



Web Tuner: Easier, personalized web browsing

Krishnan Ramanathan, Yogesh Sankarasubramaniam, Vidhya Govindaraju

HP Laboratories
HPL-2011-177

Keyword(s):

Web browsing; Personalization; Document Summarization

Abstract:

Lack of good text input mechanisms, difficulty in discovering new websites and limited reading space make consuming the web from mobile web browsers difficult. We propose an application called "Web Tuner" that aims to make web browsing on mobile devices easier. Our application is based on the metaphor of tuning into radio stations. Users are recommended new websites based on their past browsing habits, and content is presented in a summarized form. Possible monetization models include paid website insertions and advertisements.

External Posting Date: October 6, 2011 [Fulltext]
Internal Posting Date: October 6, 2011 [Fulltext]

Approved for External Publication

Web Tuner: Easier, personalized web browsing

Krishnan Ramanathan, Yogesh Sankarasubramaniam, Vidhya Govindaraju,

HP Labs India

Abstract

Lack of good text input mechanisms, difficulty in discovering new websites and limited reading space make consuming the web from mobile web browsers difficult. We propose an application called “Web Tuner” that aims to make web browsing on mobile devices easier. Our application is based on the metaphor of tuning into radio stations. Users are recommended new websites based on their past browsing habits, and content is presented in a summarized form. Possible monetization models include paid website insertions and advertisements.

1 Introduction

Mobile devices capable of accessing the Internet (such as slates, table devices and smartphones) have seen tremendous growth in recent times. Mobile phones are becoming the on-ramp to the Internet for many emerging market users. The browser is the killer app for mobile devices; however, there are three problems that limit the usefulness of web browsing using mobile phones.

The first problem is the lack of good text input mechanisms. Mobile devices do not offer good interfaces for entering text (entering long *urls* is especially difficult). While one could use search to get to *urls* of known websites, it introduces an extra hop. Most users frequently access a limited set of websites, and it would be useful to have faster shortcuts to access them.

The second problem is content discovery. Search and browse doesn't work well on mobile devices; one study put the number of mobile queries at 1% of desktop queries. New (and sometimes experienced) users have little idea of which web site to go to, and do not have the skills (or inclination) to search and query the Internet from the mobile web browser. The fact that mobile devices do not offer good interfaces for entering *urls* makes this problem worse. Also, many users (e.g. those in emerging markets) are more comfortable with content delivery metaphors such as television, radio, magazines and newspapers where the content is curated and pushed to the user. Finally, there is no way for users to track/keep up with few topics of longstanding interest.

