7-2-2006

Ultra Lightweight Web Applications: A Single-Page Wiki employing a Partial Ajax Solution

Michael J. Rees
Bond University, Michael_Rees@bond.edu.au

Ultra Lightweight Web Applications: A Single-Page Wiki employing a Partial Ajax Solution

Dr Michael Rees [HREF1], School of Information Technology [HREF2] , Faculty of Business, Technology and Sustainable Development, Bond University [HREF3], Qld 4229, Australia. mrees@bond.edu.au

Abstract

The overloaded term Web 2.0 web site usually connotes an interactive web application that offers features normally associated with free-standing applications running directly under the control of an operating system. Such an interactive web applications, also known as a rich internet application (RIA), run within web browsers and must download XHTML and client-side scripts to control user interactivity. Via a variety of technologies the web server must provide a storage mechanism to support the RIA and the presentation of dynamic data in the browser interface. Such storage may be of large volume and bring concomitant bandwidth, response and server storage problems. It is usually the case that the XHTML and client scripts are relatively small in size so the use of the browser in this context to be called a lightweight client. Certainly the dynamic construction of the RIA user interface on demand completely eliminates the download and install problem of free-standing applications and ensures the user always uses the latest version of the RIA.

This paper explores the possibility of building an ultra lightweight RIA where a single web page combines the interactive user interface and the storage mechanism in a single file. The author discovered this approach being used in the TiddlyWiki RIA created by Jeremy Ruston who employed client-side JavaScript to provide all functionality. Here all the main features of a wiki are supported by a single web page. DotWikIE is a re-implementation by the author of an ultra lightweight wiki with significantly improved editing and employing XML for storage of the wiki contents. Apart from being a useful personal wiki application DotWikIE can be extended in a number of ways. An example of automated clipboard monitoring is presented and discussed.

For its implementation DotWikIE uses the JavaScript and XML parts of AJAX (Asynchronous JavaScript and XML). Using full AJAX requires the use of a web server. The paper contains a description of DotWikIEWeb that adds the asynchronous part of AJAX to extend coverage to a more usual web-based wiki while still retaining the simplicity of single, independent wiki web pages for deployment convenience. In the conclusion the paper discusses further extensions of the ultra lightweight RIA and other applications of this RIA implementation technique.

Keywords

Ajax, XML, JavaScript, ultra lightweight web applications, rich internet applications, wikis

Introduction

After 10 years of gestation wikis have become a recognised medium in their own right. In 2005 the first ACM-sponsored meeting on wikis was held, the International Symposium On Wikis 2005, WikiSym '05. The papers discussed both the technical implementations and the social impacts of wikis which are proving to be significant. Web sites used for supporting a community of users will now have a blog, an RSS feed and a wiki. Blogs typically allow any user to add comments to a blog entry but this is in a rigid format following blog entries in a simple chronological list. Wikis on the other hand provide the ultimate feedback mechanism where the circumstance, type of feedback and context determine the extent and format of the information entered by users.

Of course it is important always to mention the Wikipedia phenomenon [HREF6]—the general, online encyclopaedia created as a wiki by tens of thousands of users, that crossed the 1 million article mark early in 2006 for the English version. The shared publishing model for such a huge wiki is now straining the sustainability of Wikipedia as conflicting edits threaten to disrupt the effectiveness of the information. Wikipedia responded by assigning volunteers to act as editors each responsible for a large number of articles. This leads to a monitoring and self-healing process. A visual example of this in action is shown in Jon Udell's screencast [HREF7] animating the changes to a Wikipedia article over time. Nevertheless with smaller, more cohesive user groups the unlimited read/write wiki access model is still effective.

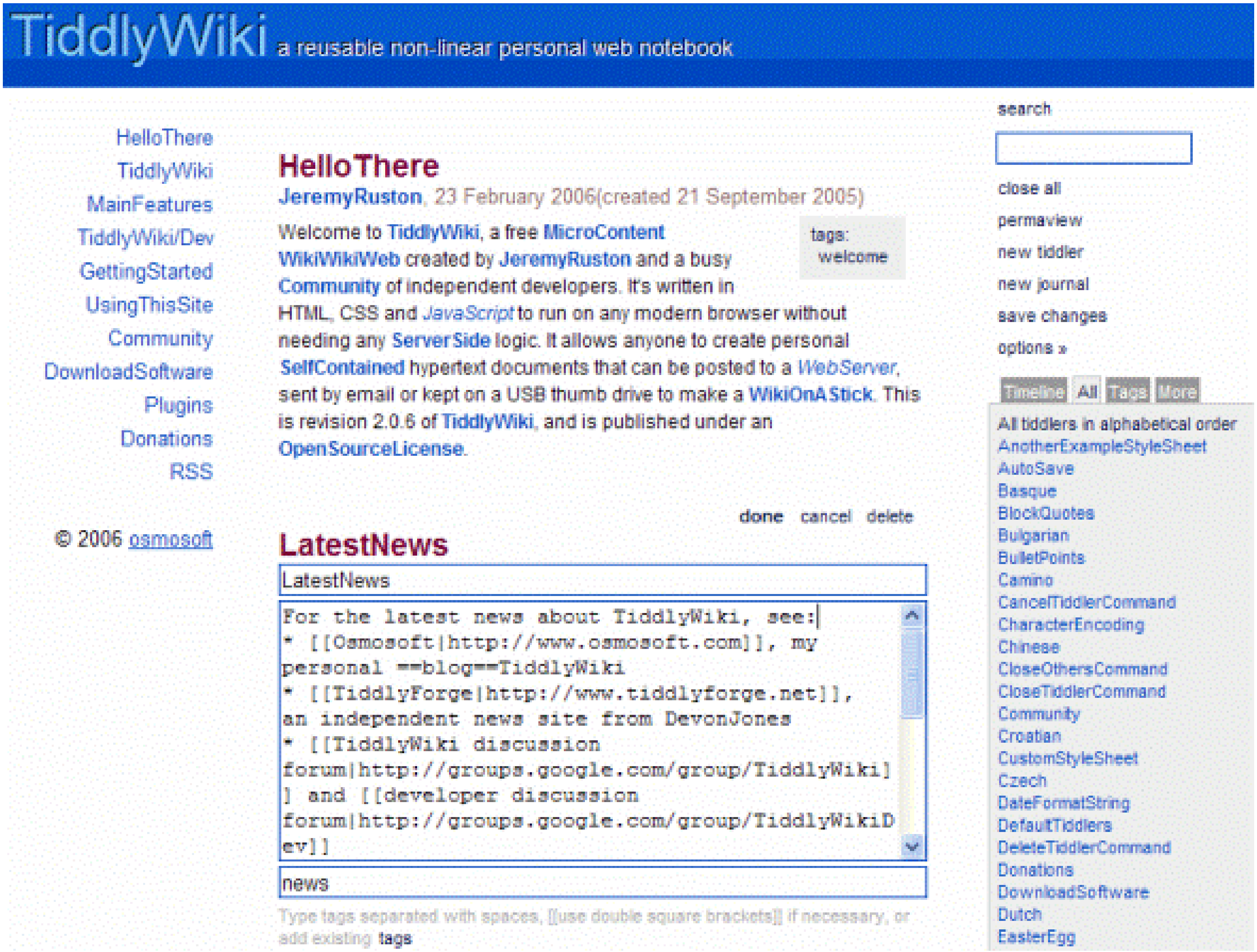
Providing a review of existing wiki applications available for download requires a survey paper in its own right. However, a brief overview and reasonably extensive list of the more popular wiki software is again available in Wikipedia [HREF8]. A glance at this list shows it to be extensive with wiki applications in open source and commercial packages written in a wide variety of traditional and scripting languages. Of course wiki hosting sites often called wiki farms, both free and for hire, also abound [HREF9]. A Web 2.0 commercial wiki farm that has gained considerable traction of late is JotSpot [HREF10] that offers wiki hosting for very small groups all the way up to enterprise grade wikis.

The work on wikis described in this paper was inspired by the TiddlyWiki [HREF4] rich internet application (RIA) by Jeremy Ruston. The whole TiddlyWiki web site itself is constructed out of TiddlyWiki pages so the viewer quickly absorbs the simple TiddlyWiki user interface. The main objective of TiddlyWiki is to encompass the main features of a wiki [HREF5]:

- content is organised in units each identified by a wikiword, a word that starts with a capital letter and contain at least one other capital letter like 'WikiWord'
- all content is easily edited, added and removed using a very simple text-based markup language
- units of content are automatically interlinked whenever their wikiwords are used to form a complete hypertext
- older versions of contents are accessible

The author suggests the term 'ultra lightweight' for TiddlyWiki because the wiki features listed above are implemented in a single web page, ie an individual file containing HTML and the wiki content itself. The wiki content consists of a series of 'tiddlers' which contain text and hyperlinks in standard HTML. Tiddlers are created, edited and deleted through JavaScript embedded in the same file. Updating the tiddler collection is simply a matter of writing the changed HTML, including the static JavaScript, back to the HTML file from which it was originally loaded. Obviously client-side JavaScript in a web page loaded via a web server is prevented from writing to the filestore on the client machine on which a browser is running. However if the web page is loaded locally, directly from the local filestore using the file: protocol, some browsers permit writing files albeit after warning the user of the possible damaging consequences. Internet Explorer and Firefox fall into this category so can be used with TiddlyWiki.

