

# Designing and Developing a Portal for the Polar Grid High Performance Computing System

Patrina Bly (ECSU), Justin Deloatch (ECSU), Jonathan Henderson (MVSU) and Camden Hearn (MVSU)  
Undergraduate Research Experience in Ocean, Marine, and Polar Science - Elizabeth City State University  
1704 Weeksville Road, Box 672, Elizabeth City, North Carolina 27909

**Abstract** - Polar Grid is a National Science Foundation (NSF) Major Research Instrumentation (MRI) program funded partnership of Indiana University (IU) and Elizabeth City State University (ECSU) to acquire and deploy the computing infrastructure needed to investigate the urgent problems in glacial melting. When fully completed the grid will be comprised of ruggedized laptops and computer clusters deployed in the field in the Polar Regions and two large scale computing clusters for detailed analysis in the U.S. – one to be installed at IU and the other at ECSU. This installation will give ECSU a 5 Teraflop Minority Serving Institution High Performance Computing (MSI HPC) system, building on its distance education and undergraduate laboratory infrastructure to create tremendous outreach capabilities.

Accessing this computing cluster will entail the development of a grid portal that provides security, access to data, the ability to process data, and education/outreach functions. Development of this portal started with the documentation of terms, processes, and software needed to develop a portal and the understanding of what the grid is. This project then progressed to produce Extensible Markup Language (XML) page structures that could display processed data acquired during expeditions in Greenland and Antarctica. Another aspect has been the development of “iGoogle Gadgets” that simulate the portal user environment along with the process of converting that gadget into an RSS feed.

## I. INTRODUCTION

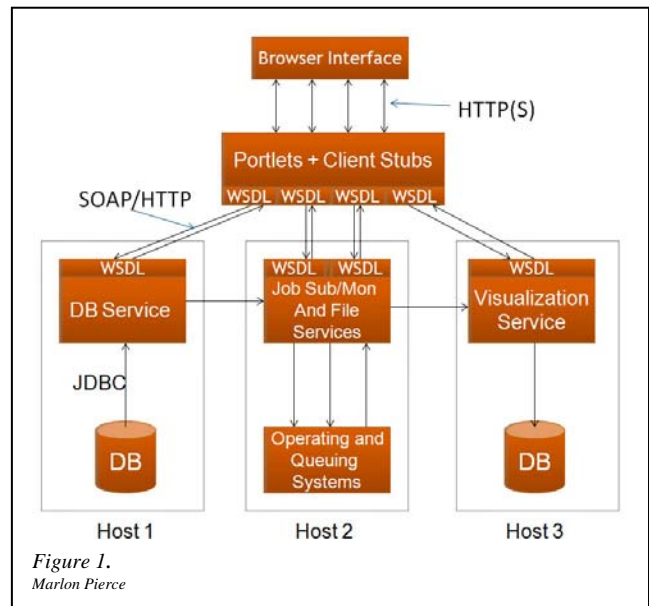
### A. Overview

The Polar Grid project is a National Science Foundation (NSF) \$1.9 million funded collaboration of Elizabeth City State University and Indiana University to gain and establish a computing infrastructure to research the critical issues in polar glacier melting. The Polar Grid project’s main goal is to help scientists better understand the current and future shape of the Earth’s melting polar ice sheets by establishing a computer grid to manage research data from Greenland and Antarctica.

A grid is a group of computers working together in various locations to collect, store, and process data. Scientists plan to use this data to determine whether or not it is usable and relay the messages back to researchers in the field as part of time management and cost effectiveness.

A portal is the median place for making all types of information accessible to an audience of varying range. They provide a single point of access to aggregated information.

Data and data processes are found within a portal. This data is accessed through the use of a portlet. An illustration of the process can be found in figure 1. A fully developed portal can take up to two years of development time. Therefore, our



focus was toward the development of portlets.

## II. TERMINOLOGY

### A. Overview

After reading several research articles and web sites related to the grid, it became very evident that the first task the team would undertake would be to learn the terminology. There are very few resources for those new to the grid and less that explain the many new terms and acronyms associated with it. The first part of understanding what we had to know was learning about the Polar Grid project. After that we had to learn what a grid is and how it works. This involved playing a game about a grid which focused on how long it takes to do a “job” at a certain location and the best type of computing system for it. The next step was to learn about how the grid is accessed which is through portals. A portal consists of a collection of portlets which provide access to data and processes on that data.

While researching these subjects, it became evident that there were many new terms that our team was unfamiliar with.

