

Widgets as Personalised Mini-Portals

Con Nikakis

Victoria University, Australia

INTRODUCTION

Wikipedia (Wikipedia, 2006) describes the term widget as a "... general purpose term, or placeholder name, for any unspecified device, including those that have not yet been invented" (Widgets, 2006) with the origin of the term attributed to the 1924 play "Beggars on Horseback," by George Kaufman and Marc Connelly (Kaufman & Connelly, 1924), where it was used to describe a product manufactured by one of the characters.

In the mid-1990s, graphical user interface (GUI) programmers cheekily used this terminology for their technical description of GUI components with which a user would interact, usually when the components would launch a small helper application (Myers, 1996). More specifically, in the MAC OS desktop, widgets were designed as small specialised visual accessories, such as clocks and calendars, with the most recent including weather and flight information (Miller, 2005). The clean, crisp, graphical design of these accessories has captured the attention of many MAC devotees.

Web programmers took this concept further when first on the MAC OS, and then on Windows platforms, widgets were used as examples of personal, customisable portals with a "gadget" feel about them that appealed to the fun or "geeky" side of a computer user's nature (Udell, 2004). They are often created with quality graphics to attract the attention of a potential user and are easy to use, operating independently of a browser by linking directly to a Web application (Frakes, 2005).

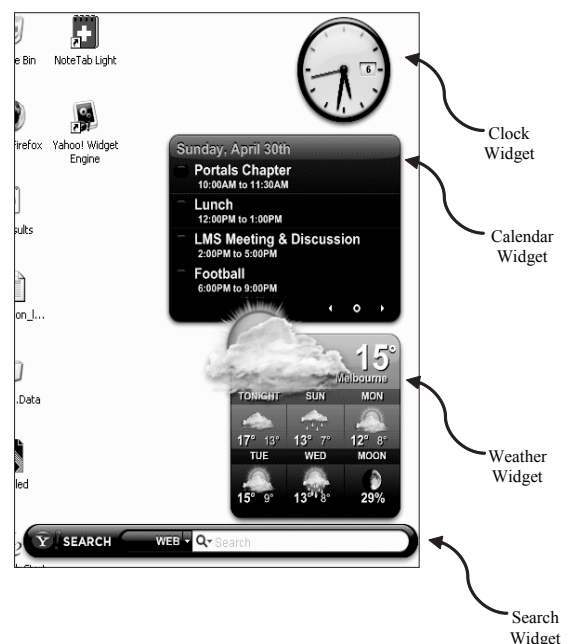
A widget is not the first or only technology that exhibits these characteristics. In many ways, a widget can be seen as a "portlet" or miniportal. As manifestations of personalised Web pages, portals have been part of the World Wide Web scene since the early days of the browser. A portal may have been seen as nothing more than a specialised Web site, but with the advent of Web commercialisation, their popularity has mushroomed (Gunther, 2001). The simple portal has now developed into a myriad of application types; a discipline in its own right?

Tatnall (2006) categorises portals as nine main types; general, vertical industry, horizontal industry, community, enterprise information, e-marketplace, personal/mobile, information, special/niche, with widgets as hybrid applications seeming to fit best under the last three. So why this categorisation? This question is probably best answered by first looking at the characteristics of a widget.

BACKGROUND: WHAT IS A WIDGET?

There are many widget definitions, Udel's (2004) "special class of small, single-purpose applications," Taylor's GUI toolkits (Taylor, Medvidovic, & Anderson, 1996), Smith's (2005) technical explanation "... at its simplest, a widget consists of four things: an image, a preference file, an HTML file, and a folder," and Cartwright's (Cartwright & Valentine, 2002) description of a flexible desktop means providing user interaction that increases Internet usage. Howard's (2005) simple description of widgets as dynamic, instantly accessible information providers revealing the market driven nature of these miniportals, is particularly appealing. After all, widgets, like many other Internet-based applications, live or die on their marketability. In other words, the personalization, item specific, dynamic, customizable, free, large library characteristics of these miniportal applications are much more in tune with a user centric definition than a technically orientated definition that belies the importance of the psychological aspects that widgets endear with their use.

Figure 1. Sample widget desktop (Source: Widget Gallery Yahoo.com)



WIDGET TECHNOLOGY

At its simplest, a widget consists of a portable network graphics (PNG) file, a preference file, an HTML file, and a folder with a name ending in .wdgt (Smith, 2005). In other words, a mix of code and graphics organised into a bundle consisting of a “contents” folder that encapsulates the required files. For instance, Yahoo’s Konfabulator (Yahoo.com, 2006) describes a .kon file, a folder in which images are kept, and sometimes one or more .js files, as well as an Info.plist file. If the widget was developed on a Mac, a .spt file might also be found.