Figure 1 shows a screen capture of TiddlyWiki with the second tiddler opened for editing. Fixed links appear on the left while the right column contains a list of all the tiddlers that all have wikiwords for their title. The various controls and option settings are accessed by link buttons at the top right. Note that no images are visible since they must be stored in separate image files. The HTML form textarea elements used to edit a tiddler shows the typical simple text markup that wikis traditionally use and this representation is stored in the wiki file and converted to HTML dynamically as needed. One useful feature is the ability to tag each tiddler with keywords for later retrieval by keyword. The 'new journal' control hints at using automatically time stamped tiddlers as blog entries, just one of the myriad uses of the wiki pages.



HelloThere

JeremyRuston, 23 February 2006(created 21 September 2005)

Welcome to TiddlyWiki, a free MicroContent WikiWikiWeb created by JeremyRuston and a busy Community of independent developers. It's written in HTML, CSS and JavaScript to run on any modern browser without needing any ServerSide logic. It allows anyone to create personal SelfContained hypertext documents that can be posted to a WebServer, sent by email or kept on a USB thumb drive to make a WikiOnAStick. This is revision 2.0.6 of TiddlyWiki, and is published under an OpenSourceLicense.

LatestNews

LatestNews

```
For the latest news about TiddlyWiki, see:
* [[Osmosoft|http://www.osmosoft.com]], my personal ==blog==TiddlyWiki
* [[TiddlyForge|http://www.tiddlyforge.net]], an independent news site from DevonJones
* [[TiddlyWiki discussion forum|http://groups.google.com/group/TiddlyWiki]] and [[developer discussion forum|http://groups.google.com/group/TiddlyWikiDev]]
```

news

Type tags separated with spaces, [[use double square brackets]] if necessary, or add existing tags

All tiddlers in alphabetical order
AnotherExampleStyleSheet
AutoSave
Basque
BlockQuotes
Bulgarian
BulletPoints
Camino
CancelTiddlerCommand
CharacterEncoding
Chinese
CloseOthersCommand
CloseTiddlerCommand
Community
Croatian
CustomStyleSheet
Czech
DateFormatString
DefaultTiddlers
DeleteTiddlerCommand
Donations
DownloadSoftware
Dutch
EasterEgg

Figure 1. TiddlyWiki Wiki Page.

The simplicity and power of the TiddlyWiki model inspired the author to investigate new potential uses of this potent technique. Note that TiddlyWiki employs only the second of the three technical components of Ajax, namely asynchronicity, JavaScript and XML. TiddlyWiki also follows one of the key philosophies of Web 2.0 Ajax-supported RIAs—limiting each RIA to a single focus with a minimal feature set while remaining a practical application. This latter concept is becoming the mantra of Web 2.0.

As Jeremy Ruston mentions on his web site if a TiddlyWiki page is displayed in a browser then the ultra lightweight RIA has already been downloaded. The user needs only to use the File/Save As... browser menu item to save a copy of the RIA. Another important characteristic of an ultra lightweight RIA is that all the content of the web page is accessible and readable by all so by nature is open source and shareable by all. Since each TiddlyWiki page file is independent, a simple copy and paste operation creates a new wiki file with its own collection of tiddlers. This applies to the version files which are just copies of the original wiki file with timestamps added to the file name. No web server is involved so the user can carry a set of wiki page files on an USB thumb drive to have access to the wiki contents on any machine which simply needs Internet Explorer or Firefox installed to gain access to the wikis. In a similar fashion wiki files can be emailed easily.

A glance at its web site demonstrates the success of the TiddlyWiki model by looking at the list of extensions and variants that others have implemented [HREF11]. While singling out one example is somewhat unfair a very good example of the use of an ultra lightweight RIA is the GTDTiddlyWiki [HREF12] produced by Nathan Bower. This uses the tiddler collection to support the information, typically lists of tasks and actions, needed to support the processes of the well known 'Getting Things Done' (GTD) techniques from time management expert David Allen [HREF13].

TiddlyWiki has definitely created the foundation for an ultra lightweight branch of RIA construction. Provided the information can be stored within the page file itself a whole range of RIAs are possible running the range of information worker applications such a word processors, blogs, spreadsheets, slide show managers, simple databases, calendars, project managers and all types of personal productivity aids. The start of such a trend is to be found at TrimPath Junction [HREF14] where early versions of a task list based on GTD called Next Action is available as well as the beginning of a spreadsheet, Num Sum, that at the time of writing must be hosted on a web site. Another interesting development here is the 'Take It With You' wiki [HREF15] that 'lets you work offline and online. When online, it will persist to the server. When offline, it will save locally via Flash, then sync later on'.

Having worked with what is now called Ajax since 1998 and with in-page editors of various kinds (Rees, 2004 and 2005) the author decided to explore the ultra lightweight RIA model further. The remaining sections of the paper describe the implementation of DotWikIE a rewriting of TiddlyWiki with a number of significant enhancements available on the Internet Explorer foundation. Next a use of DotWikIE to capture clipboard information is presented together with suggestions for how this information gathering might be further processed. The paper concludes by moving DotWikIE to the web server environment to see if the single-page paradigm can still work in a straightforward manner to construct simple web sites with a tool called DotWikIEWeb. Finally, some thoughts on the future of such ultra lightweight RIAs are discussed.

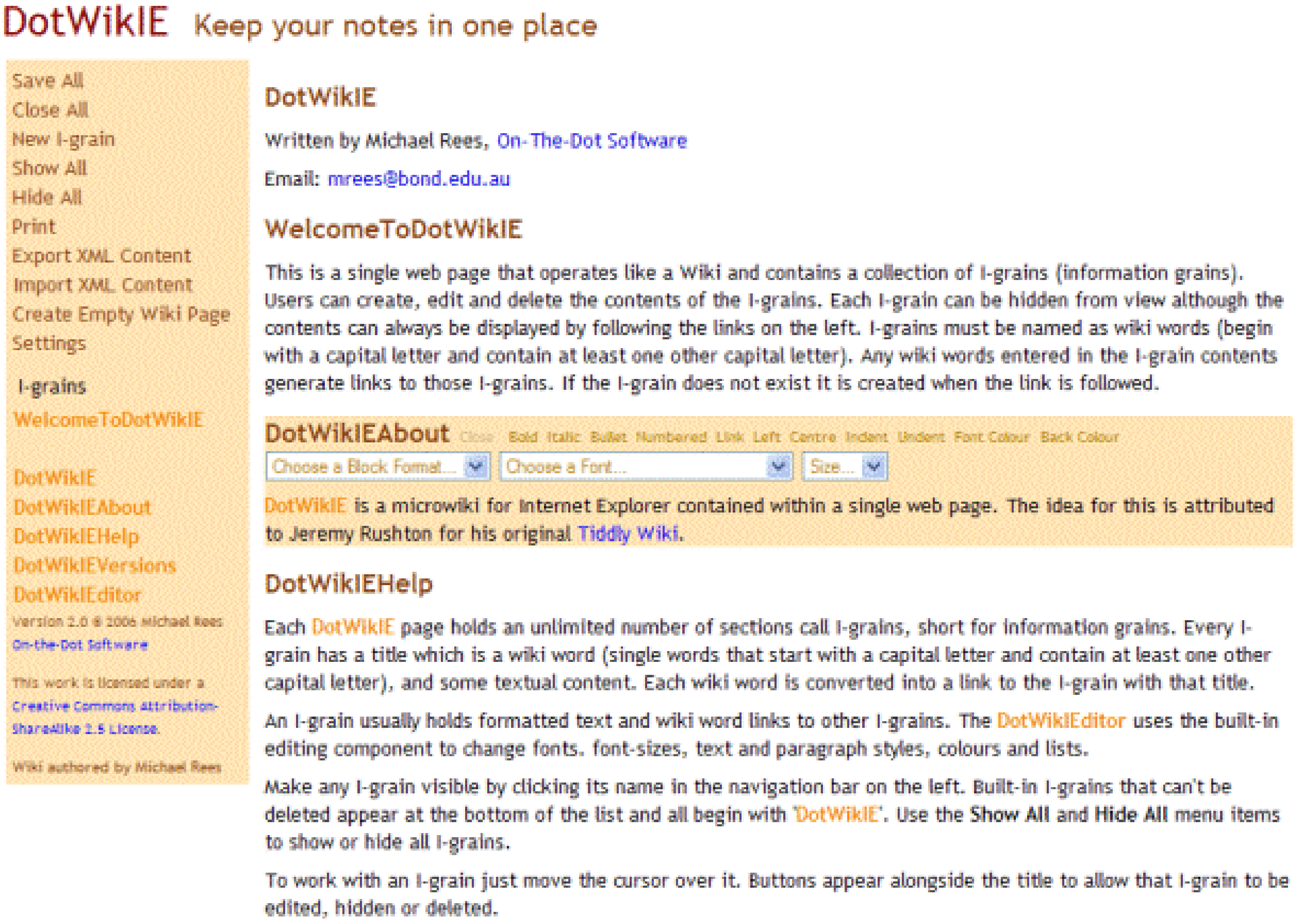
DotWikIE - an ultra lightweight wiki for Internet Explorer

Having written and distributed a number of Ajax in-page editing tools [HREF16] under the On-The Dot software label over recent years the author possesses a library of JavaScript code that could quickly be deployed to implement an ultra lightweight RIA application to support a wiki. This new RIA, DotWikIE, adapts the basic wiki framework of TiddlyWiki but improves upon several key aspects.

