TiddlyRSS: RSS reader in JavaScript

By
Wang Shangshang

Department of Computer Science
School of Computing
National University of Singapore
2007/08
Honours Year Project Report

TiddlyRSS: RSS reader in JavaScript

By

Wang Shangshang

Department of Computer Science
School of Computing
National University of Singapore
2007/08

Project No: H041170
Advisor: Dr. Martin Henz
Deliverables: Report 1 Volume
Source Code: 1 CD
Abstract

RSS - Really Simple Syndication has become the standard for publishing dynamically changing web content. RSS reader is typically an application that retrieves and displays RSS content to the client. However, the existing RSS readers are either desktop application on the client side that requires installation or browser application on the server side that relies on the access to the internet and involves privacy issue. The innovative RSS reader developed in this project - namely TiddlyRSS is designed to resolve the problems and enhance the strengths of the existing RSS readers. One of the major improvements is to support a browser application on the client side by applying the cutting-edge web technologies such as TiddlyWiki, TiddlyCard and AJAX. As a result, the TiddlyRSS is available offline and extremely portable, it requires no installation on client side. In addition, it is a pure JavaScript application which is able to integrate with any potential JavaScript web application. TiddlyRSS stands for two stages in this project: TiddlyWiki RSS and TiddlyCard RSS.

Subject Descriptors:  
D.2.1 Requirements/Specifications  
D.2.2 Design Tools and Techniques

Keywords:  
Software Engineering  
Tool Support

Implementation Software:  
Macromedia Dreamweaver as JavaScript editor,  
Internet Explorer, Mozilla Firefox, Windows XP
Acknowledgement

I would like to extend my heartfelt gratitude to my supervisor Dr. Martin Henz, without whom this thesis would hardly exist. His dedicated guidance, invaluable advice, continuous patience and support has inspired me and made this project possible.
List of Figures

2.1 RSS 2.0 file format ............................................................... 7
2.2 Tristana RSS reader screenshot .............................................. 15
2.3 Bloglines RSS reader screenshot ............................................ 15
2.4 Standard Edit dialog as Tiddler ............................................. 16
2.5 TiddlyCard single page display ............................................. 18
3.1 Relationship Chart .............................................................. 20
4.1 TiddlyRSS core working flow chart ........................................ 24
4.2 Feed channel tiddler for NY Times ....................................... 24
4.3 TiddlyWikiRSS screenshot .................................................... 24
4.4 TiddlyCardRSS screenshot ................................................... 24
5.1 Google RSS reader screenshot ............................................. 24
5.2 AgileRss reader screenshot .................................................. 24
# Table of Contents

Title ............................................. i
Abstract ....................................... ii
Acknowledgement ............................... iii
List of Figures ................................. iv

1 Introduction ................................... 1

2 Background and Related Work ............. 4
   2.1 Web Feed .................................. 4
   2.2 RSS Feed .................................. 5
   2.3 RSS reader ................................ 6
      2.3.1 RSS reader: Client Software ........ 6
      2.3.2 RSS reader: Web-based ............. 7
   2.4 TiddlyWiki ................................. 8
      2.4.1 Main Features ......................... 8
   2.5 TiddlyCard ................................. 9
   2.6 AJAX .................................... 11

3 Project Objectives ......................... 20
   3.1 Why TiddlyRSS? ............................ 11
   3.2 Project Objectives ....................... 12

4 The TiddlyRSS ................................. 20
   4.1 Core Functions ............................ 12
      4.1.1 RSS Feed Subscription ............... 12
      4.1.2 Server Communication ................. 13
      4.1.3 Parse XML File ......................... 14
      4.1.4 Generate News Title List ............ 16
   4.2 Advanced Features ....................... 17
      4.2.1 Folders ................................ 18
      4.2.2 Filter/Sort Title/Time ............... 19
      4.2.3 Comments ............................... 20
      4.2.4 Highlights ............................. 20
      4.2.5 Version Control ....................... 23
      4.2.6 Appointment Manager ................. 24
   4.3 TiddlyWikiRSS ............................. 25
      4.3.1 Navigation ............................. 26
CHAPTER 1

Introduction

Through the World Wide Web, people are able to track political events, celebrity gossip, and technology advancements. However, tremendous amount of information available online could be easily overwhelming, and hopping from site to site in search of new updates can waste hours. In addition, news websites such as CNN are frequently updating their content. As a result, users need to constantly visit the entire websites in order to track the updated news headlines. To avoid these troubles, we will need two content management tools. One is from the content publisher, to organize and publish the information to the internet users. The other is from the client side, to capture the information released by the publisher.

The content management tool from the publisher is called RSS – Really Simple Syndication. It turns latest articles and news sources into an automatically generated and organized web data format. The other tool from the client side is called RSS reader. It retrieves and syndicates the data format from the publisher and organizes and displays the information in a single location for easy viewing.

With the help of these tools, users will no longer need to look through various items from site to site in search of the new updates. Instead, what they need to do is merely
reading the syndicated updates retrieved from the publisher on their RSS readers. Various contents from different websites are easily viewed and managed by the RSS reader.

However, problem arises as the existing two types of RSS readers both have pitfalls. One is client software that requires installation which is not convenient and takes up disk space. The other is browser-based application that requires the access to the internet and might violates the user privacy. This project explores the solution for combining the strengths and eliminating the problems of the existing RSS readers. A new RSS reader called TiddlyRSS is designed to be a browser-based application relies on the client side. It is available offline and there is no installation on the client side or registration on any hosting website. In fact, it is only a piece of DHTML page running on the browser. Therefore, it is extremely portable, can be used on mobile devices. The full-fledged JavaScript library and cutting-edge web technology AJAX and Tiddlywiki has made this application possible. The TiddlyRSS can make full use of the browser integration of JavaScript with other application such as TiddlyCard, Dot and TiddlyCalendar.