Using Konfabulator (Joyce, 2005) as a representative example, a brief breakdown of the file types are:

- **.kon:** Contains the main code for a widget. Konfabulator reads instructions from this file first. The code is written as XML (eXtensible Markup Language) for initial image positioning and referencing. It can also contain most of the code that makes the widget function (JS or JavaScript). On more complex widgets, the JavaScript is usually stored in a separate .js file.
- **.jsIs:** Pure javaScript containing most of the JavaScript needed to make a widget operate.
- **.Info.plist** An XML file accessed by Mac OS X to find out version information. This file is ignored by Windows.
- **.spt:** An AppleScript document, containing AppleScript commands with the Widget only able to run on a Mac.

Figure 2. Example HTML code from a .kon file (Source: Widget Creation Tutorial.pdf Yahoo.com)

Make the following changes to "My First Widget.kon" and save it, then click "Reload Widget" in the debug window.

```

<?xml version="1.0" encoding="UTF-8"?>
<widget>
  <debug-on/>
  <window>
    <name>main_window</name>
    <title>My First Widget</title>
    <height>38</height>
    <width>300</width>
    <visible>true</visible>
  </window>
  <text>
    <name>myText</name>
    <color>#FF0000</color>
    <size>18</size>
    <align>left</align>
    <vOffset>25</vOffset>
    <hOffset>2</hOffset>
  </text>
  <timer>
    <name>timer</name>
    <interval>1</interval>
    <tickin>true</tickin>
    <onTimerFired>
      <!--
      var cpuLoad = system.cpu.activity;
      myText.data = cpuLoad + "% CPU load";
      if (cpuLoad < 40)
      {
        myText.opacity = 102;
      }
      else
      {
        myText.opacity = cpuLoad * 2.55;
      }
      //-->
    </onTimerFired>
  </timer>
</widget>

```

Once the functionality of the widget is created for real-world distribution, the images needed to present the user interface are added along with the .kon file: images that have been carefully prepared and edited using an image editor such as Photoshop.

Distribution of the widget as just a folder of files makes the application difficult to use. Proper packaging to make it look like a single file is achieved by putting all the images, the .kon file, and anything else that may apply to the widget inside a “Contents” folder. The Contents folder is then inserted inside another folder with a widget name, such as “Widget Wonder,” adding the .widget extension to the end of the file name if a Mac is being used, or using the widget Converter that packages widgets in the Windows widget format (Yahoo.com, 2006). The widget package then is in a form that can be submitted to a site, such as the Widget Gallery (<http://www.widgetgallery.com/>).

The Widget Gallery is typically a library of widget creations that exhibit attractiveness, serve a useful purpose, and have some unique features (Phelps, 2005).

WIDGET LIBRARY

There are now thousands of widget applications categorised into widgets that are used for fun and games, date and time, news feeds, system utilities, sight and sound, geek stuff, cam viewers, widget tools, application enhancers, and search tools (Smith, 2005).

Widgets also have a serious side. Snell reports on widget dictionary use (Snell, 2005); Spanbauer (2006) describes a widget online directory; image search technology is produced by Schwartz and Gormley (Schwartz & Gormley, 2005); Powell (2006) uses a WebTV widget; and Myers (1996) comments on serious user interface technology. Miller et al. also describe customised learning tools for the disabled (Miller, Brown, & Robinson, 2002); Mace (1996) discusses the use of widget Web graphing tools; Howard (2005) presents the usage of audio-conferencing widgets; Girgensohn describes database interfacing form widgets (Girgensohn & Lee, 1997); and Deaves (Deaves & Sharma, 2003) and Brown illustrate the usage of widget-based educational software in the classroom (Brown & Miller, 2002). These examples illustrate just a few of the thousands of serious applications that use the widget as an interface into serious applications.

WIDGETS: A BIT OF FUN OR USEFUL TOOL?

The history of Web development is littered with the “jazzing” up of well-established Web technologies that are then

2 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/widgets-personalised-mini-portals/18030

Related Content

A Cloud Portal Architecture for Large-Scale Application Services

Jun-Jang Jeng, Ajay Mohindra, Jeaha Yang and Henry Chang (2010). *International Journal of Web Portals* (pp. 7-21).

www.irma-international.org/article/cloud-portal-architecture-large-scale/40315

Web Portals for Financial Analytics: How Effective Are They from the End-Users' Perspective

Alexander Y. Yap (2013). *International Journal of Web Portals* (pp. 1-18).

www.irma-international.org/article/web-portals-for-financial-analytics/101801

Competitive Dynamics of General Portals

Sandra Sieber and Josep Valor-Sabatier (2005). *Web Portals: The New Gateways to Internet Information and Services* (pp. 64-79).

www.irma-international.org/chapter/competitive-dynamics-general-portals/31170

Adaptive Web Services Monitoring in Cloud Environments

Yi Wei and M. Brian Blake (2013). *International Journal of Web Portals* (pp. 15-27).

www.irma-international.org/article/adaptive-web-services-monitoring-cloud/78350

Using the Building Blocks: Evolution of a Portal Suite

Joe Lamantia (2012). *Enhancing Enterprise and Service-Oriented Architectures with Advanced Web Portal Technologies* (pp. 259-271).

www.irma-international.org/chapter/using-building-blocks/63961