We had to define the terms used in the research articles and resources to understand the topic we were studying. This involved building an online glossary.

### B. Glossary

The glossary consists of terminology that the team encountered during the reading and researching process. These words and acronyms along with links to further information were recorded in an online glossary that will eventually be moved into a database as it continues to evolve. While there were an extensive number of terms and acronyms added, the glossary will continue to grow as the research proceeds with other teams.

### C. Source

Most of the information placed into the online glossary came from a "Portals 101" document written by Anurag Shankar, of the TeraGrid Gateways project at Indiana University. Other resources for the glossary are listed below each term and its corresponding definition. The glossary has been initially placed at <http://nia.ecsu.edu/ureomps2008/team-portal/glossary.html>.

## III. IGOOGLE/PORTAL

### A. What is a Portal?

A portal is an interface that provides access to data and processes that act upon that data. A portal can be broken down into sections called portlets. Portlets are fragments of a portal without the use of html tags such as html, header, and body. There are many examples of portals such as Orkut, TeraGrid, and iGoogle. Marlon Pierce, Assistant Director at the Community Grids Lab at Indiana University, suggested that since time was a limiting factor for our project we should not build a portal from the ground up, but instead use an existing portal in order to create portlets/gadgets. During research for our project we learned that a portlet is known as a gadget in iGoogle. In iGoogle, a container is the application or site that has the ability to run gadgets.

### B. Gadgets

Gadgets are made with a few simple pieces of code such as Extensible Markup Language (XML), Hypertext Markup Language (HTML), and JavaScript. XML is a language that structures data. Its purpose is to format data so that it can be utilized by different software applications that are able to access that data. XML is a Web Service Definition Language (WSDL) that structures data and processes to be read by clients and servers. This definition language follows the Simple Object Access Protocol (SOAP). SOAP is used to exchange XML formatted messages over networks using Hyper Text Transfer Protocol (HTTP)/Hyper Text Transfer Protocol Secure (HTTPS). HTTP is a set of rules that govern how information is sent across the Internet.

We used XML to structure our data for use by iGoogle. It

allows the developer to create their own "namespaces" where HTML uses predefined "tags". XML was also used to create Really Simple Syndication (RSS) feeds. HTML is a language used to construct web pages and JavaScript is a language used to add more interactivity to the content of a gadget.

The XML code of a gadget has three main items: content, user preferences, and gadget preferences. The content section

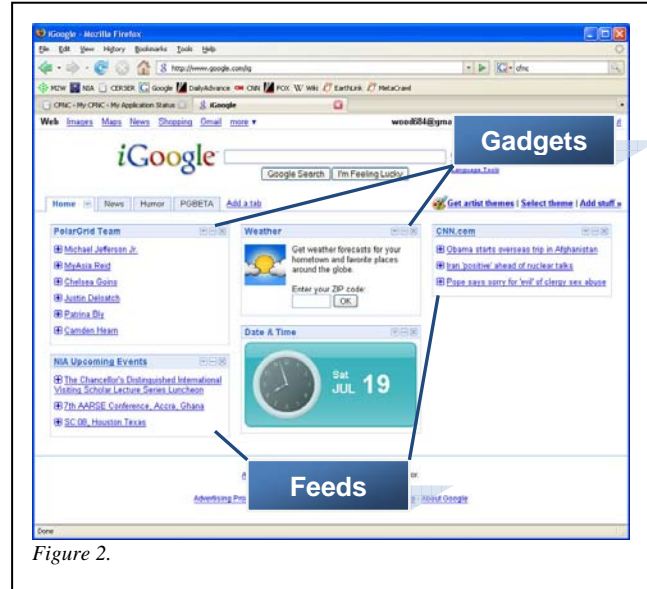


Figure 2.

is usually called the "brain" of the gadget because it is where one specifies the type of gadget, programming logic, and often the HTML elements that determine the appearance of a gadget. The user preferences section defines controls that allow users to specify settings for the gadget. The gadget preferences section specifies characteristics of the gadget, such as title, author, and preferred sizing.