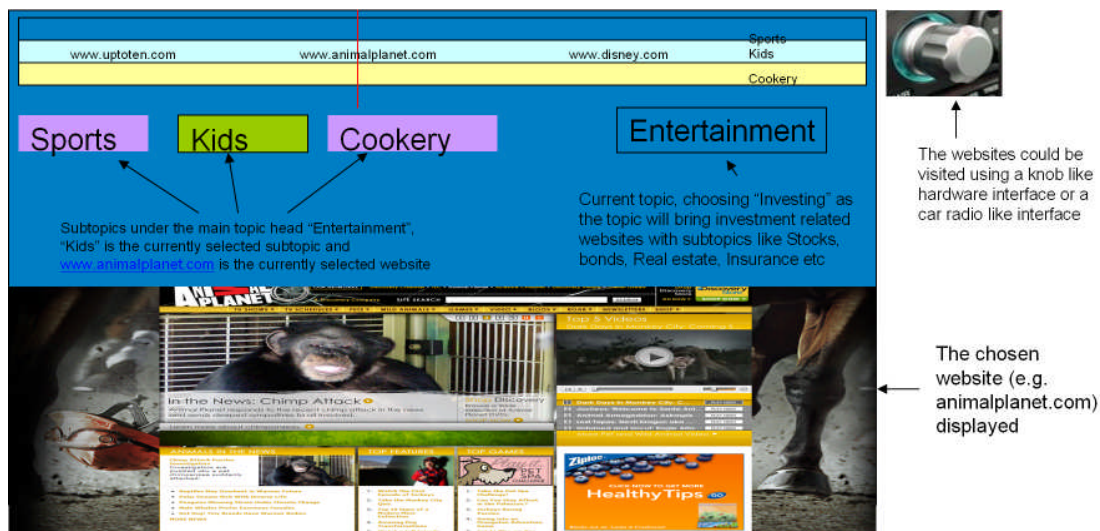


Figure 1 Web Tuner – Radio like interface for browsing websites

Finally, there is the problem of presenting content on a small screen, which is also partly responsible for the inefficiency of mobile search and browse. One has to either navigate manually through a web page or a large document, or read it in tiny font to save the effort.

We have tried to solve the above problems by creating an application called “Web Tuner”. Our application offers an interface that makes web consumption on small form-factor devices less painful by creating a local directory of frequently accessed sites. We have also tried to facilitate easier discovery of websites by providing websites similar to the ones the user is consuming through our interface. For easier and faster browsing, Web Tuner first presents a small summary and then increments it depending on the user’s interest.

The Web Tuner application is based on the metaphor of tuning into radio stations. On older radio sets, users turned a knob to navigate to their favorite stations. New radio stations appeared automatically and users tuned into them if they are interesting. We want to make the navigation and discovery of websites as simple as tuning into a radio station. Figure 1 depicts a possible UI including a knob like interface that could be used for tuning to websites.

2 System design

Figure 2 shows the architecture of the application. The application offers seven pre-loaded categories (News, Sports, Technology, Nature, Networking, Health and Video) and seven websites in each category. The user access data for both the categories and websites is stored in a SQLite database capability with HTML5 facade that HP’s WebOS provides. The personalization engine uses the access statistics to recommend new categories and websites to the user. The server side interface is used for recommendations, paid website inclusions and advertising).

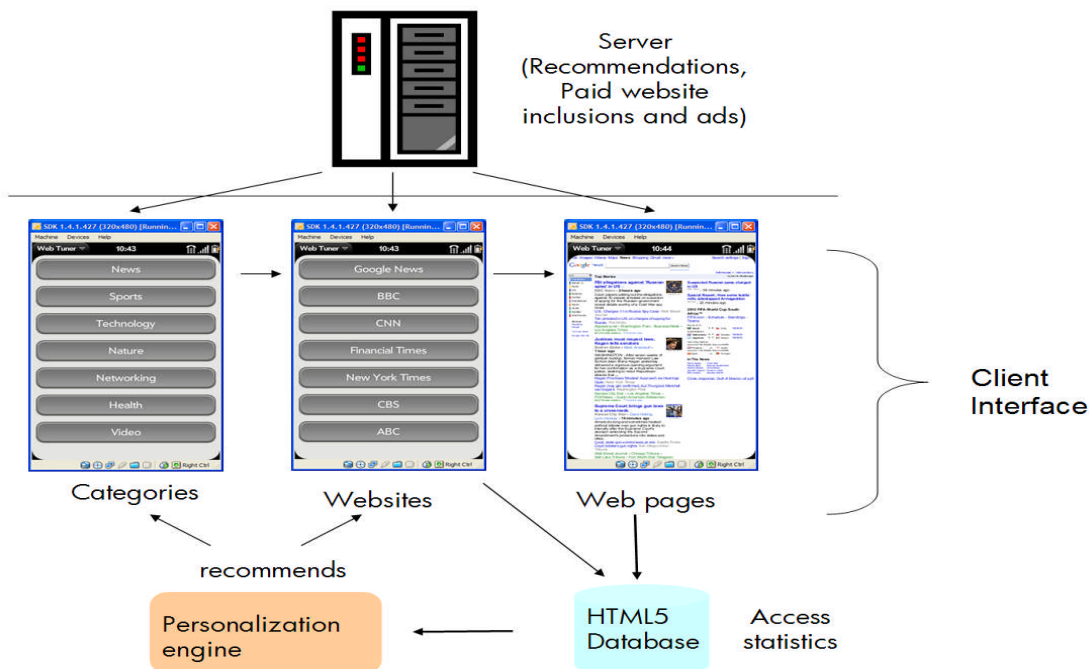


Figure 2 Web Tuner architecture and UI on a mobile phone (Palm Pre2)

2.1 Simplifying User Input

The application has a number of features to reduce or eliminate user input. These include: predefined categories and websites, content caching and a scan mode for rapidly scanning websites.