The bane of most wikis that use the simple text markup approach is that users must learn the specific markup rules even though they are usually very straightforward. Users have become used to the WYSIWYG editing of graphical user interface word processors and expect nothing less. Internet Explorer incorporates a base version of WYSIWYG editing for HTML that has been exploited by the author and others for a number of years. Not being familiar with HTML editing in Firefox the author limited DotWikIE to work with Internet Explorer alone, hence the name. This decision also allows a number of useful IE-only features to be used whilst still satisfying between 80-90% of all web browser users.

Loading DotWikIE in Internet Explorer SP2 inevitably leads the user to be prompted to allow blocked content. Any web page containing sensitive JavaScript now shows this behaviour that prevents the introduction of web page-borne viruses. The user fortunately needs only to agree to this prompt once. Another lesser prompt about allowing ActiveX execution occurs when the user saves an updated copy of the DotWikIE page or its XML contents. Again this prompt only appears on the first use of the save feature. Once free of these prompts the user of DotWikIE can proceed to use the wiki features quick responses.

While the term tiddler conveys the concept of a small amount of information it was deemed too diminutive. TiddlyWiki tiddlers can contain significant amounts of useful information. For DotWikIE the term I-grain, short for information grain, is adopted following terminology employed by the author's DotEdit tool (Rees, 2000). I-grain hopefully conveys precision and utility at the same time. Figure 2 shows a screen dump of DotWikIE that adopts a simple 2-column layout.



Version 2.0 © 2006 Michael Rees
On-the-Dot Software

This work is licensed under a Creative Commons Attribution-ShareAlike 2.5 License.

Wiki authored by Michael Rees

DotWikIE

Written by Michael Rees, On-The-Dot Software

Email: mrees@bond.edu.au

WelcomeToDotWikIE

This is a single web page that operates like a Wiki and contains a collection of I-grains (information grains). Users can create, edit and delete the contents of the I-grains. Each I-grain can be hidden from view although the contents can always be displayed by following the links on the left. I-grains must be named as wiki words (begin with a capital letter and contain at least one other capital letter). Any wiki words entered in the I-grain contents generate links to those I-grains. If the I-grain does not exist it is created when the link is followed.

DotWikIEAbout

DotWikIE is a microwiki for Internet Explorer contained within a single web page. The idea for this is attributed to Jeremy Rushton for his original Tiddly Wiki.

DotWikIEHelp

Each DotWikIE page holds an unlimited number of sections call I-grains, short for information grains. Every I-grain has a title which is a wiki word (single words that start with a capital letter and contain at least one other capital letter), and some textual content. Each wiki word is converted into a link to the I-grain with that title.

An I-grain usually holds formatted text and wiki word links to other I-grains. The DotWikIEditor uses the built-in editing component to change fonts. font-sizes, text and paragraph styles, colours and lists.

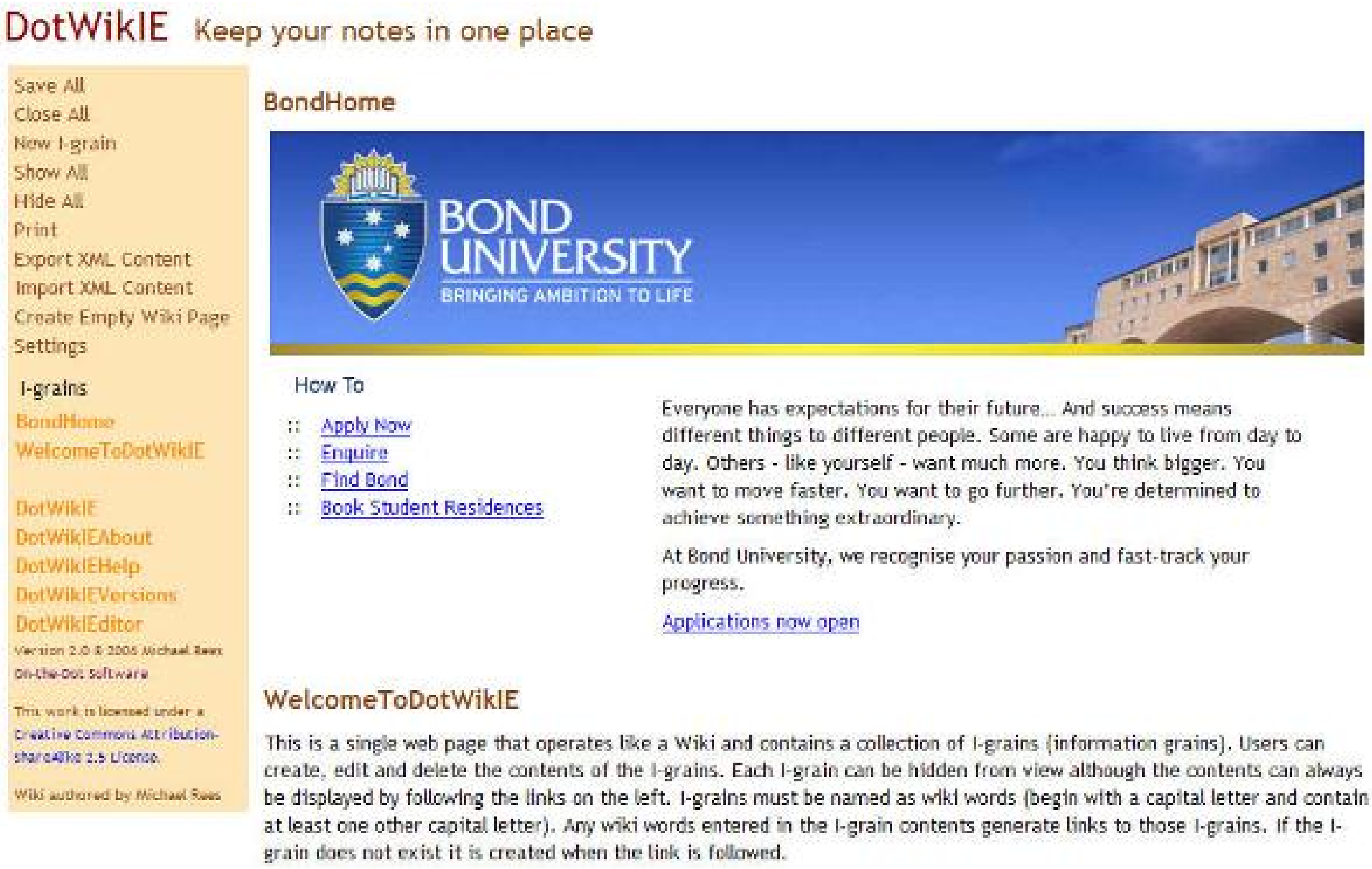
Make any I-grain visible by clicking its name in the navigation bar on the left. Built-in I-grains that can't be deleted appear at the bottom of the list and all begin with 'DotWikIE'. Use the **Show All** and **Hide All** menu items to show or hide all I-grains.

To work with an I-grain just move the cursor over it. Buttons appear alongside the title to allow that I-grain to be edited, hidden or deleted.

Figure 2. DotWikIE Wiki Page.

Text menu items appear on the top left while the complete alphabetical list of I-grains appears next. User-created I-grains are listed first followed by permanent I-grains that all start with the prefix 'DotWikIE'. It will be noted that DotWikIE is distributed as open source under the Creative Commons Attribution-ShareAlike 2.5 License. A selected subset of the I-grain titles and contents appear in the main right column. Following wiki convention the I-grain titles must be wikiwords. Hyperlinks to individual I-grains are inserted automatically wherever the user types the name of the I-grain. This requires some post-processing of the contents as each I-grain is saved. Another wiki trait is the way new I-grains are created simply by entering a new wikiword into the contents of an I-grain. Clicking on the hyperlink to a non-existent I-grain causes it to be created at that time. Hyperlinks to all I-grains appear on the left that cause the I-grain to become visible scrolling the page if needed. It is essential that the creation and management of hyperlinking between I-grains is kept as simple as possible. Misuse of wiki hyperlinks was discovered to be the biggest problem in user trials of wikis (Désilets et al, 2005).

The 'DotWikIEAbout' I-grain in Figure 2 is in edit mode. Note the marked contrast with TiddlyWiki edit mode of Figure 1. In DotwikIE the I-grain contents can be edited in-place and the expected features of font family, size and weight changes, lists, indentation, alignment and colours can all be changed. Hyperlinks to any URL can be inserted and edited. The style or block format dropdown and the list of fonts are populated automatically using the DialogHelper object built-in to Internet Explorer. Not only can the user generate rich formatting of each I-grain contents but can also copy and paste any fragment of a web page as shown in Figure 3. Here a part of Bond University home page was copied and pasted into a new I-grain. Provided the machine running DotWikIE has Internet access then the image and other hyperlinks will render and operate correctly.



Everyone has expectations for their future... And success means different things to different people. Some are happy to live from day to day. Others - like yourself - want much more. You think bigger. You want to move faster. You want to go further. You're determined to achieve something extraordinary.

At Bond University, we recognise your passion and fast-track your progress.