As a result, my project consists of two stages. In the first stage, I develop the TiddlyRSS reader within the Tiddlywiki environment as a plugin application. In the second stage, I further develop the TiddlyRSS reader within the TiddlyCard environment as command line application and integrate the TiddlyCard RSS reader with
version control and TiddlyCalendar in order to demonstrate its ability of being applied and integrated with other JavaScript application. Therefore, the TiddlyRSS refers to TiddlyWiki RSS in the first part and TiddlyCard RSS in the second part.

The rest of the paper is organized as follows: we introduce the background and related work in chapter 2 and hence explain why the TiddlyRSS is necessary and useful and specify the concrete objectives of this project in Chapter 3. In chapter 4, we explain the core functions and advanced features of the TiddlyRSS. We discuss the two stages of TiddlyRSS, TiddlyWikiRSS and TiddlyCardRSS and how they can be integrated with other applications. In chapter 5, we compare the TiddlyRSS with other RSS readers and discuss the advantages of TiddlyRSS. In chapter 6, we explore the possible improvements. Finally, in chapter 7, we conclude the project.
CHAPTER 2

Background and Related Work

2.1 Web Feed

Web feed is a data format captures dynamically updating web content. It is published by the content providers such as news websites and weblogs and syndicated and subscribed by the user with a feed reader.

Regularly updated news articles or blog entries are the main sources of the web feed content. Feed normally contains updated summary of the web content along with links to the full versions of that content. Besides, feeds are also used to describe structured information such as weather data and search results.

Web feed is a document delivers content in HTML, and structure and store the content in XML format. It is designed to be machine-readable rather than human-readable. Thus, web feed requires a feed reader to display the document in a well-organized manner. Two main web feed formats are RSS and Atom. RSS is the most popular web feed used by most of the websites.
2.2 RSS Feed

RSS - Really Simple Syndication is a family of web feed formats for content distribution.

RSS feeds are provided by site owners to publish a list of recent articles or content in a standardized and machine-readable format.

RSS document may include headlines, summaries and links back to the publishing website for the full article.

RSS content can be syndicated by other websites that downloaded the RSS feed, or by feed readers that subscribed the feed.

Really Simple Syndication format (RSS 2.0) is preceded by earlier attempts such as RDF Site Summary (RSS 1.0 and RSS 0.90) and Rich Site Summary (RSS 0.91) that did not achieve widespread popularity.

RSS formats are specified using XML. Here is an example of RSS 2.0 document [1]:

```xml
<xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Liftoff News</title>
    <link>http://liftoff.msfc.nasa.gov/</link>
    <description>Liftoff to Space Exploration.</description>
    <language>en-us</language>
  </channel>
</rss>
```
How do Americans get ready to work with Russians aboard the International Space Station? They take a crash course in culture, language and protocol at Russia's Star City.

Compared to earlier spacecraft, the International Space Station has many luxuries, but laundry facilities are not one of them. Instead, astronauts have other options.
2.3 RSS reader

RSS reader is a client software or web application that retrieves and interprets RSS feed and provides a consolidated view of the updated web content such as blogs and news headlines. It becomes a necessary tool to track the dynamic updates of the web content and display them in a single location for easy viewing.

Users subscribe the RSS feeds by entering the feed URL into the RSS reader. RSS reader will check the feed regularly and download new updates from the feed. Users will be able to view the latest updates from several websites in one display, thus, diminishing the needs of clicking around different sites to conduct search. Usually, the single news or blog entry display will consists of a summary of updated content and a link to its full version.

Many RSS readers are “stand-alone” programs on client side. The others are web services supported by web servers.
2.3.1 RSS reader: Client Software

RSS reader as client software is a desktop application. After retrieving the new updates from the feed URL via internet, for the rest of the time, it is able to work offline for users to view the updates and manage the display. However, it requires installation and takes up disk space, thus, it is not portable. The graphical user interface of this kind of reader is user-friendly and often consists of three-panels such as in Figure 1.

![Tristana Reader screenshot](image)

Figure 2.2: Tristana RSS reader screenshot [2]

2.3.2 RSS reader: Web-based
Web-based RSS reader resides on the remote server. Users register on the web portal and login to their personal web RSS reader to subscribe and read feeds with access to the internet from any computer. The RSS reader application is maintained on the server. Therefore, there is no installation required and it can be accessed from anywhere with internet connection. However, it can not work offline, and the client registration and subscription of feeds also release private information.

Figure 2.3: Bloglines RSS reader screenshot [3]

2.4 Tiddlywiki
Tiddlywiki [4] is a single page browser application running on the client side, without needing any server side logic. It is entirely self-contained in a single HTML file that includes CSS and JavaScript code. It is extremely portable and can be posted onto web server, sent by email or kept on thumb drive. It is designed as a personal note book and allows users to view and edit hypertext document at the same place.

A feature that set Tiddlywiki apart from other web technology is its content presentation. Before Tiddlywiki is invented, web content is dominated by page concept. In Tiddlywiki, the hypertext document is presented in micro content format namely tiddler which is self-contained fragment rather than entire page. Therefore, Tiddlywiki is like a non-linear blog being divided into little chunks (tiddlers). It aggregates the micro content items into a whole. Users view the items by clicking the hypertext links rather than sequentially.