To simplify user input, Web Tuner allows the user to browse websites based on a predefined set of categories such as sports, politics, gadgets etc. (see figure 2). It also allows users to add custom categories. The category could be any topic of longstanding interest to the user. On the addition of a new category by the user, websites relevant to the category are automatically populated. This is done based on queries to a search engine but is transparent to the user. Our solution also allows users to add and delete new websites within existing categories. In order to reduce the latency of viewing websites, the app features a “quick scan” mode that allows the user to cycle through all the websites in a category. For instance, users can use this mode to have a quick glance at their facebook/twitter/gmail accounts in the networking category.

2.2 Location based websites

We have implemented a feature for recommending location specific websites. The user location is first obtained (using GPS, wi-fi or triangulation based methods). In the user interface, a location specific category is created. Clicking the location category brings up malls, restaurants, hospitals etc near the current location. The user can also search for other entities near the current location (figure 3).

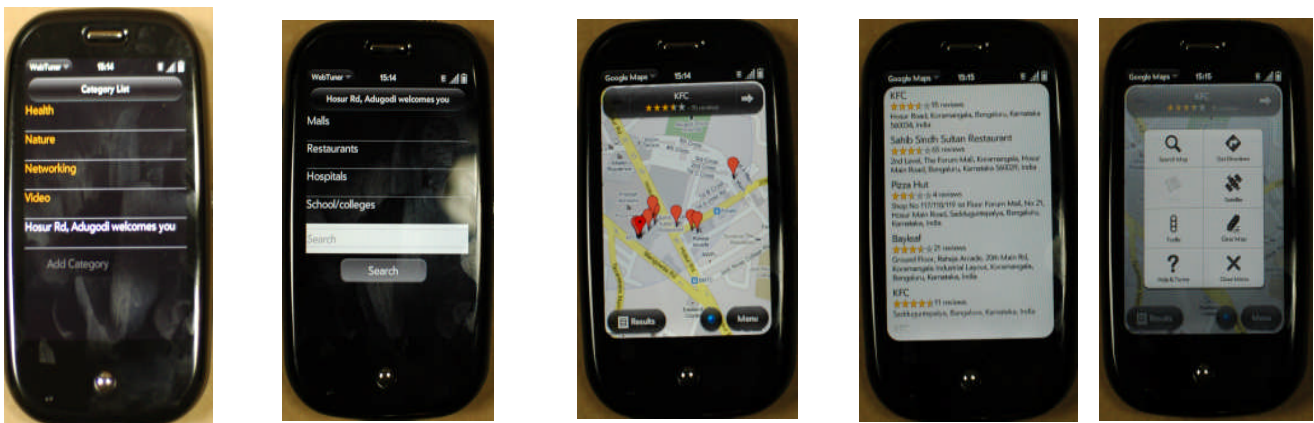


Figure 3 Location based category

2.3 Personalization and discovery

To facilitate the discovery of new websites, we have implemented a personalization engine. The personalization engine keeps track of the evolving user interests and recommends websites and categories to the user on an ongoing basis. The simplest form of personalization is re-ranking of websites. Websites that the user accesses most often are ranked higher and moved up in the UI.

We also recommend new websites and categories based on user browsing habits. If it is found that the user browses www.cricinfo.com very often and rarely visits the www.soccer.com website, the soccer website will be replaced by a new cricket website. In general, we replace rarely visited websites by sites

similar to the frequently visited ones.

2.4 Content presentation

We use our document summarizer [1] to present to the user a small summary (2-4 sentences) of a web page. The user can then make a decision whether to read the entire page, request a longer summary, or move on to the next page. By removing redundancy across multiple inputs, the summarizer can also be used to automatically present the gist of a category in Web Tuner.