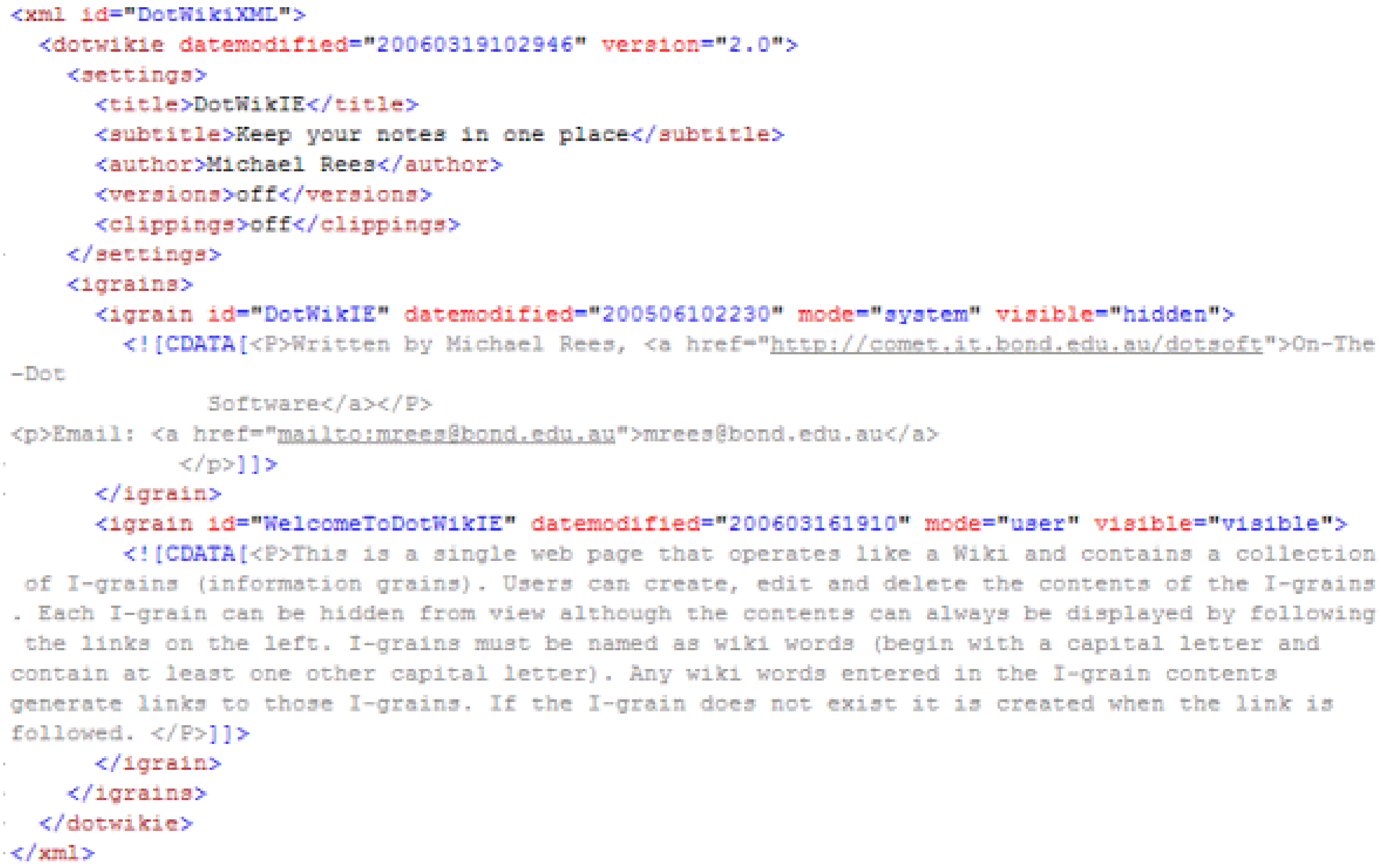
WelcomeToDotWikIE

This is a single web page that operates like a Wiki and contains a collection of I-grains (information grains). Users can create, edit and delete the contents of the I-grains. Each I-grain can be hidden from view although the contents can always be displayed by following the links on the left. I-grains must be named as wiki words (begin with a capital letter and contain at least one other capital letter). Any wiki words entered in the I-grain contents generate links to those I-grains. If the I-grain does not exist it is created when the link is followed.

Figure 3. Inserting Web Page Fragments in DotWikIE.

Individual I-grains are manipulated by hovering the cursor over them. A short menu appears that allows the I-grain to be edited, hidden or deleted as shown in Figure 4. The whole I-grain collection can also be managed with the text menus like 'Show All' and 'Hide All'. By contrast the 'Close All' menu saves all the I-grains open for editing; it is possible to have many I-grains open for editing to allow copy and paste between them. The currently visible I-grain collection can be printed using the 'Print' menu that hides the left column before printing and restores it afterwards.

Another major innovation of DotWikIE is to use XML to store the contents of each I-grain. This is made possible within the DotWikIE file through the use of the Internet Explorer HTML data island tag, <xml>, which encapsulates an XML document within the HTML itself. The first two I-grains shown in Figure 2 would be simply represented as shown in Figure 4. A similar XML data island is used to store the XSLT style sheet that transforms the XML I-grain contents to HTML for rendering on the DotWikIE page. The support of the XML DOM, XSLT and XPath expressions within client-side JavaScript is made possible by the ever present MSXML.DOMDocument ActiveX control available to Internet Explorer.

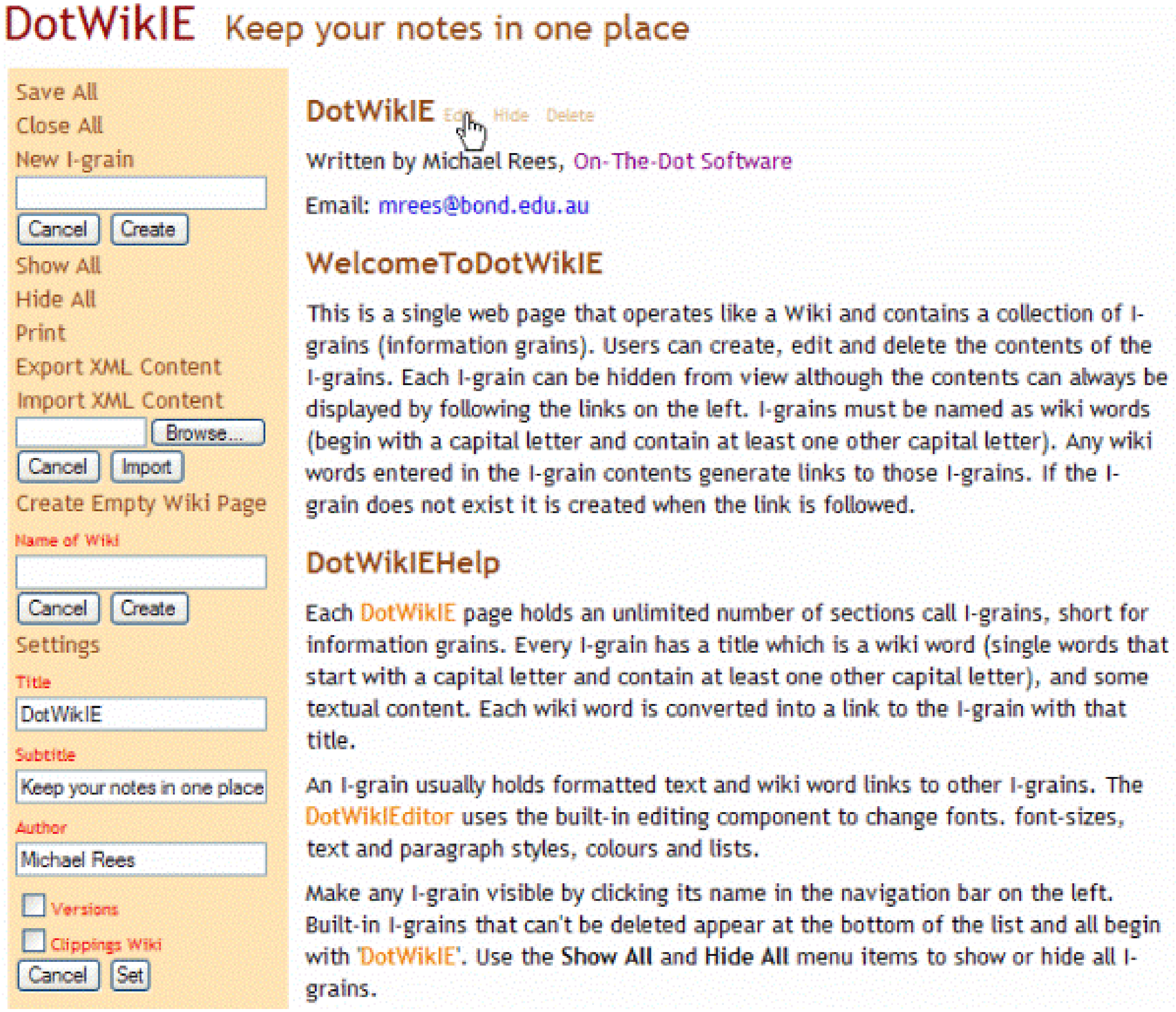


```
<xml id="DotWikiXML">
  <dotwikie datemodified="20060319102946" version="2.0">
    <settings>
      <title>DotWikIE</title>
      <subtitle>Keep your notes in one place</subtitle>
      <author>Michael Rees</author>
      <versions>off</versions>
      <clippings>off</clippings>
    </settings>
    <igrains>
      <igrain id="DotWikIE" datemodified="200506102230" mode="system" visible="hidden">
        <![CDATA[<P>Written by Michael Rees, <a href="http://comet.it.bond.edu.au/dotsoft">On-The
-Dot
              Software</a></P>
<p>Email: <a href="mailto:mrees@bond.edu.au">mrees@bond.edu.au</a>
            </p>]]>
      </igrain>
      <igrain id="WelcomeToDotWikIE" datemodified="200603161910" mode="user" visible="visible">
        <![CDATA[<P>This is a single web page that operates like a Wiki and contains a collection
 of I-grains (information grains). Users can create, edit and delete the contents of the I-grains
. Each I-grain can be hidden from view although the contents can always be displayed by following
 the links on the left. I-grains must be named as wiki words (begin with a capital letter and
contain at least one other capital letter). Any wiki words entered in the I-grain contents
generate links to those I-grains. If the I-grain does not exist it is created when the link is
followed. </P>]]>
      </igrain>
    </igrains>
  </dotwikie>
</xml>
```

Figure 4. I-grains as XML.