Tiddlywiki adopts the wiki concept and serves as a database for creating and browsing information. However, it is fundamentally different from conventional wiki as wiki manipulates with pages and the pages are highly interconnected via hyperlinks.

A defining characteristic of Tiddlywiki is its code editing and interpreting. It serves as an editor as well as an interpreter. Programmers could use Tiddlywiki as JavaScript editor and create a code fragment as in a tiddler. When users finish coding, the Tiddlywiki will automatically interpret the code and immediately display the result in
the same setting.

Hence, Tiddlywiki is an excellent environment to write and test JavaScript application. Tiddlywiki itself is written using JavaScript, CSS and HTML. There is a lot of supporting methods defined in Tiddlywiki available to write a plugin application for Tiddlywiki. Therefore, in the first part of this project, we use Tiddlywiki as a platform to develop the RSS reader.

2.4.1 Main Features

In this project, we will make use of TiddlyWiki’s special features such as tiddler, macro, tags, non-wiki word links, and plugins.

**Tiddler** - a name given to a unit of micro content in TiddlyWiki. It can be initiated by clicking a new tiddler button. Each tiddler has two modes: editing mode and view mode, and three sections: title, text and tag sections. **New tiddler macro** - display a button that can be clicked to create a new tiddler. By default, the new tiddler is opened in edit mode (see Figure 3). The last editing section in the bottom is the place for tags. **Tags** - allows you to categorize a tiddler by assigning it special keywords.

**Macro** - allows you to write tiddlers containing more exotic objects than just text. You can define any new functions using macro. There are also some built-in macros such as new tiddler macro.

**Non-wiki word links** - A set of non wiki words links to a tiddler as hypertext. The tiddler title is the set of non wiki words.
Plugins - TiddlyWiki can be extended by installing plugins that implement new macros, themes, tweaks or other features.

Store Area – A div element contains a list of tiddler div elements. It functions as TiddlyWiki’s data storage contains in the single-file TiddlyWiki itself.

Store – A data structure as TiddlyWiki dynamic storage contains all the tiddlers that are retrieved from store area.

Story – A HTML Div element contains a sequence of tiddlers. Story manipulates all tiddlers that are currently opened on display.

![Figure 2.4: Standard Edit dialog as Tiddler](image)

2.5 TiddlyCard

TiddlyCard is a server side system that combines tiddly-related applications and
support multiple users to concurrently view and create content on the single front-end TiddlyCard page.

TiddlyCard is a framework that leverages on JavaScript, TiddlyWiki and concepts from HyperCard to bring an environment to the web browser. To developers, TiddlyCard is an easy-to-use browser-based environment for rapid application development. To individual users, TiddlyCard is an open collaborative environment integrating various applications such as blog and calendar tool as a whole. To enterprises, TiddlyCard provides an open, sharing environment that will benefit the flow of information within the organization.

TiddlyCard shares similar attributes with TiddlyWiki. First, it is portable as being platform independent and accessible through desktop computers, mobile phones and many other browser-enabled devices. Second, it is client centric and TiddlyCard applications remain functional with or without connectivity. When connectivity is available, TiddlyCard applications automatically synchronize with a server. Third, it is self-contained single HTML file with a free layout building tool, as well as a built-in programming environment and free content management system.

As such, TiddlyCard serves as an excellent platform to develop and integrate TiddlyRSS.
AJAX

AJAX stands for Asynchronous JavaScript And XML. It is a new web technology for
creating better, faster, and more interactive web applications. AJAX makes web pages more responsive by exchanging small amount of data with the server behind the scenes, so that the entire web page does not have to be reloaded each time the user requests a change. [6].

AJAX uses JavaScript technology through the XMLHttpRequest object to handle all the server communication. When user exchange data with server, the data is not sent directly to the server, instead, JavaScript program will send a request to the server and load data from server behind the scenes, without interfering with the behavior of the existing page. And the request is sent asynchronously such that JavaScript program will not wait for the server to respond. Therefore, the user can continue using their application without the need of waiting for the server to process the request.

In addition, only a small amount of data is requested in this case and server does not need to send an entire HTML web page back to the user, because JavaScript program will handle the data transfer and local HTML generation for the user in the background, and XMLHttpRequest object is able to talk back and forth with the server without any user intervention. As a result, AJAX makes web application highly interactive, much more responsive and user-friendly like a desktop application.

In addition, AJAX is browser and platform independent as it is based on well-defined web standards such as JavaScript and DOM.

In this project, the core of TiddlyRSS is developed using AJAX techniques.
Project Objectives

3.1 Why TiddlyRSS?
As discussed in the introduction, we know that the existing RSS readers all have pitfalls. They are either server side web application or client side desktop application. For web-based RSS readers, they are relying on the connectivity with the web server. Users need to have access to the internet and to reveal personal information to the web server. For client RSS reader software, installation is required thus it is not portable. Therefore, our initial goal is to redefine the RSS reader as a client side browser-based application. TiddlyRSS is designed to be extremely portable and requires no installation. It is a highly-interactive web application.

3.2 Project Objectives
In order to build a RSS reader, two things have to be taken care of, one is the data
communication with any news or blog website server, the other is the content rendering and storage. Since our TiddlyRSS is designed to be a client side browser-based application, AJAX technology is used to handle the data communication between browser and server and develop a highly interactive and responsive RSS reader. TiddlyWiki as a self-contained single page HTML file will be used as a rendering machine to generate the updated news or blog entry as micro content and as a database to store the content.

