web Assistant, as an "Intelligent Content Proxy" or a "Service"

There are some shortcomings of the first "Browser Embedded Web Assistant" idea. These are:

1. WA is bandwith hungry, if you are checking 20-30 site every 10 minutes , it may eat all modem bandwith
2. WA checks web sites only while you are on-line, you may miss something while you are off-line for a long time
3. WA is also CPU and memory hungry, on slow systems it can be a burden for user.

If we transform WA into a server, lets call "MozillaWA.org"



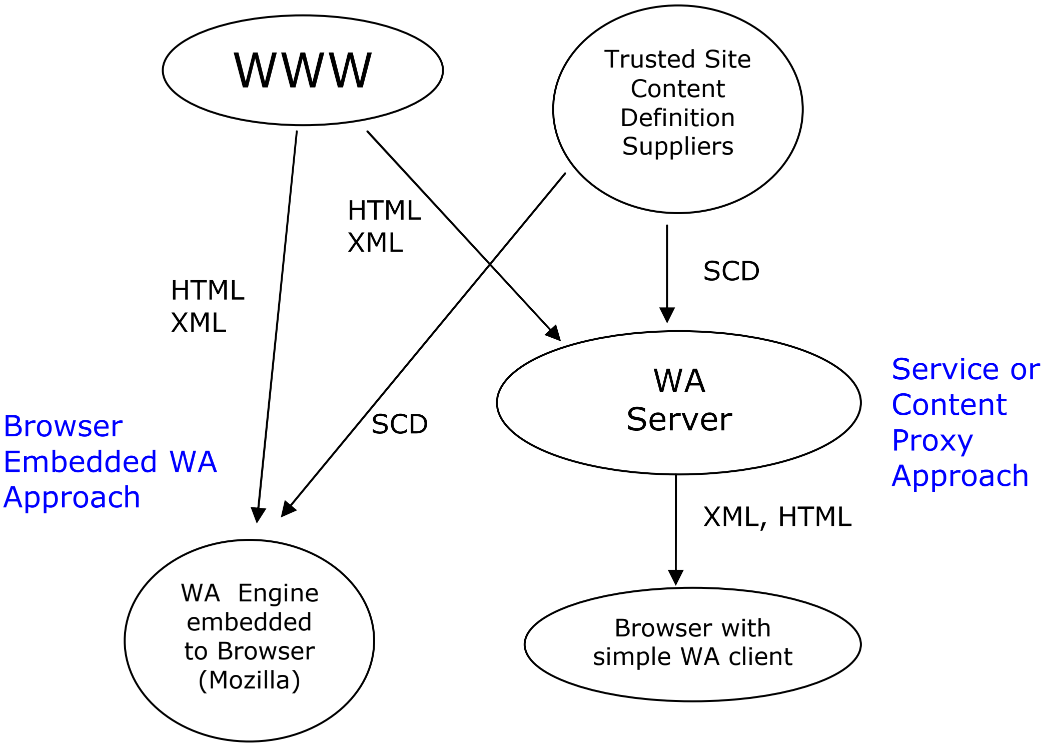
In this approach, WA server checks for the content updates and updates its own database acoordingly. Meanwhile checks new content for users, predefined alerts are checked server side, and WA clients on the user side (embedded to browser) just connects the WA server, and receives the content and alerts in an appropriate format (possibly RDF or XML)
To use the service, users first register to WA server and define their profile and alerts on sites which WA server checks continuously.
New SCF files can be added to WA server according to needs or request of users. In this approach, server must have an enermous bandwith and processing power , while client requires far less bandwith as compared to first solution. The implementation of such a system is more difficult and complex, it may require a distributed processing schema, a huge data storage, complicated load balancing and easy to use and powerful web interface.

This kind of server can also be placed on big companie's internet connection point as an intelligent content proxy to provide better internet service and probably less average internet usage time per worker.

Who will create and distribute SCD Files

SCD files can be difficult to write, HTML documents should be examined and content tree, starting and ending string patterns should be defined etc. It's not expected for an average user to create such a complex file.

The problem can be solved. First, a SCF editor can be implemented. In this editor, SCD File creator marks the content from a Mozilla's DOM inspector like interface, and picks the start-end patterns. Editor may also allows him to test the resulting SCD File. Using either the SCD File editor or a simple test editor any people can create these files which represents the characteristics and content of any dynamically updated site. These files can be distributed via trusted channels just like Netscape's sidebar directory for Mozilla sidebars.


WWW
Trusted Site Content Definition Suppliers
HTML XML
HTML XML
SCD
SCD
WA Server
Service or Content Proxy Approach
Browser Embedded WA Approach
XML, HTML
WA Engine embedded to Browser (Mozilla)
Browser with simple WA client

Where to Use, Highlights?

- For everyday use, as a news alert, a tool for gathering information of your interest.
- Web masters, administrators can be warned instantly against security alerts, new patches, new versions of programs.
- You can configure it to check if your favorite application has a new version, you may check prices of a particular product.

- Weather forecast alerts can be very useful. You define a “below zero” “fog” or “storm” alert for a meteorology site, and you are warned without visiting the site.
- You can even see whether your trolling attempt in a web forum had any response.
- You can check your favorite blogs, whether he/she added something new, without surfing the net.

Future

Content avareness features can be extended further. For example there can be mechanisms to allow defining operations on digested content, like if this value decreases more than %20 then alert me, or defining regular expressions on content.

Problems

- Implementation of such a project requires skillful designers and programmers. It may take 4-6 months to come up with a prototype.
- There can be some legal problems about using a web assistant service as a content proxy or open web service.

Programs Employing Similar Ideas

There are programs, using similar techniques to extract data from web pages. But none of them presents a gereral and widely applicable method like WA. Some of this programs are:

- Meta search engines: Copernic,Web Ferret etc.
- Personalised portal approaches: myYahoo. This time you are limited by what yahoo offers to you. Although yahoo offers more than 100 partly configureable content resources, they dont supply alerts and there are ads on each page. Microsoft’s myMSN is similar to Yahoo but content is even more limited.
- News tickers: Infogate, MSNBC news alerter, Excite.
- Smart Tags: Microsoft’s Smart tags technology allows programs to embed values from HTML pages to applications.
- Specialised search engines: Sherlock for Mac.
- Info bots: Price watchers, Program version trackers etc. There are also programs which reports when a particular page is updated. But they dont supply information about content. Some of these programs are listed in bots.org

But WA approach is generalised and can transform into all systems mentioned above.

Conclusion

Web Assistant can be a very powerful idea for a web browser, which is the obvious selection for such an application, since its completely about making browsing easy and effective. People want only that from browser developers;

make their life easier as an internet user. It can be the next big thing in the browser arena.
If it is also backed by a poverful content proxy server mentioned before, this system would be extremely helpful to internet users.

Mehmet Dundar Akin, Sn. Researcher
TUBITAK - UEKAE
Gebze - Turkey
mdakin@uekae.tubitak.gov.tr