Figure 5 also shows the menus that require further input from the user. 'New I-grain' is the direct way to create a new I-grain by prompting the user for a new name which must of course be a wikiword unique in the wiki. The I-grains XML representation can be output to a separate XML document using the 'Export XML Content' menu item which names the document with the wiki title and adds a time stamp. Using the 'Import XML Content' menu it is possible to browse to any of these exported DotWikIE XML documents to insert the I-grains into the current wiki. Thus a simple copy and paste mechanism is achieved for I-grains between different DotWikIE files. However, the XML documents become useful in their own right as archives but more importantly for further processing in ways that are not fully realised to date. One such use is discussed later in this paper.

The 'Create Empty Wiki Page' menu in Figure 5 shows the way to spawn new DotWikIE pages conveniently. All content, including the permanent 'DotWikIE' I-grains, are deleted a new wiki page is created and loaded into the browser. A new collection of I-grains can be built in the new file. Of course Windows Explorer can also be used to create any number of wiki file copies each with a different title so that the user can store collections of I-grains in several categories.



DotWikIE Keep your notes in one place

Title: DotWikIE

Subtitle: Keep your notes in one place

Author: Michael Rees

☐ Versions

☐ Clippings Wiki

DotWikIE

Written by Michael Rees, On-The-Dot Software

Email: mrees@bond.edu.au

WelcomeToDotWikIE

This is a single web page that operates like a Wiki and contains a collection of I-grains (information grains). Users can create, edit and delete the contents of the I-grains. Each I-grain can be hidden from view although the contents can always be displayed by following the links on the left. I-grains must be named as wiki words (begin with a capital letter and contain at least one other capital letter). Any wiki words entered in the I-grain contents generate links to those I-grains. If the I-grain does not exist it is created when the link is followed.

DotWikIEHelp

Each DotWikIE page holds an unlimited number of sections call I-grains, short for information grains. Every I-grain has a title which is a wiki word (single words that start with a capital letter and contain at least one other capital letter), and some textual content. Each wiki word is converted into a link to the I-grain with that title.

An I-grain usually holds formatted text and wiki word links to other I-grains. The DotWikIEditor uses the built-in editing component to change fonts. font-sizes, text and paragraph styles, colours and lists.

Make any I-grain visible by clicking its name in the navigation bar on the left. Built-in I-grains that can't be deleted appear at the bottom of the list and all begin with 'DotWikIE'. Use the **Show All** and **Hide All** menu items to show or hide all I-grains.

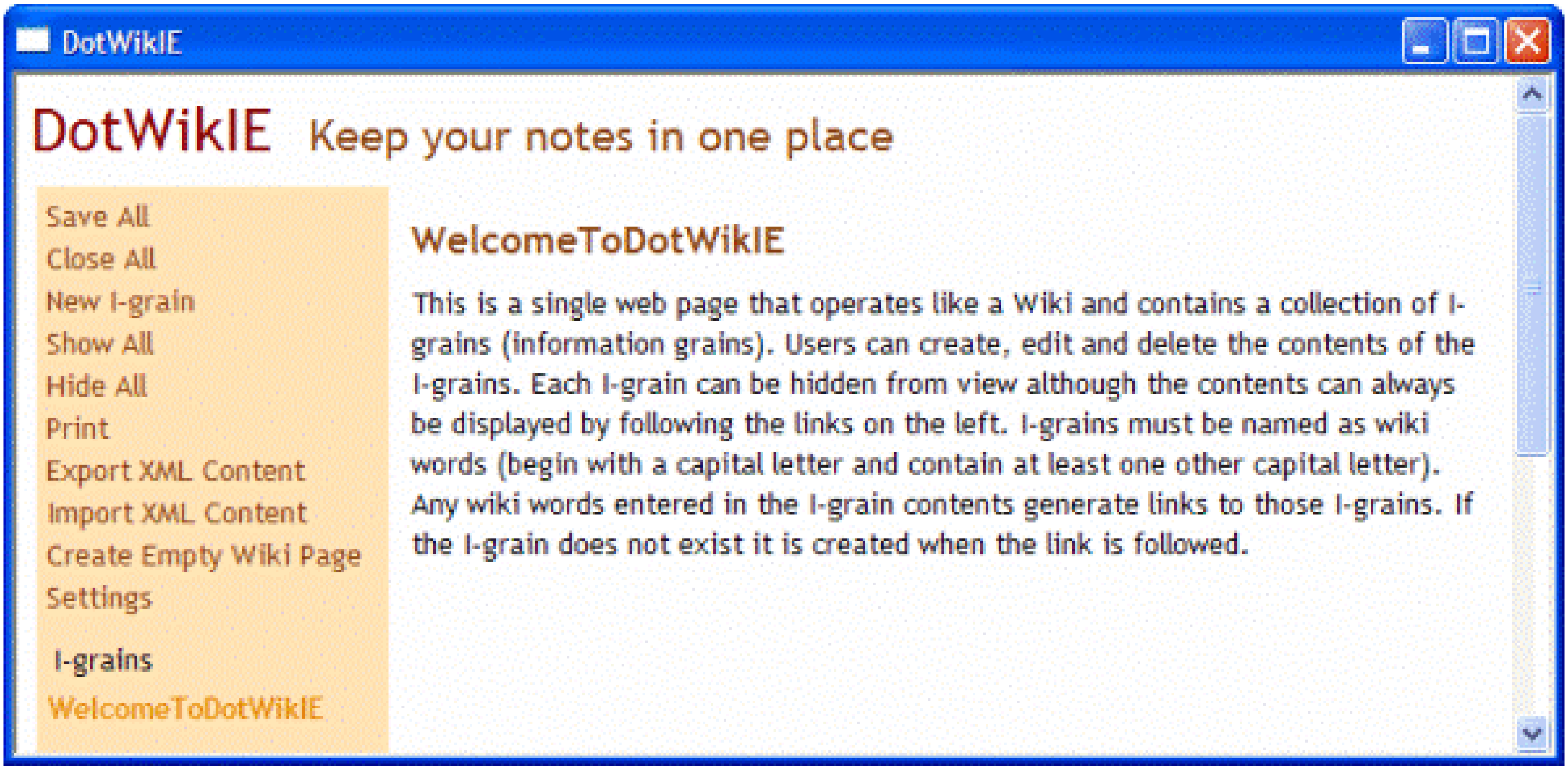
Figure 5. DotWikIE Settings.

The 'Settings' menu allows the user to specify a wiki title and subtitle and the author. The wiki title also becomes the page title so if several wiki pages are open the task bar is a convenient way to navigate between them. Versions and clippings are mentioned later.

Another totally benficial by-product of limiting DotWikIE to Internet Explorer is the trivial conversion of a wiki page to a Microsoft Hypertext Application (HTA). This is achieved by changing the file extension to .hta and the optional insertion of the following special tag in the page head:

```
<HTA:APPLICATION id=HTADotWikIE navigable="yes" />
```

The resulting DotWikiIE HTA file is treated like a .exe program file. Double clicking the file for example will execute the application. Run in this mode DotWikIE is freed from the sandbox restrictions that bind the browser. In the case of DotWikIE this eliminates the user prompts on startup and first save. As Figure 6 shows the Internet Explorer toolbars and menus are also eliminated to give a much cleaner user interface. The operation is otherwise identical.

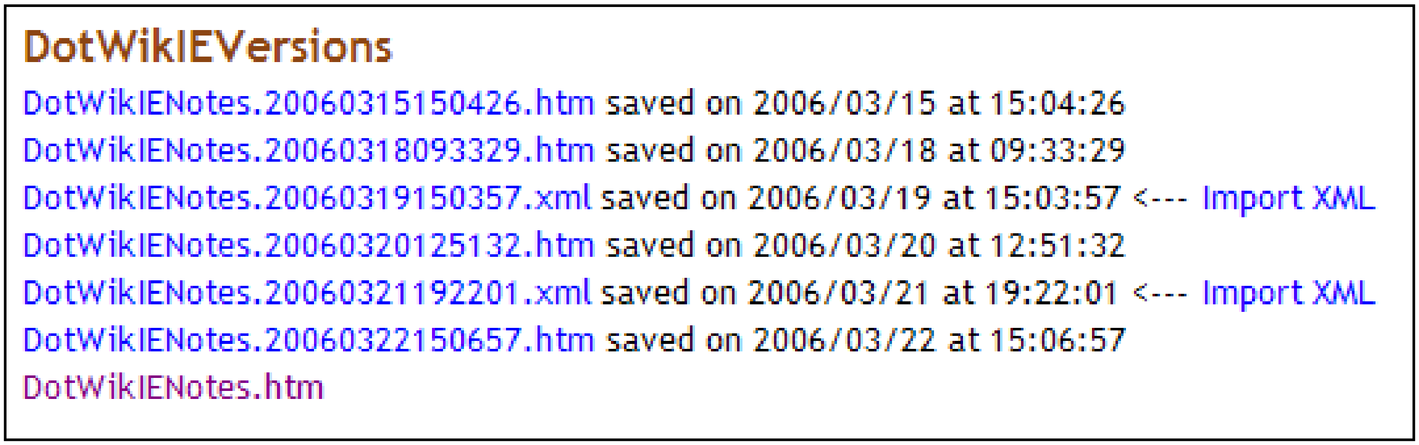


WelcomeToDotWikIE

This is a single web page that operates like a Wiki and contains a collection of I-grains (information grains). Users can create, edit and delete the contents of the I-grains. Each I-grain can be hidden from view although the contents can always be displayed by following the links on the left. I-grains must be named as wiki words (begin with a capital letter and contain at least one other capital letter). Any wiki words entered in the I-grain contents generate links to those I-grains. If the I-grain does not exist it is created when the link is followed.