In addition, we use TiddlyWiki as platform to develop the TiddlyRSS application by making use of TiddlyWiki built-in methods and data structures. Therefore, the first part of the project is to develop the TiddlyRSS as a plugin application for TiddlyWiki. Thus, the TiddlyRSS along with TiddlyWiki will be the single page browser-based application resides on the client. Clients will be able to download the single page HTML file and run the TiddlyRSS immediately on any browser without any installation. Once the RSS feeds are subscribed and updated content are retrieved from the websites, TiddlyRSS will be able to work offline at any time as a desktop application available for users to manage the news or blog items. It is also extremely portable as it is a single HTML page which can be sent by email, store in thumb drive, used in mobile devices.
Web feed

RSS feed

ISA

Subscribe

RSS reader

Web-based server-side

TiddlyRSS: browser-based client-side

TiddlyWiki

TiddlyCard

AJAX

HTML

JavaScript

Figure 3.1: Relationship Chart
CHAPTER 4

The TiddlyRSS

4.1 Core Functions

TiddlyRSS refers to TiddlyWikiRSS and TiddlyCardRSS. TiddlyWikiRSS is built on top of TiddlyWiki framework. TiddlyCardRSS is a plugin application which can be integrated into TiddlyCard environment.

Both applications are implemented by defining macros and functions. In particular, a main RSS reader macro is defined. It is responsible for executing all the main functions and features for RSS reader application. Many other macros are called and initiated from this macro. The macro is triggered by putting itself into a tiddler, whenever the tiddler is opened, the macro will be executed immediately. This is achieved by making use of TiddlyWiki’s story data structure. Story manipulates all the tiddlers to be opened. Story is able to interpret the content of the tiddler, execute and display the content properly. Therefore, once the tiddler which contains the RSS reader macro is opened, story is able to execute the macro.

The basic working flow of TiddlyRSS is illustrated by the following chart:
4.1.1 RSS Feed Subscription

RSS feed is a RSS or XML file with a feed URL. The user subscribes a feed by clicking add channel button. A tiddler window will pop up prompting user to enter the feed URL.

![TiddlyRSS core working flow chart]

Figure 4.1: TiddlyRSS core working flow chart

**NY Times**

```html
<<tiddlywikiRSS asHtml http://www.nytimes.com/services/xml/rss.xml>>
```

Figure 4.2: Feed channel tiddler for NY Times
We are able to achieve this function by making use of TiddlyWiki’s tiddler and macro features. Tiddler is used to create and store a new feed channel. User is also able to name the feed channel in the title field of tiddler editing dialog box and assign any tags to the feed channel tiddler.

4.1.2 Server Communication

Our TiddlyRSS handles server communication by making use of AJAX technique. To initiate the connection with RSS feed channel server, JavaScript program in the RSS reader macro will prepare an XMLHttpRequest object first. If using IE as browser, the object will be a new ActiveXObject. Next, the program will open a connection to the server by making a GET request to the server with the feed URL, and specify a On-ready-state-change function which tell the server what to do when it finishes running. The function checks if the state is 4 meaning loading of server XML response is complete, and if the URL exists, it will proceed to process the response object. Finally, send any information you want to send along in the body of your request.

4.1.3 Parse XML File

After receiving the server XML response, we will parse the XML document object by analyzing each tagged element. We go through the whole XML document which contains a list of news items, retrieve each news item by its news title and description tagged elements, and store the retrieved content in a tiddler and assign news tiddler tag. The TiddlyWiki built-in storage area becomes the database for our news items or blog
entries, thus, diminish the need for additional separate database. If a news item title already exists in the storage, we will directly retrieve the news tiddler instead of creating new tiddler for that news.

4.1.4 Generate News Title List
In the feed channel tiddler, after successfully getting the server response XML file and storing all the news items, we will create and display a news title list in the channel tiddler. Since TiddlyWiki provides the feature of non-wiki-word links, we can wrap each news title with a hyperlink that links to the news tiddler. Thus, user is able to view a particular news tiddler by clicking the title in the news title list.

4.2 Advanced Features
There are many features implemented for the TiddlyRSS. Here we only choose a few to illustrate the special characteristics of TiddlyRSS.

4.2.1 Folders

The concept of folders comes in nature as we want to allow users to organize and categorize their news/blog feed channels. Users are able to group different channels into different categories by putting them into different folders. TiddlyRSS provides two built-in folders as well as an option for user to define their own folders. For example, users can add a Business folder simply by specifying the name Business in the prompt window, and then put a feed channel which relates to business area into the Business folder using the AddIntoFolder button. Thus, a folder may contain a list of channels, and a channel may belong to several folders.

We are able to establish this folder option by making use of TiddlyWiki’s tags. Tags themselves are their own tiddler titles. For example, a tag named Business will automatically associate with a tiddler with title Business. This Business tiddler will by default contains a tagged list. Each element in the list is a link to another tiddler which has the tag Business. In short, a tag is a tiddler that collects all other tiddlers which have this tag in their tag field. Thus, the folder we created is actually a new tag. Putting a channel into a folder is actually done by assigning a tag to this channel.

4.2.2 Filter/Sort Title/Time
TiddlyRSS offers filter function in which users can specify a keyword and then only those news/blog entries with titles contain the keyword will be displayed. Besides, users are also able to sort news/blog items by title or time. Items can be sorted according to the publish time or in alphabetical order or their reverse orders.