2.5 Monetization models

There are two monetization models in Web Tuner. One model is based on paid categories and paid websites within a category. The other is banner advertisements. For both these models, we have implemented a server side interface.

In the paid categories model, advertisers pay for creating a category (and websites within the category). This can be done based on location or other demographic criteria. These categories and websites are pulled by the application at startup and displayed. In the paid websites model, advertisers bid to have their websites included in categories that users have already created.

There is a pre-existing category within the application called advertisements. Banner ads are pushed from the server and are displayed when a user browses this category within the application.

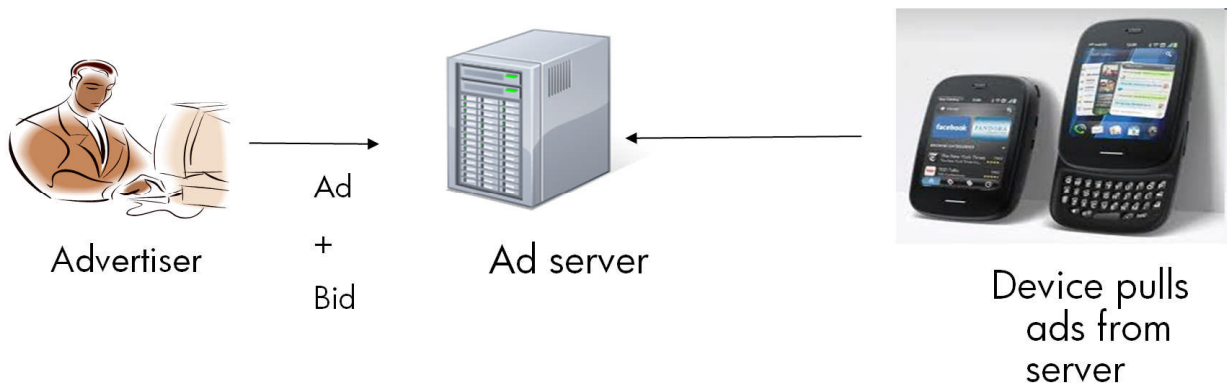


Figure 4 Advertisements in Web Tuner

3 Related work

There have been a number of efforts to solve the latency and discovery problems; we mention only a few we feel are most promising. Apple recently acquiring Siri, an application that acts like a personal assistant to which tasks like booking tickets could be delegated. Latency is tackled by using voice as the input method. Google Fastflip is an application that attempts to speed up browsing by allowing users to rapidly flip to interesting content. Fastflip also personalizes the experience by taking cues from selections to show more content from sources, topics and journalists that users like. However, the experience seems to be tailored at the desktop user; the mobile app displays pages in small fonts and expects users to pan. At a framework level, Sproutcore aims to eliminate latency by moving business logic to the client and running javascript apps in the browser.

The Kylo browser from HillCrest Labs offers an interface with pre-packaged websites, however there is no adaptation to the users' interests. The interface is tailored for large-screen devices (TV's) while ours is designed for small screen devices. Flipboard is an application that creates a personalized magazine for user's based on inputs from their social network (via twitter and Facebook). Another interesting effort is Instapaper where users are recommended interesting articles that they can save for later reading however Instapaper does not attempt to personalize the content to the user interests.