Figure 6. The DotWikIE Hypertext Application.

When the 'Versions' checkbox is checked then when the user clicks the 'Save All' menu DotWikIE saves the current version of the page before creating the updated page. All previous versions are also recorded in the special 'DotWikIEVersions' I-grain. Similarly when the 'Export XML Content' menu is clicked to save the XML contents a record is maintained in the same 'DotWikIEVersions' I-grain. When this I-grain is viewed the existing versions and XML documents are displayed as shown in Figure 7. Clicking on the wiki version page links (.htm) loads that previous version while clicking on a 'Import XML' link will import the I-grains stored in the associated XML document.



DotWikIEVersions

DotWikIENotes.20060315150426.htm saved on 2006/03/15 at 15:04:26
DotWikIENotes.20060318093329.htm saved on 2006/03/18 at 09:33:29
DotWikIENotes.20060319150357.xml saved on 2006/03/19 at 15:03:57 <--- Import XML
DotWikIENotes.20060320125132.htm saved on 2006/03/20 at 12:51:32
DotWikIENotes.20060321192201.xml saved on 2006/03/21 at 19:22:01 <--- Import XML
DotWikIENotes.20060322150657.htm saved on 2006/03/22 at 15:06:57
DotWikIENotes.htm

Figure 7. DotWikIE Versions and XML Import.

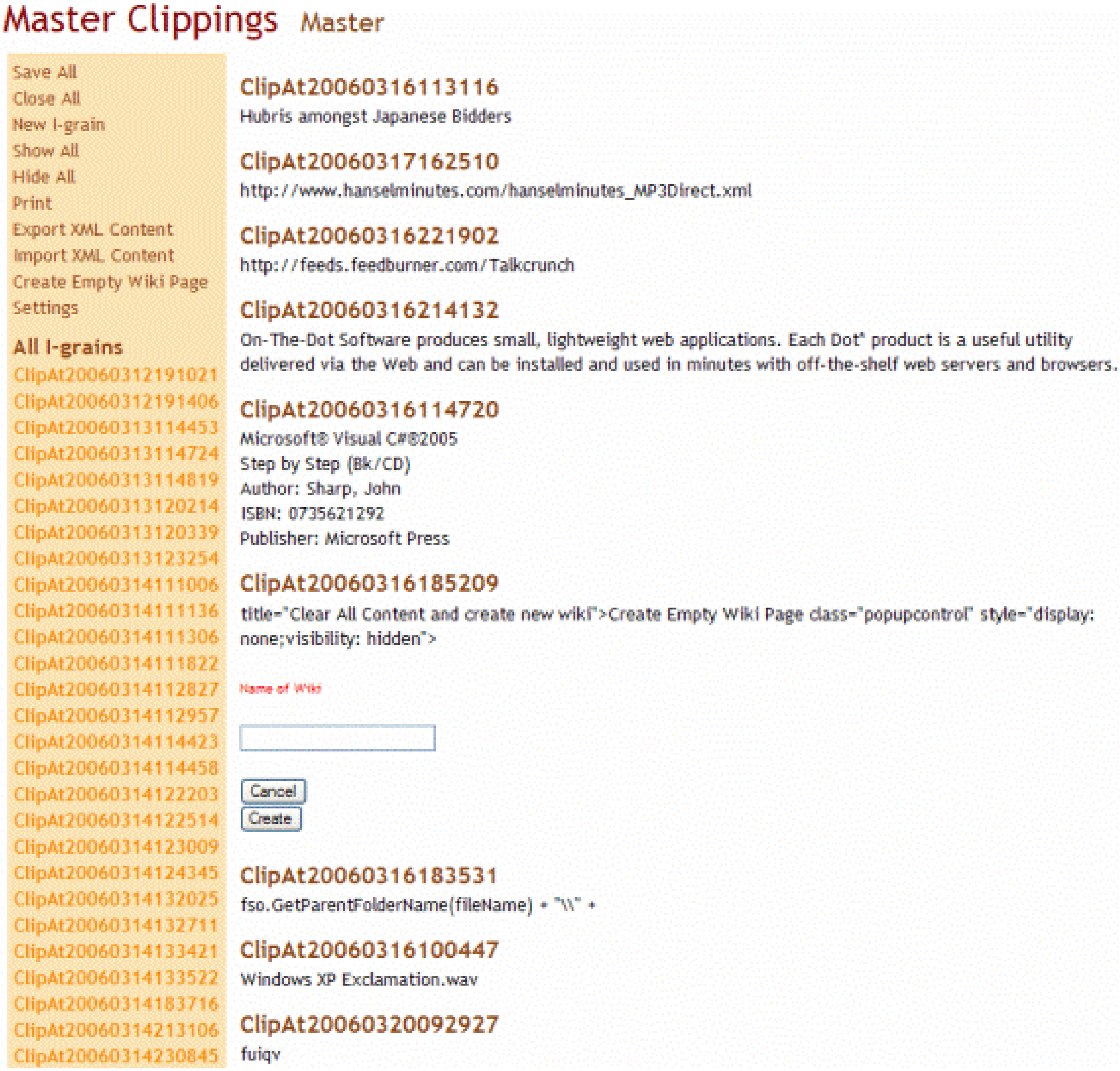
The 'Scripting.FileSystemObject' ActiveX control is not available to HTML pages loaded via a web server. This means if a DotWikIE page is placed on a web site and accessed in the browser all features will work correctly except for saving updated content or XML. Users essentially receive a read-only experience. This apparent downside can actually be an advantage for building simple web sites made up of DotWikIE pages simply uploaded to a web server. The site author can create and edit the DotWikIE pages locally then upload them to the production web site. DotWikIE becomes a very simple web site building mechanism without the need to know HTML. (As mentioned previously this is how the TiddlyWiki web site is built.)

Automatic Capture of Wiki Fragments

Readers who value simple hints, now more commonly called life hacks, on time management and task organisation will be familiar with the 43Folders website maintained by Merlin Mann [HREF17] who is also a proponent of GTD. Mann talks of a simple Mac-based application that he uses to build a structured set of files to store text fragments in an organised fashion. He has written macros that allow a piece of text to be selected from an information source such as email, web page, calendar entry or any other application, and with a special hotkey the text can be saved into the text filing system for later retrieval and use. I-grains are typically small fragments of text. The author postulated that the hotkey, Ctrl-C, available in all Windows applications, could copy the interesting text to clipboard. What is needed is a background process to save the clipboard text into DotWikIE as an I-grain. Fortunately the Internet Explorer HTML DOM is extended with the clipboardData object which allows client-side JavaScript to access (and set) the clipboard text.

A few lines of JavaScript can activate an interval timer that triggers a check for the clipboard text contents. If the text has changed from the last check then the new contents can be added as a time stamped I-grain. Such a modification is added to DotWikIE and activated by checking the 'Clippings' checkbox. A time interval of 5 seconds is used. That DotWikIE clippings page becomes a clipboard scavenger and can be left open in the browser to run in the background as the user continues to use the machine in the normal way. Every few hours or at other suitable intervals the 'Save All' menu will save the clippings I-grains which can also be exported to XML if desired. This equates to running a clipboard scavenger in a passive mode where only clippings intended for copying and pasting are captured. Mann's capture of text selections with his hotkey equates to an active mode where the capture is made with the intention of storage and later processing. With the DotWikIE clipboard scavenger this can be achieved at any time by consciously copying any text to the clipboard, perhaps without the intention of pasting the text to a destination. This active use of Ctrl-C knowing the text will be scavenged will probably lead to more useful text fragments being recorded. To date the author has not experimented with this approach but future development work will be carried out with this active approach.

DotWikIE clipboard scavengers were run on three of the author's machines (home desktop, work desktop and laptop used to develop DotWikIE) for about 10 days prior to submitting this paper. A very simple clipboard scavenger log processing utility, ClipLog, was written as JavaScript in a web page to merge clippings logs and eliminate any duplicates as various versions of DotWikIE were developed. In total 179 clippings were captured at an average size of 235 characters. Note that non-textual clipboard contents are ignored such as the copy and paste of images into this paper, for example. The accumulated master log file generated by ClipLog is in the same XML format used for I-grains. This allows the master log to be imported into its own wiki page and the individual clippings I-grains to be manually inspected. Figure 8 shows a representative sample of the clippings wiki with each I-grain title acting as a timestamp. Importing and rendering 179 I-grains introduced took much longer than expected and points to a need to optimise the JavaScript routines in DotWikIE.