The logic behind these three functions is the macro function call reset and channel tiddler re-rendering. We know already that RSS reader macro is triggered upon opening the channel tiddler that contains this macro. The macro is executed immediately and the list of news items is created and displayed dynamically. In other words, we do not assign static text to the tiddler for creating the item list. Instead, we display them on the fly. The reason is that item list is a very dynamic content, each time the RSS reader is initiated and the content are loaded from the server website, and the list generated may be different. Therefore, we do not use static text format. However, the user may only specify their filter or sorting option after the dynamic content is displayed and we can’t do change to the display. Therefore, we need to reset the macro call to specify the filter or sorting requirement, and re-render the tiddler to re-execute the macro.

4.2.3 Comments

One of TiddlyRSS defining feature is the commenting function. Users are able to write
comments for particular news. Comments are edited and stored in tiddlers. As we know that TiddlyWiki is an excellent editing tool, the commenting function could utilize TiddlyWiki’s enhanced editing features. Besides, the embedded hyperlinks enable easy navigation between news and comments. Thus, comments can actually be a full article or other kinds of documentation that relates to the original news.

4.2.4 Highlights

TiddlyRSS highlights the item title in the item list whenever a particular item is viewed in its full version. In the news channel tiddler, listed news title will be highlighted as red once the external link to the full article is clicked. We are able to keep track of such link clicking history by modification of external link on-click function.

4.2.5 Version Control

TiddlyCard is a server side system that combines tiddly-related applications and supports multiple users to concurrently view and create content on the single front-end
TiddlyCard page. In particular, different users may make changes to the same tiddler several times. Each version of the change has to be recorded and updated properly. Version Control is a mechanism for synchronizing the tiddlers between multiple users. It is responsible for merging different tiddler versions as well as keeping a history of changes. Version Control keeps version information for each user. For example, it records which version users own now and what changes has been made between the current and the previous version.

Since the version information is updated frequently which is a situation analogical to updated news headlines and blog entries, we can make use of TiddlyRSS to keep track of and report the version information for the TiddlyCard users.

The version information is treated as a feed channel for Version Control. TiddlyRSS retrieved the version information from the Version Control and display the information in the channel tiddler.

Through this function, we can envisage that TiddlyRSS is not only a RSS reader that tracks news/blog updates, it has the potential to be a tracking system for all kinds of updates.

4.2.6 Appointment Manager

Appointment Manager is one of the components in TiddlyCalendar. TiddlyCalendar is a Personal Information Management (PIM) application developed for TiddlyCard.
This calendar tool is created to help users of TiddlyCalendar to manage appointments and events from the same workspace in the web browser. In particular, Appointment Manager manages all the events in TiddlyCalendar. It provides the core functions for the Calendar Manager to add, edit, delete and even search events.

The events are frequently updated content which need to be keep track of. Since TiddlyRSS is a kind of tracking system, we want to explore the possibility of using TiddlyRSS to track the appointment updates. Hence, we could show that TiddlyRSS is a powerful tracking tool that is able to keep track of different kinds of updated content.

In order to do the integration with Appointment manager, we first need to create a RSS file to publish all the updated appointments from Appointment Manager. Then return this RSS file to the application so that it could post the file on some host server. Then the appointment manager will publish the feed URL of this RSS file. TiddlyRSS will subscribe the feed and display the updates information and provide a function for users to update their TiddlyCalendar to handle the updated events.

4.3 TiddlyWikiRSS

TiddlyWiki is a development tool for TiddlyRSS. RSS reader is an integrated application that built on top of TiddlyWiki framework. The entire TiddlyWiki is there to be customized and reshaped to be a RSS reader. RSS reader is the final application
that makes heavy use of TiddlyWiki’s pre-defined methods, built-in data structure and storage, micro content and self-contained concepts.

**Figure 4.3**: TiddlyWikiRSS screenshot

This section explores the synergy between TiddlyWiki and TiddlyRSS.

### 4.3.1 Navigation

TiddlyWiki is wiki-modeled with its hypertext links. Navigation is through hyperlinks rather than sequentially. Traditionally, the content on the web are organized and presented in some kind of hierarchical order or logical order. Navigation and linking of content depends on the logical structure of the content. However, with TiddlyWiki’s hyperlinks, content is linked with similar or related content. As a result, linking depends on the content itself rather than the structure of content.
TiddlyRSS incorporates this hyperlink feature for the linking between news and comments. Each news item has embedded hyperlinks of its comments, while each comment has embedded news hyperlinks. Users are able to jump from news to comments, and traverse back to the news.

4.3.2 Hierarchical Structure

Besides the hyperlinks structure we discussed in the previous section, TiddlyRSS also makes use of the hierarchical structure which is realized by TiddlyWiki’s tags mechanism. Tags in TiddlyWiki are tiddlers themselves. A tag is a tiddler that collects all other tiddlers which have this tag in their tag field. Therefore, the hierarchical structure is presented such that a tag links to a list of tiddlers and each tiddler has its own tags.

The hierarchical order in TiddlyRSS is the folder, channel and news structure. There is a list of folders, each folder contains a list of channels, and each channel contains a list of news items. The folder is actually a tag tiddler. The folder lists all the channels by making use of the tag tiddler that collects all the tiddlers with that tag.

