6.870 Final Project Proposal
Webnnel: A channel-based Web navigation system

Chen-Hsiang Yu and Oshani Seneviratne
{chyu,oshani}@mit.edu

1. Introduction

1.1 Motivation & Ideas

The Web has become an important medium for delivering information, and more and more people reply on it for work or entertainment. For example, users like to check e-mails, read news, watch videos, listen to music and do shopping on the Web. With the success of Web browsing on the PC environment, people are familiar with using the Web, and start to apply similar experience to different domains, such as mobile browsing and browsing on different Wi-Fi enabled devices. In this project, we envision an application for home environment. We plan to design a TV channel like Web navigation system. In this system, the user can use the speech to navigate the menu of web sites and control browsing behavior via the speech command. Figure 1 illustrates the idea we propose.
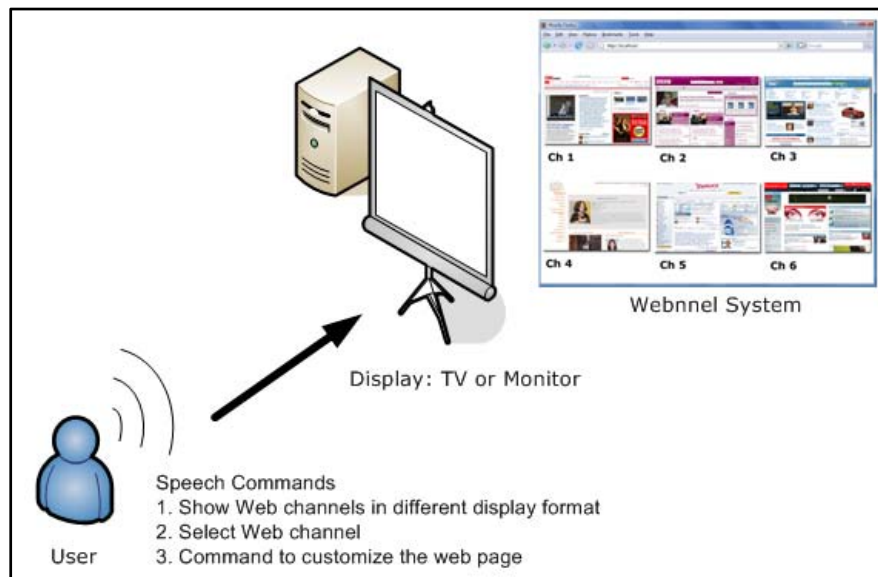

Figure 1: Expected scenario of Webnnel – A channel-based Web navigation system

1.2 Technology

In this project, we will investigate, design and integrate possible technologies to make Webnnel system feasible and easy to use for home users. The technologies we plan to investigate include Web content manipulation (XHTML / CSS / DOM / JavaScript), speech command recognition (Java), browser extension development (XUL and JavaScript), and user interface design.

1.3 Expected Result

We expect to have a channel-based Web navigation system that could be easily used at the home environment. In the human computer interaction part, we expect to provide speech command input for the users to control Web site selection displaying in different formats, Web navigation and content manipulation, such as removing all the images or enlarging the content.

2. Related Work

2.1 UI and Content Access

Information display with TV channel format can be seen on some applications. Youtube uses frame list and flash animation to display the video clips. Joost [5] and Mogulus [7] use grid-based arrangement to display live TVs clips with multiple small screens. Even in the mobile device, Avot mV [1] uses similar display to provide video search. The idea of TV channel format presenting web sites is inspired by these applications, because we think it could save users' time from typing in the URL address and provide more natural interaction and better Web browsing experience. However, to the best of our knowledge, we do not see similar system proposing an idea to display web sites as TV channels for the users to access frequently used web sites at the home environment.