ClipAt20060316113116

Hubris amongst Japanese Bidders

ClipAt20060317162510

http://www.hanselminutes.com/hanselminutes_MP3Direct.xml

ClipAt20060316221902

http://feeds.feedburner.com/Talkcrunch

ClipAt20060316214132

On-The-Dot Software produces small, lightweight web applications. Each Dot" product is a useful utility delivered via the Web and can be installed and used in minutes with off-the-shelf web servers and browsers.

ClipAt20060316114720

Microsoft® Visual C#®2005
Step by Step (Bk/CD)
Author: Sharp, John
ISBN: 0735621292
Publisher: Microsoft Press

ClipAt20060316185209

title="Clear All Content and create new wiki">Create Empty Wiki Page class="popupcontrol" style="display: none;visibility: hidden">

ClipAt20060316183531

fso.GetParentFolderName(fileName) + "\\" +

ClipAt20060316100447

Windows XP Exclamation.wav

ClipAt20060320092927

fuiqv

Figure 8. Samples of the author's clippings.

Manual inspection of the clippings wiki easily conveys a record of the many of the author's activities over the ten-day period. The clipping examples represent:

- a seminar title
- a URL for an XML document
- an RSS feed URL
- a product description
- a book reference
- a piece of HTML that is rendered in DotWikIE
- a JavaScript fragment

- a file name
- a password from a confirmation email (hurriedly changed)

These are just a small sample of the information types that are likely to pass through the clipboard. Readers will see the scope for post-processing of this rich information. One can imagine an intelligent processor that identifies URLs, email addresses, HTML and XML fragments and so on. A formatted clippings journal with associated indexes and lists of different information types will form a valuable record of one's activity. Using active copying and more intelligence at the time of capture will also considerably add to the utility of the clippings information.

Definitely not shown in Figure 8 are some very sensitive clippings that appeared in the collection. These were a copy of the author's payslip that is emailed monthly! This information is copied to special files for printing. Another clipping I-grain was the author's credit card number copied to an online shopping checkout page. This highlights the intensely private nature of the clipboard contents and shows a clipboard scavenger shares some of the dangers of a keystroke logger.

Using the DotWikIE clipboard scavenger is only a tiny attempt to record one computing activity. However in the limited experience so far it is sufficient to trigger memory over a period of days, enough to help trigger blog entries and time activity recording. The clipboard scavenger in no way compares to the Microsoft MyLifeBits project (Gemmell et al, 2006) where every activity captured in audio, video stills and movies, all documents, emails. movement maps and so on is accumulated and post-processed into a visually stunning record of one's entire life.

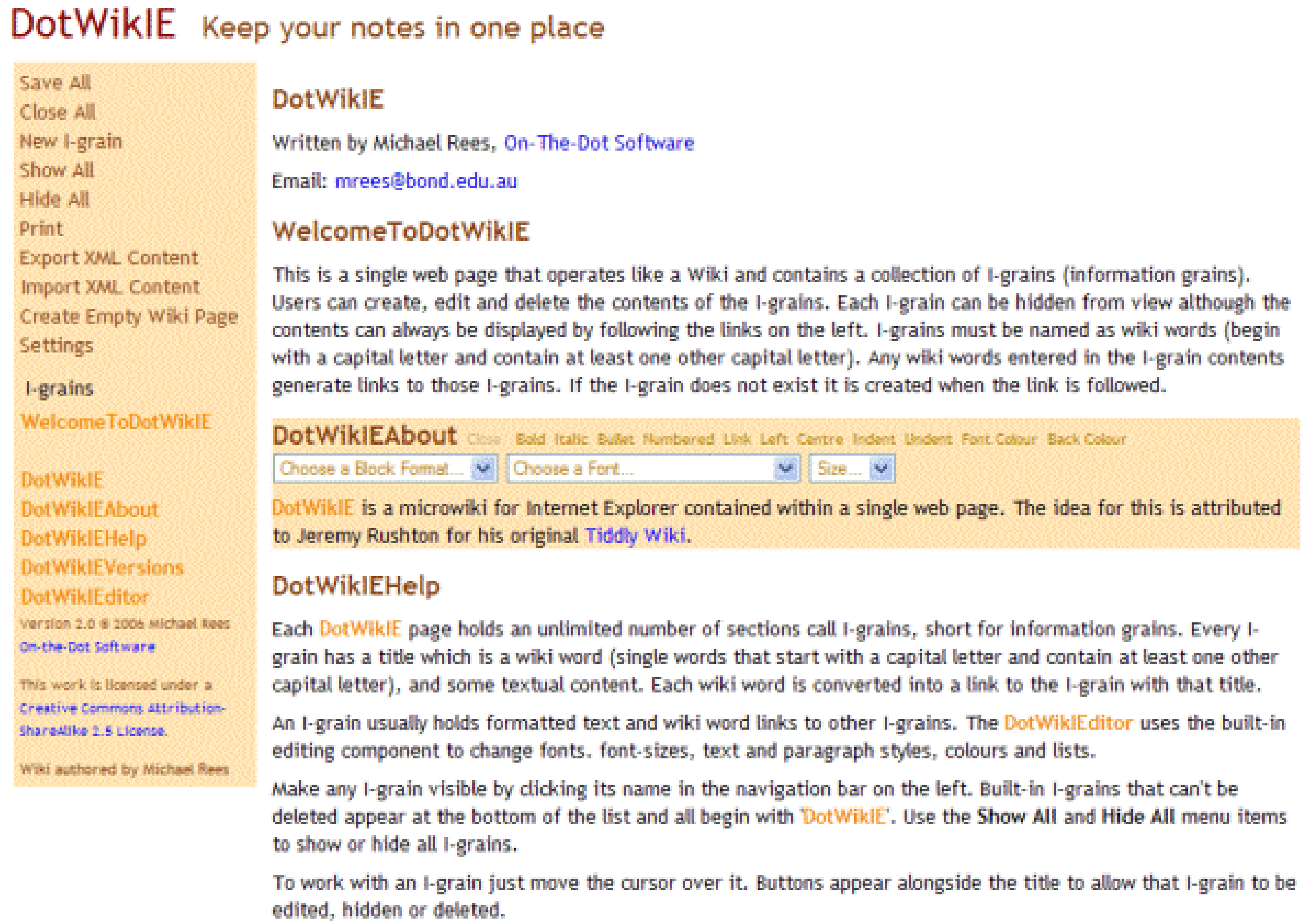
Web Application Version of DotWikIE

It has been seen that DotWikIE pages can be placed on a web site. Served in such a way the wiki becomes read-only and a passable simple web page with consistent internal navigation. Multiple wiki pages can be used with relative hyperlinks between them. Obviously there would be increased utility if the wiki pages could be read/write just as they are when accessed from the local filestore. The next extension to DotWikIE investigated how web-based wiki pages could be updated when viewed via the browser. This of course took the implementation into the very areas that the author's existing DotSoft tools already cover, the DotTegular tool in particular (Rees, 2005). However the DotSoft tools all consist of multiple web server files for scripts, images, XML documents and the HTML pages themselves.

The challenge then was to attempt to provide a single-page version of DotWikIE renamed to DotWikIEWeb that can update its own contents when accessed on a web server. Some server-side script execution is needed to read and write files on the server. In addition the folder containing the files to be updated must have Write and/or Modify permission set for the web server process account. The author is most familiar with ASP pages containing JavaScript server-side scripts running on the IIS web server.

Using this design approach a single file, DotWikIEWeb.asp, can be uploaded to a folder on the IIS web server and the folder given suitable write permissions. On first access, server-side script in the DotWikIEWeb page checks for the existence of a simple document management page called DotWikIEWebControl.asp. If this page does not exist it is created and populated with about 50 lines of server-side JavaScript. DotWikIEWeb pages, when accessed in the browser, then use client-side JavaScript to issue HTTP requests to this control page using the time-honoured Ajax XMLHTTPRequest ActiveX object. The server-side control script generates listings of existing pages in the folder, and reads and writes the contents of XML and HTML documents.

Thus a DotWikIEWeb user uploads a single page that spawns the ASP control page on first access. Subsequently the HTTP requests issued in the background from the browser take care of writing and updates to the wiki pages and XML contents. When run on a web server the page looks very little different from DotWikIE except for the 'Pages' button with a light blue background alongside the 'I-grains' button. Clicking the 'Pages' button hides the I-grains and shows a list of other wiki pages in the folder as shown in Figure 9.



Version 2.0 © 2006 Michael Rees
On-the-Dot Software

This work is licensed under a Creative Commons Attribution-ShareAlike 2.5 License.

Wiki authored by Michael Rees

DotWikIE

Written by Michael Rees, On-The-Dot Software

Email: mrees@bond.edu.au

WelcomeToDotWikIE

This is a single web page that operates like a Wiki and contains a collection of I-grains (information grains). Users can create, edit and delete the contents of the I-grains. Each I-grain can be hidden from view although the contents can always be displayed by following the links on the left. I-grains must be named as wiki words (begin with a capital letter and contain at least one other capital letter). Any wiki words entered in the I-grain contents generate links to those I-grains. If the I-grain does not exist it is created when the link is followed.

DotWikIEAbout

DotWikIE is a microwiki for Internet Explorer contained within a single web page. The idea for this is attributed to Jeremy Rushton for his original Tiddly Wiki.

DotWikIEHelp

Each DotWikIE page holds an unlimited number of sections call I-grains, short for information grains. Every I-grain has a title which is a wiki word (single words that start with a capital letter and contain at least one other capital letter), and some textual content. Each wiki word is converted into a link to the I-grain with that title.