The following is an example of an XML file:

```
<?xml version='1.0' encoding='utf-8'?>
  <title>PolarGrid Team</title>
  <id:tag:ECSU-NIA-2008</id>
  <updated>2008-07-15T12:12:40Z</updated>
  <author>
    <name>URE Polar Grid Portal</name>
  </author>
  <entry>
    <title>Michael Jefferson Jr.</title>
    <content type='html'>
      &lt;b&gt;2008 URE &lt;/b&gt;
      &lt;br&gt;Louisburg, NC
      &lt;br&gt;ECSU
      &lt;br&gt;
    </content>
    <georss:point>36.0990393 -78.301106</georss:point>
  </entry>
</feed>
```

### C. Constructing Gadgets

The first step to creating a gadget is to use a text editor, preferably notepad if you are using Windows operating system and text edit if you are using Macintosh operating system. Using these text editors allows for clean programming which means that the editor is not adding behind the scenes code to your document. Once you have specified the type of gadget, you should place the gadget on a public web server.

The second step to writing your own gadget is placing the gadget into a container. A container is an application or site that has the ability to run a gadget. This will ensure that your gadget will work or it will tell if your gadget does not work, in which case you should check your programming. An example of a container is iGoogle or Orkut. As a team we used iGoogle to access our gadgets.

## IV. JAVASCRIPT

### A. Overview

JavaScript is a scripting language that allows the use of dynamic content. In spite of the similarity in name to Java, it is not related to Java. Java needs to be compiled into machine code to run but JavaScript does not need to be compiled. JavaScript is interpreted by the browser eliminating the need for a compiler. JavaScript allowed us to pull information from a database and output it in XML format. The XML file is then placed in Google Maps to display the location and information associated with that entry. This is the Java Script that we used.

```

1. <%
2.   response.ContentType = "text/xml"
3.   set conn=Server.CreateObject("ADODB.Connection")
4.   conn.provider="Microsoft.Jet.OLEDB.4.0;"
5.   conn.open server.mappath("/pg08.mdb")
6.   sql="SELECT *, name FROM pg08 ORDER BY name"
7.   set rs=Conn.Execute(sql)
8.
9.   response.write("<?xml version='1.0' encoding='ISO-8859-1'?>")
10.  response.write("<xmlns:georss='http://www.geo.org/'>")
11.  response.write("<title>PolarGrid Team</title>")
12.  response.write("<id:tag:ECSU-NIA-2008</id>")
13.  response.write("<updated>08-07-15T12:12Z</updated>")
14.  response.write("<author>")
15.    response.write("<name>URE Polar Grid Portal</name>")
16.  response.write("</author>")
17.
18.  response.write("<pg08>")
19.  while (not rs.EOF)
20.    response.write("<entry>")
21.      response.write("<title>" & rs("Name") & "</title>")
22.      response.write("<content type='html'>")
23.
```

```

24.      response.write("<b>2008 URE:" & rs("Team") & "<br>"
& rs("HomeTown") & "<br>&lt;img src='&quot;' & rs("Picture") &
"&quot;&gt;")
25.
26.      response.write("</content>")
27.      response.write("<georss:point>" & rs("Location") &
"</georss:point>")
28.      response.write("</entry>")
29.      rs.MoveNext()
30.  wend
31.  rs.close()
32.  conn.close()
33.  response.write("</pg08>")
34.  response.write("</feed>")
35. %>
```

### B. Database Implementation

Lines 1-7 implement the database connections and retrieval. Line 1 begins the JavaScript code. The 2<sup>nd</sup> line is a string that describes the content or file type returned to the user. In the 3<sup>rd</sup> line, “set conn”, is used to create a connection to a data source. Through this connection, you can access and manipulate a database. Line 4 sets or returns a string value that contains the provider name for a specific connection object. The 5<sup>th</sup> line is used to open the database. The path to the database is in parentheses. The 6<sup>th</sup> line is an American National Standards Institute (ANSI) standard computer language for accessing and manipulating database systems. Structured Query Language (SQL) statements are used to retrieve and update information in a database. SQL works with database programs like MS Access, DB2, Informix, MS SQL Server, Oracle, and Sybase. In this case the command retrieves all of the records in the database from table “pg08” and orders them by the field “name”. The 7<sup>th</sup> line creates an ActiveX Data Objects (ADO) labeled “rs”. This object holds the commands necessary to manipulate and access the database utilizing SQL statements.

### C. Lines 9-16

Lines 9-16 set up the heading content for the XML file. The 9<sup>th</sup> line is a method that writes a specified string to output. This output states what version of xml the code follows and also the encoding, in this case the Latin alphabet. The 10<sup>th</sup> line outputs the protocols or rules for the namespaces used in the XML code. The 11<sup>th</sup> line outputs the title “Polar Grid Team”. The 12<sup>th</sup> line outputs “ECSU NIA 2008”. The 13<sup>th</sup> line outputs what time it was last updated. The 14<sup>th</sup> line outputs and opens the author tag. The 15<sup>th</sup> line outputs “URE Polar Grid Portal”. The 16<sup>th</sup> line outputs and closes the author tag.