To the Web content access, programmers can write JavaScript programs, which are embedded into HTML or XHTML web pages, to access web content dynamically. Chickenfoot [3] and Greasemonkey [4] are two web scripting framework to allow users to write scripts to customize the Web pages. Accessmonkey [2] is another script framework that allows multiple users, including web users, web developers and web researchers, to collaboratively write the scripts to enhance web page accessibility. Unfortunately, all of them do not provide interface for user to customize the web page by using natural interaction, such as speech or gesture.

2.2 Speech invoked content access

Speech invoked web content access is suitable for people with disabilities (especially people with dysfunctional hand motor-abilities), workers who need to access information in a hands-off manner to improve their productivity or simply the general user who wishes to have a much more natural interaction in accessing web pages with spoken commands. Microsoft Windows Vista Speech Recognition system [6] provides a platform for users to control Windows applications. This supports dictation of documents and emails in mainstream applications, voice commands to start and switch between applications, control the operating system, and even fill out forms on the Web. However, it does not have much flexibility in the Web browsing and Web content manipulation.

3. Plan of Implementation

The Webnnel system architecture contains four components: (1) Content Manipulation Module (CMM); (2) Channel Aggregation and Presentation (CAP); (3) Command Abstraction Interface (CAP); and (4) Speech Command Extraction (SCE). We plan to integrate these four parts as a Firefox browser extension to demonstrate our ideas. The whole system architecture is illustrated as Figure 2.
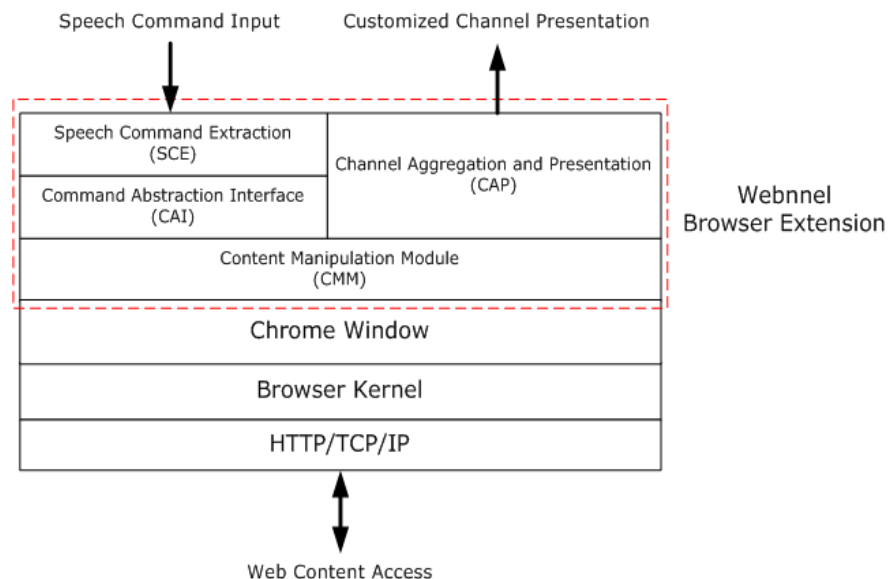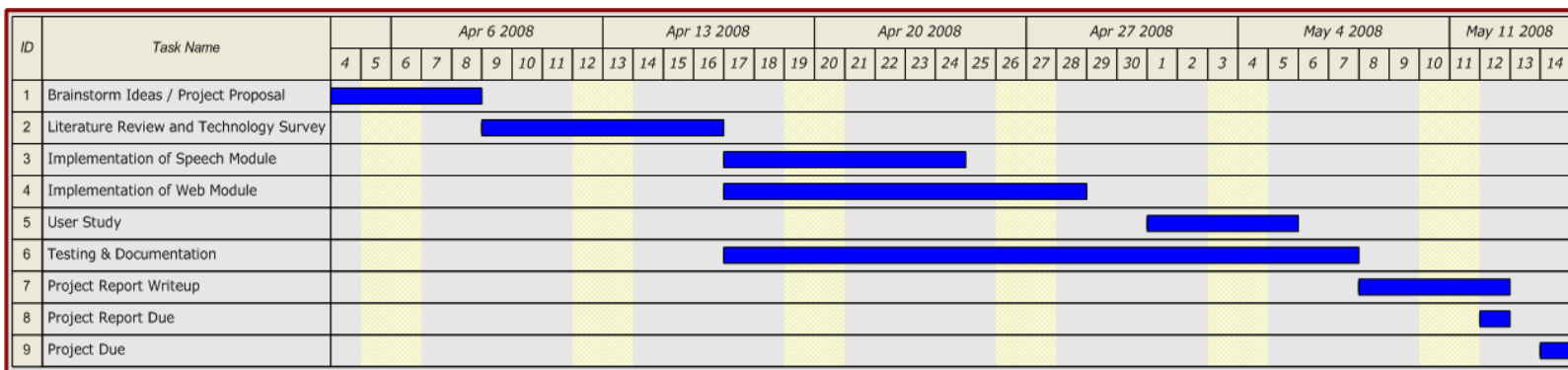