4.3.3 Tiddler

TiddlyRSS makes full use of TiddlyWiki’s tiddlers. The tiddlers are used in three ways, tag tiddler, macro tiddler, and content tiddler.
TiddlyRSS folders are tag tiddlers. A tag tiddler collects and displays all the tiddlers with that tag. The tiddler title is the tag itself, and the content is an automatically generated list. Such tag tiddler allows us to automatically group tiddlers together.

Macro tiddlers are tiddlers contain and execute macros. Once a tiddler is opened, TiddlyWiki will interpret the content of the tiddler, if it encounters a macro, the macro will be executed. The display of the macro tiddler depends on the definition of macro. TiddlyRSS channel tiddler is an example of macro tiddler. Each channel tiddler is corresponding to a feed channel, and it contains a RSS reader macro along with the feed URL information. Once the channel is opened, the RSS reader macro will be executed and a list of news items will be displayed.

Content tiddler is the normal tiddler for any micro content. It contains simple text or other document format. News in TiddlyRSS are stored and displayed in content tiddlers.

4.3.4 TiddlyWikiRSS Usage

TiddlyWikiRSS can be used as a stand-alone RSS reader application. Users can easily download it and run the application on any modern browser without any installation. It works both online and offline. TiddlyWikiRSS is extremely portable. It can be sent by
email, store in thumb drive and posted on the web. It is platform independent and can be accessed via mobile devices.

4.4 TiddlyCardRSS

Conceptually different from TiddlyWikiRSS, TiddlyCardRSS is not an application that makes use of TiddlyCard, instead, it is one of many applications that can collaborate with each other within TiddlyCard environment and together form a collaboration tool.
We extend the first stage TiddlyWikiRSS into this second stage TiddlyCardRSS in order to demonstrate that our TiddlyRSS actually has a great potential to be integrated into other JavaScript environment gracefully. In addition, the TiddlyCardRSS defines a number of interface methods which can be used as commands. Therefore, TiddlyCardRSS is also command line application can be called from TiddlyCard and other JavaScript applications.

![TiddlyCardRSS screenshot](image)

Figure 4.4: TiddlyCardRSS screenshot

This section explores the synergy between TiddlyCard and TiddlyRSS

4.4.1 Story

Story is a HTML div element contains a sequence of tiddlers in TiddlyWiki. Story
manipulates all tiddlers that are currently opened on display. Because TiddlyWiki is single page application, there is only one story that manipulates all the tiddlers on display. However, TiddlyCardRSS requires several stories to build a stand-alone application that can be plugged into TiddlyCard environment.

TiddlyCardRSS has its own panels such as folder panel, channel panel and news panel, as compared to TiddlyWiki’s single page. Panels are represented in hierarchical order such that folder contains list of channels, channel contains list of news. Each panel is actually a new story. Each story manipulates their own sequence of tiddlers for a particular panel. By creating new story, we are able to make use of TiddlyWiki content presentation for each panel.

4.4.2 TiddlyCardRSS Usage

The application of TiddlyCardRSS is very broad. It can be easily integrated with browser applications such as Appointment Manager and Version Control, and it can be utilized in other applications such as Asalta and blogging websites. Besides, it is also a command line application that provides standard interface. As a result, it can be initiated through command window within other applications such as TiddlyCard.

4.4.2.1 Integration with Appointment Manager

Appointment Manager manages all the events in TiddlyCalendar. The appointments are frequently updated. TiddlyCardRSS is able to integrate with this application to keep track of the updated appointments. TiddlyCardRSS behaves as a tracking tool for
Appointment Manager application. Such integration is achievable because TiddlyRSS is able to define and create a standard RSS file format for the Appointment Manager which means TiddlyCardRSS is actually acting as a content management tool for the Appointment Manager. Appointment Manager will post the file onto some web server, and subsequently, TiddlyCardRSS will subscribe the feed and generate the updated appointments. Such integration demonstrates that TiddlyCardRSS can be a powerful tracking tool for many kinds of updated content, not only news headlines and blogs, but also for content such as appointments.

4.4.2.2 Integration with Version Control

Version Control manages the updates and changes to the tiddlers in TiddlyCard. Different users using TiddlyCard can make changes to the same tiddler, thus, resulting in different version of TiddlyCard. Version Control has to coordinate all these changes. TiddlyRSS as a tracking system is able to keep track of the changes. It will display the version information of TiddlyCard as well as what changes has been done with the last version. Such integration is achievable as TiddlyCardRSS provides functions to access the Version Control and retrieve updates information. It further demonstrates the use of TiddlyCardRSS as an all-in-one tracking system.

4.4.2.3 Integration with Asalta

Asalta is a WYSIWYG webpage editor based on DHTML. It is a tool for designing web pages for the TiddlyCard-based content. It provides a true WYSIWYG experience while developing webpage. The target audience is people who have no or little
experience in designing webpage. With Asalta, users directly manipulate the design and display of the webpage and see the change immediately.

TiddlyCardRSS is a pure JavaScript application that can be used by Asalta to demonstrate the effect of Asalta application. The reason we can apply Asalta on TiddlyCardRSS is that Asalta is actually manipulating the div elements on the webpage, and TiddlyCardRSS is made up of five div elements as panels. The application of TiddlyCardRSS by Asalta demonstrates that as a pure JavaScript application, TiddlyCardRSS is easily integrated and used with other browser application.

4.4.2.4 Command Line Application

TiddlyCardRSS defines a list of interface methods as commands. Therefore, TiddlyCardRSS is also command line application. Other applications which provides console window can call the TiddlyCardRSS commands to run the application. TiddlyCard has a console implementation with a simple command line interface that allows inspection of the JavaScript environment. TiddlyCardRSS can be called by any other application within TiddlyCard environment from command lines.

