Development Manual

Table of Contents

[Introduction 2](#_Toc165726603)

[Toolbar Architecture 2](#_Toc165726604)

[Basic Operations: Move and Engage 2](#_Toc165726605)

[Contexts and the Context Manager 2](#_Toc165726606)

[Surf Mode (Simple Mode) 2](#_Toc165726607)

[Technologies, Libraries, and References 2](#_Toc165726608)

[Prototype.js 2](#_Toc165726609)

[Essential Components 2](#_Toc165726610)

[Additional Components 3](#_Toc165726611)

[Using Preferences 3](#_Toc165726612)

[Developing and Extending the Hawking Toolbar 3](#_Toc165726613)

[Helpful Resources 3](#_Toc165726614)

[Meet the Developers 3](#_Toc165726615)

[John Foushee 3](#_Toc165726616)

[Andrew Hulbert 3](#_Toc165726617)

[Brian Louden 3](#_Toc165726618)

[XUL and Firefox Tutorials and References 3](#_Toc165726619)

# Introduction

The Hawking Toolbar Project's goal is to design an extension for Mozilla’s Firefox browser to enable users to view and interactively navigate the content of the web using a limited input interface. The target input interface for which the toolbar is designed is a one or two switch interface. The physical switch will be mapped to an event in the browser such as a key press or mouse click. These switches will allow the user to interact with the toolbar which in turn will allow them to view web content as well as browser functionality such as clicking links, scrolling, or interacting with the browser history.

This document contains an overview of the basic design and implementation of the toolbar. It begins with an overview of the architecture and a list of technologies and references used, then moves on to describe some of the basic components and how they were implemented, and finally finishes with documentation on how to extend the current toolbar to add additional functionality.

# Toolbar Architecture

## Basic Operations: Move and Engage

The Hawking Toolbar is designed around two basic input mechanisms: move and engage. These two mechanisms allow the user to access and interact with all of the functionality of the toolbar and thus content of the web page the user is visiting.

## Contexts and the Context Manager \*\*\*John comment

A Context within the Hawking Toolbar is an abstraction that is designed to present the user with a set of functionalities that can be accessed using the move and engage architecture. Conceptually, a Context is simply a list of functions that the user can iterate through using the Move input and then choose one of the functions from the list using the Engage input. A Context is implemented in the code as a ContextList

The Context Manager is a JavaScript class

## The Main Toolbar

The main toolbar serves as the main interface from which all other toolbar functionality can be accessed. It is simply a context that contains buttons which either directly activate functionality of the Hawking Toolbar or enter another context using the Context Manager to access additional functionality.

## Sub Toolbar Components

Families of functionalities are implemented on the toolbar via the creation of sub toolbars. For instance, the Navigation sub toolbar contains a family of functions (Forward, Back, Refresh) that are accessed via buttons on the Navigation sub toolbar. Sub toolbars are Contexts that can be passed to the Context Manager and thus are built upon the move and engage architecture.

# Technologies, Libraries, and References

## Prototype.js

The Hawking Toolbar makes use of the prototype.js library found at:

<http://prototype.conio.net/>

This library serves as a JavaScript Framework that eases development and allows easier interaction with the DOM. The main purpose of using the prototype framework is to access its ability to work in a more structured object like thought process than used with standard JavaScript style.

# Essential Components

## Highlighting

The main purpose of using the Hawking Toolbar is to navigate web pages by detecting and clicking links using input switches. The Highlighter class implemented in hawkinghighlight.js provides supports highlighting within a window in Firefox. The Highlighter class creates an HTML <div> element and then uses absolute positioning and component resizing to place this highlighter div

## Event Capturing

## Surf Mode (Single Input Mode)

Surf Mode or Simple Mode is a reduced functionality mode in which the toolbar only iterates through the links available in

## Sound Manager

# Additional Components

Navigation SubToolbar

# Using Preferences

# Developing and Extending the Hawking Toolbar

# Helpful Resources

## Meet the Developers

### John Foushee

John Foushee is a Class of 2008 computer science major at the University of North Carolina at Chapel Hill originally from XXX.

Email – John Foushee [jgf@email.unc.edu]

### Andrew Hulbert

Andrew Hulbert is a Class of 2008 computer science major at the University of North Carolina at Chapel Hill originally from Snow Hill, NC.

Email – Andrew Hulbert [hulbert@cs.unc.edu]

### Brian Louden

Brian Louden is a Class of 2009 computer science major at the University of North Carolina at Chapel Hill originally from Raleigh, NC.

Email - Brian Louden [blouden@email.unc.edu]

## XUL and Firefox Tutorials and References