Figure 2: Webnnel system architecture

- Content Manipulation Module (CMM)
  - Define functions for specific purpose, such as images detection and content access, for CAI.
  - Define functions for CAP to render content, such as web site snapshots or displaying with different formats.

- Channel Aggregation and Presentation (CAP)
  - Rendering web site snapshots and providing appropriate UI supports.

- Command Abstraction Interface (CAI)
  - Define high level APIs for SCE to use to satisfy users' commands.

- Speech Command Extraction (SCE)
  - Speech recognition and command extraction.

4. Timeline

The following Gantt chart shows the tentative timeline we have allocated for this project.

ID	Task Name	Timeline (Apr 6 2008 – May 11 2008)
1	Brainstorm Ideas / Project Proposal	Apr 4 – Apr 8
2	Literature Review and Technology Survey	Apr 9 – Apr 14
3	Implementation of Speech Module	Apr 15 – Apr 25
4	Implementation of Web Module	Apr 16 – Apr 28
5	User Study	Apr 30 – May 5
6	Testing & Documentation	Apr 16 – May 8
7	Project Report Writeup	May 9 – May 13
8	Project Report Due	May 13
9	Project Due	May 14

5. Collaboration

The Webnnel project is collaborated by Chen-Hsiang Yu and Oshani Seneviratne. We divide our project into several tasks and each focuses on specific tasks mentioned below.

Chen-Hsiang Yu:
- Web content manipulation (CAI and CMM)
- UI design (CAP)
- Extension Development
- User Study
- Report Write-up

Oshani Seneviratne:
- Speech recognition and extraction (SCE)
- Extension Development
- User Study
- Report Write-up

Currently, we use Eclipse with SVN and Google Code online version control (http://code.google.com/p/webnnel/) to manage and synchronize our documents (such as this proposal), references, images and project source codes.

6. References

[1] Avot mV, http://www.avotmedia.com/

[2] Bigham, J. P., and Ladner, R. E. Accessmonkey: a collaborative scripting framework for web users and developers. *In W4A '07, ACM Press*, pp. 25-34, 2007.

[3] Bolin, M., Webber, M., Rha, P., Wilson, T. and Miller, R.C. Automation and customization of rendered web pages, *Proceedings of the 18th annual ACM symposium on User interface software and technology*, October 23-26, 2005.

[4] Greasemonkey, https://addons.mozilla.org/en-US/firefox/addon/748

[5] Joost, http://www.joost.com/

[6] Microsoft Windows Vista Speech Recognition system
http://www.microsoft.com/enable/products/windowsvista/speech.aspx

[7] Mogulus, http://www.mogulus.com/

[8] Petrie, H., Hamilton, F. and King, N. Tension, what tension? Website accessibility and visual design. *Proceedings of the 2004 international cross-disciplinaryworkshop on Web accessibility (W4A)*, pp. 13-18, 2004.

[9] Richards, J. and Hanson, V. Web accessibility: a broader view. *Proceedings of the 13th international conferenceon World Wide Web*, pp. 72-79, 2004.