4.5 Implementation Issue

There are a few difficulties involved in the implementation of TiddlyRSS.

First, TiddlyRSS is primarily built on top of TiddlyWiki. There are more than 7000 lines of this single page TiddlyWiki JavaScript program. Comprehending the whole
chunk of code is very time consuming. Divide and Conquer strategy is applied in the project. The whole program is divided into several parts according to the functionality of each part. We have parameters, templates, formatters, macros, wikifier and helpers. These components are connected with each other in various ways, thus, we need to understand each part as well as the whole program flow.

Second, different browsers provide different implementation for the JavaScript program. The same function may require two different implementations for IE and Mozilla. For example, when creating the XMLHttpRequest object, different ActiveXObject has to be defined for IE. In addition, JavaScript debugging is a very troublesome. Each code piece has to be traced and examined line by line. The simple alert function has been used to check result of each implementation step for debugging.

Third, the logic behind TiddlyWiki’s design has to be studied and applied into developing TiddlyRSS. Since TiddlyRSS has made full use of TiddlyWiki’s pre-defined methods and content presentation, we have to understand and be able to think in TiddlyWiki’s way of defining things such as tags and tiddlers in order to utilize and simplify the TiddlyRSS implementation.
CHAPTER 5

Comparisons

5.1 Comparison with server side RSS reader service
Traditional web-based RSS reader is hosted on some web server. It requires internet access and the user has to stay online while using the service. User has to register on the server with some personal information provided. Besides, user’s choices of feed channels or what updated content they are tracking is known by the server. One example of such web maintained RSS reader service is Google RSS reader. [7]

![Google RSS reader screenshot](image)

Figure 5.1: Google RSS reader screenshot

TiddlyRSS has some clear advantages over web-based server side RSS reader. It is client side browser based RSS reader. Therefore, it is able to work offline. User does not need to stay online all the time. Once it retrieves the latest updates from the server, user can go offline and do any reading or management to the downloaded updates. It does not require any registration, and user can download the software for free. All the activities are private.
5.2 Comparison with desktop application RSS reader

Some RSS reader are desktop application resides on the client side. Such RSS reader is able to work both online and offline and it enables more user-friendly interface. However, it requires installation, therefore, it is not portable. User can only access the service from the computer where the software has been installed. Besides, it will takes up memory space. One of the examples of such RSS reader software is AgileRss.
TiddlyRSS has some clear advantage over such desktop application. It is single page browser based application such that it requires no installation. User can easily download the single page HTML file and run the application immediately. Therefore, it is extremely portable. Users can access the service from any computer or mobile devices. It can be sent via email and carried on thumb drive.

CHAPTER 6

Possible Improvements

6.1 TiddlyWiki Independent
This TiddlyRSS is inspired from the TiddlyWiki. In order to achieve the self-contained single page solution, we make full use of TiddlyWiki’s methodology. Therefore, one of the possible improvements is to develop a TiddlyRSS application independent of TiddlyWiki. Hence, TiddlyRSS will become a real stand-alone application that can work in any environment. It will no longer need the support by TiddlyWiki, and can be integrated with any other browser application that supports JavaScript. For example, the TiddlyRSS can be used as a tool for blog websites.

Blogs are a type of website that contains mainly web pages authored in html. Millions of people use blogs online; a blog is now the new "Home Page", that is, a place where a persona can reveal personal information, and/or build a concept as to who this persona is. Since blog is a home page, users may want to keep track of their favorite updated content on their home page. Thus, TiddlyRSS can be used as RSS reader that embedded in the blogs. Users of the blogs are able to keep a list of feed channels, and access those updated content from the blog. If TiddlyRSS is a pure stand-alone application, it is easy to be imported by any blogs.

6.2 Aesthetics Design

The current version of TiddlyRSS makes sure that all the basic functions and advanced features are working properly. However, due to the tight timeline, it does not take much effort on the aesthetics design. Right now, the TiddlyWikiRSS adopts TiddlyWiki’s content presentation, while the TiddlyCardRSS consists of five panels which look more like a stand-alone application. Panels are presented as rectangular boxes. Each
panel is actually a TiddlyWiki story. Therefore, the TiddlyWiki CSS and templates are applied to the tiddlers on the panels. Possible GUI design improvements are adding icons for buttons, and customizing the existing templates.

6.3 Tracking System

Besides a tracking system only for updated news or blog entry, TiddlyRSS actually can be extended into a tracking system that keeps track of everything that is frequently updated. For example, the tracking function for Version Control already demonstrates such possibility of building an all-in-one updates tracking system. Appointments in the Appointment Manager can be tracked as well.
The main difficulty is that in order to track all kinds of updates, TiddlyRSS has to conform to all kinds of format standards. Different application has defines different content format. For example, RSS feed using XML file format, some other application use text format to save updated content. TiddlyRSS has to conform to all kinds of formats which is unlikely to achieve. Therefore, we have to make TiddlyRSS a content management tool to establish the standards as well as generate the actual file format for other applications. As a result, TiddlyRSS is retrieving and parsing its self-defined format from the content publisher which makes use of the file format TiddlyRSS generated.