An I-grain usually holds formatted text and wiki word links to other I-grains. The DotWikIEditor uses the built-in editing component to change fonts, font-sizes, text and paragraph styles, colours and lists.

Make any I-grain visible by clicking its name in the navigation bar on the left. Built-in I-grains that can't be deleted appear at the bottom of the list and all begin with 'DotWikIE'. Use the **Show All** and **Hide All** menu items to show or hide all I-grains.

To work with an I-grain just move the cursor over it. Buttons appear alongside the title to allow that I-grain to be edited, hidden or deleted.

Figure 9. DotWikIEWeb Extra Feature.

The page names are simple hyperlinks to the individual wiki pages so the user can easily switch between them. New pages are created as before by using the 'Create Empty Wiki Page' menu item. These newly created pages need only have the .htm prefix as they have no need to execute server-side script as this function is taken over by the control ASP page. Saving, versioning and XML document manipulation operates in the same way as for DotWikIE. Clicking on the 'I-grains' button switches to listing the I-grains in the current wiki page.

While the user interface of DotWikIEWeb and DotWikIE are very similar and interact in the same way the changes to the underlying JavaScript are much more significant. All file manipulation in DotWikIE with the 'Scripting.FileSystemObject' has to be replaced with HTTP requests using 'MSXML. XMLHTTPREQUEST'. Scripting to do both operations is included in each wiki page which senses the environment, local file or web page, and uses the appropriate script. The JavaScript in DotWikIEWeb is about 950 lines and occupies 35 Kb of the wiki. When the embedded CSS and XSLT style sheets and the skeleton HTML are added the 'engine room' of each wiki page takes up about 47 Kb.

Obviously DotWikIE is a single-user ultra lightweight RIA intended for personal use. Once DotWikIEWeb wiki pages are served from a web server they become accessible by any user with access to that web site, potentially the whole Internet. It is possible for two or more users to conflict in their updates to an individual wiki page. However, it is envisaged that even on the web users will only access their own pages. This means that a single set of personal wiki pages can be accessible via the web rather being carried around on a USB drive. If the conflicting update problem is too severe it will be simple matter to implement checking script as used in DotTegular that warns users of the conflict and allows a save to be aborted and the new version retrieved before saving.

As a single web page DotWikIEWeb employs the ultra lightweight RIA implementation approach to provide a very usable and practical personal web-based wiki. Although delivered as a single file DotWikIEWeb does break the strict ultra lightweight RIA definition as it must spawn a control ASP file before it can function properly. Nevertheless the result is still an extremely lightweight solution. If the 47 Kb per wiki page overhead is thought excessive then a fairly simple further extension would be to write the JavaScript, XML content and XSLT style sheet to separate files and thus avoid the duplication. The resulting wiki would then consist of a whole collection of support files that would make migration from one web site to another more complex. The current DotWikIEWeb solution allows just the original DotWikIEWeb.asp file and other wiki content pages to be copied to a new web site in order to function correctly.

Conclusions and Future Work

The building of DotWikIE as an ultra lightweight RIA has shown the significant potential of this web application implementation technique. The author has shown that in-place WYSIWYG editing of HTML content and the use of XML in-page storage and manipulation are practical using Internet Explorer from within a single web page file. DotWikIE happens to be a viable wiki for I-grains containing useful HTML content. The ultra lightweight RIA technique can be applied to a whole range of other types of applications that display, manage and store HTML/XML information within a single page.

Being able to turn DotWikIE into an executable application by adding a single <HTA> tag and changing the extension to .hta is a very happy by-product of limiting the browser to Internet Explorer. DotWikIE. hta exposes a much cleaner and simpler user interface from the ultra lightweight RIA user's point of view. The fact that the Internet Explorer HTML rendering engine and Javascript interpreter are being used is completely hidden from the user. Most browser users tend to have several pages open at one time even if page tabs are available. In the browser it is easy to accidentally close the DotWikIE page when intending to keep it open for further editing. Running the ultra lightweight RIA as free-standing application makes this problem less likely.

An XML document is said to be a self-defining structure with embedded data. An ultra lightweight RIA can be said to be a self-managing data document. The management engine and data co-exist within the same file. Surely this can lead to a plethora of useful types of applications. A speculative list of possible applications mentioned above includes word processors, blogs, spreadsheets, slide show managers, simple databases, calendars, project managers and all types of personal productivity aids. All these can be implemented using the browser and local HTML files.

Moving an ultra lightweight RIA to a web site as a single file proves to be more difficult although not impossible. The single file needs to spawn at least one additional file containing server-side script to control the management of the web server's files. Nevertheless a very compact solution is still possible with the added benefit of universal access to a single wiki repository. Additional problems of conflicting simultaneous changes must be solved to make web-based ultra lightweight RIAs like DotWikIEWeb into a robust solution.

An obvious extension to DotWikIEWeb is the addition of an RSS feed for each wiki page. Embryonic RSS support script is actually built in to DotWikIEWeb but is not in a fully developed form. RSS in TiddlyWiki visible in Figure 1 relies on the user entering the URL of the web site folder into a special tiddler. This is used by the RSS generation script to insert correct URLs for the RSS aggregators that visit the wiki page when it is resident on a web site. In the case of DotWikIEWeb correct URLs can be generated automatically by the ASP control page by scripting suitable extensions, a task set for the near future.

Feature additions to DotWikIEWeb are still needed to match the facilities of TiddlyWiki. The most obvious omission is a search mechanism. In the case of Internet Explorer with its built-in XPath processor the searching of the I-grain XML is very straightforward and reasonably efficient. Adding the very popular tagging mechanism to allow the I-grains to be easily partitioned for browsing and searching needs non-negligible additional scripting and some redesign of the user interface layout. Presenting I-grains in a number of other orders particularly in chronological order would make DotWikIEWeb more attractive to diary, blog and journal applications, for example.

Another promising line of investigation left for the future involves the exploitation of the growing number of flexible storage web services. Google's rumoured GDrive promises to be one such service. One of the better fully announced service is Amazon's S3 [HREF18]. Via web service REST and/or SOAP calls sent using XMLHTTPRequest S3 allows script to read, write and delete objects from 1 byte to 5 Gigabytes of data each. The number of objects is unlimited and access is controlled using unique developer keys. This is not a free service but costs only a few cents per Gigabyte-month of storage and a similar amount per Gigabyte transferred. With a service like S3 to provide massive, structured online storage then very much more powerful ultra lightweight RIAs become possible. Relatively small bodies of JavaScript in a single HTML page can provide managed access to huge repositories of information. The range of possibilities appear immense. When one adds the growing range of other web services for mapping, bookmarking, searching and so on the opportunities for large numbers of powerful ultra lightweight RIAs would seem to have no bounds.

Acknowledgement

I have already conveyed to Jeremy Ruston, author of TiddlyWiki, my thanks for his brilliantly simple idea of a wiki in a single page. This truly inspired the work described in this paper and I would like to formally record my gratitude here.

References

Désilets, A., Paquet, S., and Vinson, N. G. (2005). Are wikis usable?. In Proceedings of the 2005 international Symposium on Wikis (San Diego, California, October 16 - 18, 2005). WikiSym '05. ACM Press, New York, NY, 3-15.

Gemmell, J. , Bell, G. and Lueder, R. (2006). MyLifeBits: a personal database for everything, Communications of the ACM, vol. 49, Issue 1 , pp. 88-95.

Rees, M. J. (2000). Implementing Responsive Lightweight In-Page Editing. AusWeb2K - The Sixth Australian World Wide Web Conference, Rihga Colonial Club Resort, Cairns.

Rees, M. J. (2004). User Interfaces for Search Information Management Web Sites. AusWeb2004 - The Tenth Australasian World Wide Web Conference, Gold Coast, Australia.

Rees, M. J. (2005). Supporting Highly-Interactive Web Applications with Server-Side Code. AusWeb2005 - The Eleventh Australasian World Wide Web Conference, Gold Coast, Australia.

Hypertext References

HREF1 Dr Michael J Rees
http://www.bond.edu.au/it/staff/michael.htm
HREF2 School of IT, Bond University
http://www.bond.edu.au/it/
HREF3 Bond University
http://www.bond.edu.au/
HREF4 TiddlyWiki
http://www.tiddlywiki.com/
HREF5 Wiki defined in Wikipedia
http://en.wikipedia.org/wiki/Wiki
HREF6 Wikipedia English Edition
http://en.wikipedia.org/wiki/Main_Page
HREF7 Wikipedia Screencast
http://weblog.infoworld.com/udell/gems/wikiAnimate.html
HREF8 List of Wiki Software
http://en.wikipedia.org/wiki/List_of_wiki_software
HREF9 List of Wiki Farms
http://c2.com/cgi/wiki?WikiFarms
HREF10 JotSpot
http://www.jot.com/
HREF11 TiddlyWiki adaptations
http://www.tiddlywiki.com/#TiddlyWikiAdaptations
HREF12 GTDTiddlyWiki
http://shared.snapgrid.com/gtd_tiddlywiki.html
HREF13 Getting Things Done
http://www.davidco.com/
HREF14 TrimPath Junction
http://trimpath.com/
HREF15 'Take It With You' Wiki
http://ajaxian.com/archives/take-it-with-you-wiki
HREF16 On-The-Dot Software
http://comet.it.bond.edu.au/dotsoft
HREF17 43Folders
http://www.43folders.com/
HREF18 Amazon S3 Service
http://www.amazon.com/gp/browse.html/002-7638199-3479239?node=16427261

Copyright