4 Next steps and conclusion

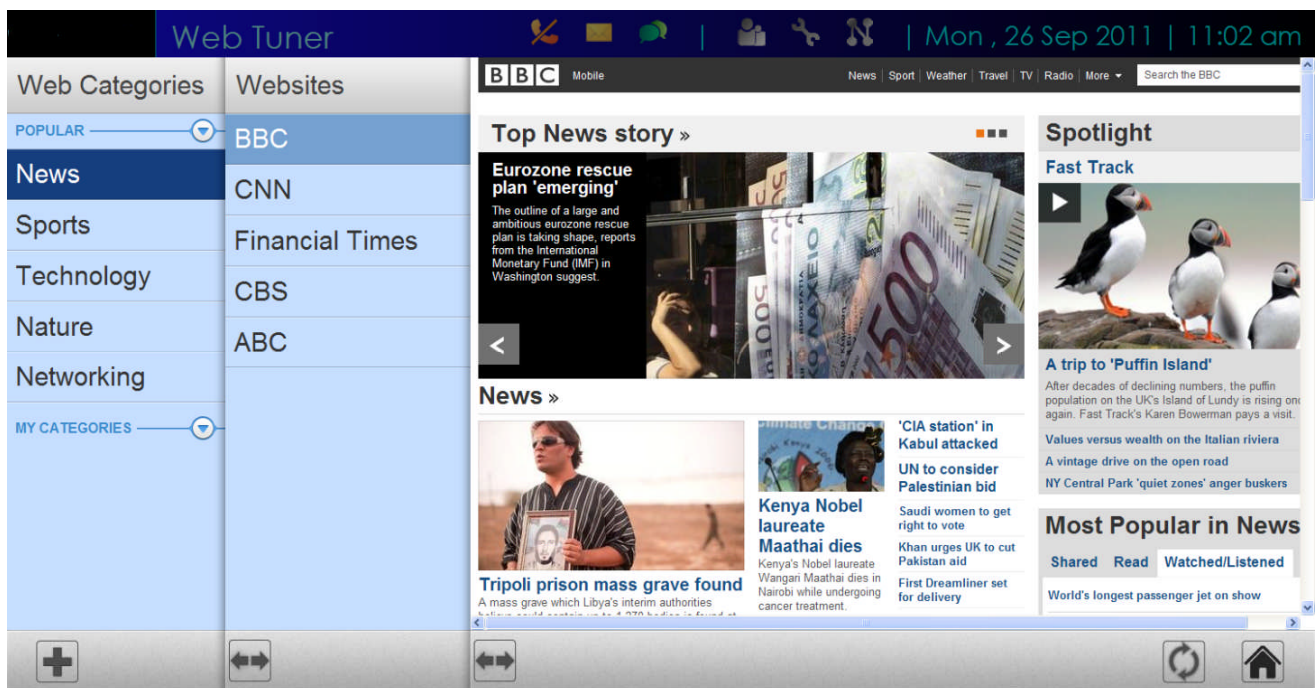


Figure 5 HP TouchPad version of Web Tuner

We have implemented the application on the Palm Web OS for both mobile phones and the Touchpad (using the WebOS Enyo framework). The interface for this is shown in figure 5. We plan to make subsequent versions of the app available users via the Palm appstore. We also plan to conduct an evaluation to assess how much the app improves web surfing (from the perspective of both ease of consumption and content relevance). A US patent application has been filed [2].

We plan to integrate some of the technologies we have developed for user profiling [3,4], similar content searching [5] into the application to create a more personalized experience. We plan to extend the application to automatically create categories based on context information such as the users' profile, users' recent web queries and current location. To further speed up category and website input, we plan to explore voice based input and paper based input. Finally, we plan to make the user interface closer to a radio-like UI.

Acknowledgements

We would like to thank the developers who worked on Web Tuner namely TS Kavitha, Gouri

Deshpande and Gali Prasad for their efforts.

References

1. Ramanathan, K.; Sankarasubramaniam, Y. (2010). Document summarization: An Information Perspective. HPL-2010-127. HP Labs India Technical Report.
2. Krishnan Ramanathan, Method and system for navigating websites, US Patent application 20110061026.
3. Krishnan Ramanathan and Komal Kapoor, Creating user profiles using Wikipedia, ER 2009 conference, LNCS Volume 5829/2009, 415-427, DOI: 10.1007/978-3-642-04840-1_31, Springer Verlag.
4. Sankarasubramaniam, Y.; Ramanathan, K.; Lahiri, A.; Govindaraju, V. (2010). Group recommendations using Joint Profiling. HPL-2010-126. HP Labs Technical Report.
5. Vidhya Govindaraju and Krishnan Ramanathan, Similar Document Search and Recommendation, HPL-2011-150, HP Labs Technical Report.