### D. Lines 18-22

Lines 18-22 open the first entry utilizing a “While” loop to retrieve data from each record. The 18<sup>th</sup> line outputs “pg08” as a channel for the xml file. Channels group entries into like

characteristics. The 19<sup>th</sup> line is a while statement that continues the commands contained inside the tag until the End Of File (EOF) (end of the database entries) is reached. The 20<sup>th</sup> line outputs another entry from the database. The 21<sup>st</sup> line accesses and outputs the field “name” from the database. The 22<sup>nd</sup> line tells the xml that the following content is in HTML format.

#### E. Lines 24-35

The 24<sup>th</sup> line outputs “content” for the entry in HTML code. An example of HTML is utilizing &lt; when the character “<” is required. The 26<sup>th</sup> line closes the content section. Line 27 makes the location from the database a georss:point which can be read by Google Maps. The 28<sup>th</sup> line closes the entry namespace. The 29<sup>th</sup> line is a statement that lets the program continue on if the loop has reached the end of the database entries. Line 30 is the end of the “While” loop. Line 31-35 finish’s the program off by closing the connections to the database.

## V. RSS

### A. RSS

Really Simple Syndication (RSS) is designed to automatically update a web page, RSS reader, or other device with up to date content. To view an RSS you must subscribe to a source outputting an RSS file using an RSS reader, feed reader, or an aggregator. RSS icons appear as sound waves coming from a speaker in an orange square. To subscribe to an RSS feed of a certain web page look for the RSS symbol as shown in figure 3. Once you have clicked on the icon it will allow you subscribe to the RSS feed. These services are available in many locations including ECSU’s Campus News section. If you were to subscribe to a sports RSS feed, every time something new was uploaded as news about that certain sports event the RSS feed information would automatically update on your system. This would occur without having to go and view the updated information on their web site. This project used Microsoft Outlook and iGoogle to view our RSS feeds. Outlook has an RSS reader included in the software package. When a new update occurs the RSS sends the update in the form of an email that can easily be accessed.



Figure 3: Example of the RSS Feed Subscriber Icon

### B. RSS Coding

An RSS feed is written in XML code utilizing namespaces instead of tags as in HTML. Here is an example of a simple RSS feed:

```
1. <?xml version="1.0"?>
2. <rss version="2.0">
3. <channel>
4. <title>NIA Upcoming Events</title>
```

```
5. <description>ECSU Undergraduate Research</description>
6. <link>http://nia.ecsu.edu/pgbeta5.xml</link>
7. <item>
8. <title>Distinguished Visiting Lecture Series</title>
9. <description>Professor Mohan Munasinghe - 2007 Nobel Peace Prize Co-Winner on Global Warming, July 21, 2008 in the KE White
10. </description>
11. <link>http://nia.ecsu.edu/ureomps2008/ </link>
12. </item>
13. </channel>
14. </rss>
```

Line 1 tells the RSS reader that this file is an XML document. Line 2 tells the reader it is also an RSS feed and which version of XML is used. Line 3 lets the RSS reader know that the following items are grouped together under a similar type. Line 4 tells the type of RSS it is and where it is coming from. Line 5 gives a description of what the RSS feed is about. Line 6 creates a link to the channel so the information can automatically update. Line 7 begins the creation of the item. Line 8 gives the title of the item. Line 9 gives a description of the event or news item on line 9. Line 10 closes the description section. Line 11 creates a link to more information about the event. Line 12 ends the item. If there were more items to be included in this feed, another item would be opened and lines 8-12 would be repeated. Line 13 closes the channel namespace. Line 14 ends the RSS feed update.

## VI. GEORSS

### A. Overview

Geographical Encoded Objects for Real Simple Syndication or GeoRSS is the method use in pinpointing physical locations of geographical content. GeoRSS gives the user the ability to search the web for information by using geographical locations.

### B. Project Use

In researching relevant structures for the compiling of the Polar Grid data, GeoRSS was used as the inputted information to denote the location of research students in the summer 2008 program. Student information concerning names, locations, personal websites, and pictures were stored into tags using html coding and outputted as pinpoint locations using Google Maps. The namespace employed in the XML file was <georss:point> Latitude Longitude </georss:point>.