CHAPTER 7

Conclusion

In conclusion, the TiddlyRSS created satisfied the initial goal in this project. TiddlyRSS realizes the goal of building a browser-based client side application that works both
online and offline. It is extremely portable as a single page browser application. It requires no installation on the client side. As a pure JavaScript application, TiddlyCardRSS is able to integrate with many other browser applications such as Version Control and Appointment Manager. It has the potential to become an all-in-one tracking system which is able to keep track of all kinds of updated content. However, the aesthetic design has to be improved to attract more and more users.

In the development, TiddlyRSS makes full use of cutting edge web technologies such as TiddlyWiki, AJAX and TiddlyCard. It takes two stages to develop the TiddlyRSS. First is TiddlyWikiRSS for which the RSS reader is built on top of TiddlyWiki. TiddlyWikiRSS makes use of TiddlyWiki as development environment as well as rendering machines. The final product is a single page TiddlyWiki file that is customized with RSS reader functionalities. The second stage is the development of TiddlyCardRSS which is modular stand-alone application integrated in the TiddlyCard environment. TiddlyCardRSS is also a command line application.

References


[3] Bloglines screenshot

http://upload.wikimedia.org/wikipedia/commons/archive/5/58/20070704174127!Bloglines_screenshot.png

Appendix A

TiddlyCardRSS User Manual

1. Install TiddlyCardRSS plugin

There is a button in the left MainMenu called RSSreader. Once you click the button, TiddlyCardRSS is launched and its "main panel" (control panel) will be displayed.

2. There are five buttons in TiddlyCard "main panel" (control panel):
2.1 All Folders
Upon clicking, "folders panel" will be displayed, you will see predefined folders. Later, user defined their own folders and will be displayed in this "folders panel" accordingly. The existing folders: One is "channel" which contains a list of all the channels on this tiddlycardRSS. The other one is "Favoritech" which contains a list of favorite channels which user added into this folder by their own choices. Upon clicking each folder, the list of channels belongs to that folder will be displayed on the "channels panel".

2.2 All Channels
Upon clicking, a list of all channels will be displayed on " channels panel". Each channel is a tiddly link to a news list which will show a list of news retrieving from channel feed URL maintained on the news website server. Upon clicking on each channel link, the list of news belong to that channel will be displayed on "newslist panel". For the first time, there are two existing channels, one is favorite news channel which contains a list of user indicated favorite news, the other one is the commented news channel, which contains a list of previously commented news.

2.3 Add folder
Upon clicking, a dialogue box will pop up, user specify their new folder name and click ok. The new folder is created and shown on the "folders panel".

2.4 Add channel
Upon clicking, a new tiddler in a editing template will be shown on the "newslist panel". In the name of the tiddler, user type in the name of the channel after "Channel Name: ". In the content of the tiddler, the format «<tiddlycardRSS asHtml 'replace this with your feedURL'>> is already presented, user only need to replace the 'replace this with your feedURL' with the feedURL.
If your TW is online version, the format is different: {{{{<<tiddlycardRSS asHtml proxy.cgi?url=feedURL>>}}}} Notice that you need to add @@proxy.cgi?url=@@ before your http://feedURL

<<tiddlycardRSS noDesc|asHtml rssUrl ['filtering string']>>

asHtml: if you know that description contain html (links, img ...),
The text is enclosed with <html> </html>

rssUrl: the RSS feed URL that could be accessed.

After that click done, a new channel is created and displayed on "newslist panel". There will be a list of news shown. Each news item is a title of a news tiddler which upon clicking will be displayed on "news panel".

Example
For example if you have a RSS news channel @@NY Times: http://www.nytimes.com/services/xml/rss/nyt/WorldBusiness.xml@@ which you want to keep track of the updates, you can make use of the tiddlycardRSS through following steps:

- Click the "Addchannel" button on the "main panel". Type @@NY Times@@ after "Channel Name: ".
- Copy http://www.nytimes.com/services/xml/rss/nyt/WorldBusiness.xml to replace 'replace this with your feedURL' so it becomes like this: <<tiddlycardRSS asHtml http://www.nytimes.com/services/xml/rss/nyt/WorldBusiness.xml>> click done.

2.5 Refresh
Upon clicking refresh button, all the news on the tiddlycardRSS will be deleted except
the news which are previously commented or added into favorite news.

3. Functions available in "newslist panel"

3.1. "Filter button": user will be able to specify the filtering string besides the "filter" button. And then click the filter button, only news title contains the filtering string will be displayed on "newslist panel".

3.2. "SortTitle button": upon clicking, the list of news will be sorted and displayed according to the alphabetical order of the titles.

3.3. "SortTime button": upon clicking, the list of news will be sorted and displayed according to the published time, and the time will be shown in UTC time string format.

3.4. "Update button": will clear the cache of the feedURL and redisplay list of news in the basic format which has no filtering or sorting.

3.5. "All unread button": list of news which are not read which means the user does not read the full news article on the news website.

3.6. "AddIntoFolder button": user is able to add the current displayed channel into one of the folders (both existing and user defined ones) by specify the name of folder in the text box. And click the AddIntoFolder button.

4. Functions in "news panel"

4.1. "Open external link": open the external news link. After clicking the external link button, the news will be automatically updated into read news. and the tiddler link of the news title on the "newslist panel" will be changed color to red to indicate user already read this full news.
4.2. "Add comments": upon clicking, the tiddler in editing format will be displayed in "news panel". User is able to editing their comments. After done, the tiddler link of this comment will be automatically appended onto the content of the original news.

4.3. "Add/Remove favorite news": add/remove news from/into user's favorite news channel. After adding or removing, the favorite news channel will be automatically updated.