## VII. CONCLUSION

Using Adobe Dreamweaver as the development tool individual porlets and RSS feeds were created using iGoogle as a portal. GeoRSS namespaces were used to generate points into Google maps. Within Google Maps Each point displayed information using HTML tags for designated students including name, location, and an image. In correlation to the

Polar Grid project the tags created would have included the radar data, a transformed image, the coordinate locations of taken data, the date, and other various attributes concerning the research of the Antarctic and Greenland Polar Regions. A second portlet was also created using the framework of the previous gadget. This portlet displays the upcoming events for the NIA web site. In corresponding the Polar Grid project this portlet would have the capabilities of providing up-to-date information on upcoming expeditions, a calendar capable to alter with a tentative scheduling, and other various capabilities.

### VIII. FUTURE WORK

The future goal of this project is to create working portlets/gadgets, which will implement the MatLab, and GOAP code, which is currently being developed at the University of Kansas. These portlets/gadgets will store, collect, and process data from working scientists in the northern and southern Polar Regions to promote a cost efficient way to test and analyze data in a timely manner. The Polar Grid teams at Elizabeth City State University will be developing portlets/gadgets in conjunction with Indiana University and The University of Kansas to store and manipulate data from Antarctic and Greenland expeditions. These portlets/gadgets will be developed to allow students and teachers access to data, applications, and educational lessons for the continued study of ice sheets.

The conversion of the glossary into a database format is recommended with the proceeding works of the Polar Grid project. Additional terms are also to be included into the glossary as research continues. With the obtainment of field research from Kansas University student researchers will add relevant information concerning the glacier studies into gadgets. Preceding this information, a tutorial will also need to be created. This tutorial will be utilized by beginning developers on how a grid is to be constructed using specific of set procedures.

### ACKNOWLEDGEMENTS

The 2008 Undergraduate Research Experience in Ocean Marine and Polar Sciences (URE OMPS) Polar Gateway Team would like to thank Dr. Marlon Pierce for his contributions to our project through his fundamental works as a lead researcher in the area of Grid Web Portal development. Acknowledgements would also like to be extended to Dr Linda B. Hayden as the Principle Investigator of URE OMPS program because without her dedication to the promotion of undergraduate student's professional growth funding for this program would be non-existent. Recognition would also like to be provided to all staff members of Elizabeth City State University and other cooperating Universities that aided in that completion of a successful summer project.

## REFERENCES

### RESEARCH PAPER LINKS

- [1] "Globus Toolkit Version 4: Software for Service-Oriented Systems" by Ian Foster, Retrieved June 16, 2008 from the World Wide Web, <http://www-unix.globus.org/alliance/publications/papers/IFIP-2006.pdf>
- [2] "The Anatomy Grid" by Ian Foster, Retrieved June 16, 2008 from the World Wide Web, <http://www-unix.globus.org/alliance/publications/papers/anatomy.pdf>
- [3] "Installation and Implementation of a LAMP Documentation Server" , Retrieved June 20, 2008 from the World Wide Web, Tyrone Whitehurst, Illiana Thomas, Andrea Grumbles, Amanda Bland and Camden Hearn, <http://nia.ecsu.edu/ureoms2007/teams/grid/GRIDTeamPaper2.pdf>
- [4] "What is the Grid? A Three Point Checklist" by Ian Foster, Retrieved June 20, 2008 from the World Wide Web, <http://www-fp.mcs.anl.gov/%7Eifoster/Articles/WhatIsTheGrid.pdf>
- [5] "FAQS: Grid computing" by John Moore, Retrieved June 19, 2008 from the World Wide Web, [http://www.fcw.com/print/11\\_25/news/89311-1.html](http://www.fcw.com/print/11_25/news/89311-1.html)
- [6] "A General (Gentle?) Introduction to (Grid) Portals/Gateways" by Anurag Shankar, Retrieved June 18, 2008 from the World Wide Web, <http://dhruv.uits.indiana.edu/portals/portals-101.pdf>
- [7] "TeraGrid Forum", Retrieved June 18, 2008 from the World Wide Web, <http://www.teragridforum.org/mediawiki/index.php>
- [8] "TeraGrid Science Gateways Primer", Retrieved June 18, 2008 from the World Wide Web, [http://www.teragridforum.org/mediawiki/index.php?title=TeraGrid\\_Science\\_Gateways\\_Primer](http://www.teragridforum.org/mediawiki/index.php?title=TeraGrid_Science_Gateways_Primer)
- [9] "HPC: The Coming (Present?) Challenge -- A Perspective from the Trenches" by Anurag Shankar, Retrieved June 18, 2008 from the World Wide Web, <http://dhruv.uits.indiana.edu/portals/hpc-coming-challenge.txt>

### EDUCATIONAL LINKS

- [10] "Welcome To The GRID On Tryscience", Retrieved June 16, 2008 from the World Wide Web, <http://www.tryscience.org/grid/home.html>
- [11] "Grid Café", Retrieved June 17, 2008 from the World Wide Web, <http://gridcafe.web.cern.ch/gridcafe/>
- [12] "Open Grid Services Architecture Glossary of Terms" by J. Treadwell and Hewlett-Packard, Retrieved June 17, 2008 from the World Wide Web, <http://www.ogf.org/documents/GFD.120.pdf>

### RESOURCES

- [13] "Open Grid Forum" ", Retrieved June 19, 2008 from the World Wide Web, <http://www.ogf.org/index.php>
- [14] Marlon Pierce Portal PowerPoint, Retrieved June 19, 2008 from the World Wide Web, <http://grids.ucs.indiana.edu/ptliupages/presentations/Pierce-ScienceGateways-RoundTable.ppt>

### GEORSS XML RESOURCES

- [15] "GEORSS feed xml tags", Retrieved June 23, 2008 from the World Wide Web, <http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=1648907&SiteID=1>
- [16] "GeorSS Java Library", Retrieved June 23, 2008 from the World Wide Web, [http://mars.uni-muenster.de:8080/GeoRGE/georss\\_java\\_library.htm](http://mars.uni-muenster.de:8080/GeoRGE/georss_java_library.htm)
- [17] "Yahoo! Maps GeoRSS - Version 2", Retrieved June 23, 2008 from the World Wide Web, [http://developer.yahoo.com/maps/georss/\(with examples\)](http://developer.yahoo.com/maps/georss/(with%20examples))

### RSS FEED RESOURCES

- [18] "What Is RSS" , Retrieved June 23, 2008 from the World Wide Web, <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>
- [19] "Video: RSS in Plain English", Retrieved June 23, 2008 from the World Wide Web, [www.youtube.com/watch?v=0klgLSxGsU](http://www.youtube.com/watch?v=0klgLSxGsU)

- [20] "RSS Feed Reader", Retrieved June 24, 2008 from the World Wide Web, <http://rssfeedreader.com/>
- [21] "Introduction to RSS", Retrieved June 24, 2008 from the World Wide Web, <http://www.webreference.com/authoring/languages/xml/rss/intro/>
- [22] "Making An RSS Feed" by Danny Sullivan, Retrieved June 16, 2008 from the World Wide Web, <http://searchenginewatch.com/showPage.html?page=2175271>
- [23] "RSS: What it is, Where to get it, How to make it, How to use it" by Robert Teeter, Retrieved June 23, 2008 from the World Wide Web, <http://www.interleaves.org/~rteeter/rss.html>
- [24] "RSS Tutorial", Retrieved June 23, 2008 from the World Wide Web, <http://www.mnot.net/rss/tutorial/>
- [25] "RSS Feeds" by Wendy Boswell, Retrieved June 23, 2008 from the World Wide Web, <http://websearch.about.com/od/rsssosocialbookmarks/f/rss.htm>
- [26] "Development Fundamentals" , Retrieved June 24, 2008 from the World Wide Web, <http://code.google.com/apis/gadgets/docs/fundamentals.html>
- [27] "How To Create Your Own iGoogle Gadget", Retrieved June 24, 2008 from the World Wide Web, <http://johnnyblogger.wordpress.com/how-to-create-your-own-rss-igoogle-gadget/>
- [28] "What are gadgets powered by Google?", Retrieved June 24, 2008 from the World Wide Web, <http://www.google.com/webmasters/gadgets/>

#### IGADGET RESOURCES

- [29] "Development Fundamentals", Retrieved June 22, 2008 from the World Wide Web, <http://code.google.com/apis/gadgets/docs/fundamentals.html>
- [30] "How To Create Your Own iGoogle Gadget", Retrieved June 22, 2008 from the World Wide Web, <http://johnnyblogger.wordpress.com/how-to-create-your-own-rss-igoogle-gadget/>
- [31] "What are gadgets powered by Google?", Retrieved June 22, 2008 from the World Wide Web, <http://www.google.com/webmasters/gadgets/>

#### PRINTED RESOURCES

- [1] "Instant JavaScript" by Brian K. Holman & William Lund 1997
- [2] "XML by example" by Benoit Marchal 2002