SysCon 2011 – 5th Annual IEEE International Systems Conference
Montreal, Canada, April 4–7, 2011

Sensor Integration and Analysis for Visual Identification of Environmental Patterns

Patricia Morreale
Department of Computer Science
Kean University
Union, NJ 07083
pmorreal@kean.edu

Johana Callegari
Department of Computer Science
Kean University
Union, NJ 07083
callegaj@kean.edu

Guillermo Valle
Department of Computer Science
Kean University
Union, NJ 07083
memo15ec@gmail.com

Francis Kendall
Department of Computer Science
Kean University
Union, NJ 07083
flkendall@comcast.net

***Abstract*— Sensor data integration is crucial for the analysis and identification of environmental patterns. In this research project, publicly available datasets of environmental sensor data are integrated with Google Earth. Information on environmental patterns is identified, analyzed, and presented on the Web in response to user queries. This prototype demonstrates how sensor data can be integrated with Google Earth to enable the geographic and temporal context display of environmental data, with an interactive format, for users to analyze and identify environmental patterns in real-time.**

Keywords-sensor networks; systems integration; data mining; application design.

I. INTRODUCTION

Large datasets of time-series data are challenging to work with. The volume of data is overwhelming, and traditional data mining technique are designed for transaction-oriented data, not time series data. Recent work [1,2] has identified methods which can be used for data mining on time-series datasets, but presentation of the information, once identified, in a geographical and temporal context, remained as a challenge. The goal of data visualization is to assist in data understanding and comprehension by using the human visual system to identify patterns and trends. In addition, visualization encourages the identification of outliers. Well-designed visual representations can replace cognitive calculations with simple perceptual inferences and improve comprehension, memory, and decision making [3**]**. By making data more accessible and appealing, visual representations may also help in engage more diverse audiences in exploration and analysis.

Prior research [2] has shown that most sensor networks collect an overwhelmingly large volume of data, very little of which is ever analyzed or used in a meaningful way. Certain parameters can be modified to adjust the frequency and size of data reports. The challenges of distributing and replicating data management [1] are important when coordinating the movement of large data volumes. Spatio-temporal datasets [4] are often very large and difficult to analyze and display.

The project presented here includes a demonstration of the NOAA surface data which was extracted and saved in a local relational database. User queries, from the web, were sent to the relational database. Responses to the queries were presented in real-time back to the user. Google Earth [5] was used to display the data from the NOAA dataset based on the data location, which involved reporting stations from all states in the United States. This implementation provides a basis for future predictive algorithms based on environmental measurement.

II. SYSTEMS DESIGN APPROACH

Google Earth, a widely available 3D graphical application, was selected as the primary environmental data visualization tool. Very large datasets of environmental information, such as those available from NOAA, NASA, and other U.S. government entities, were used to provide data for analysis and presentation. The NOAA Integrated Surface Data (ISD) dataset [6] was used as it had data values, collection years, and data formats which would be most useful.

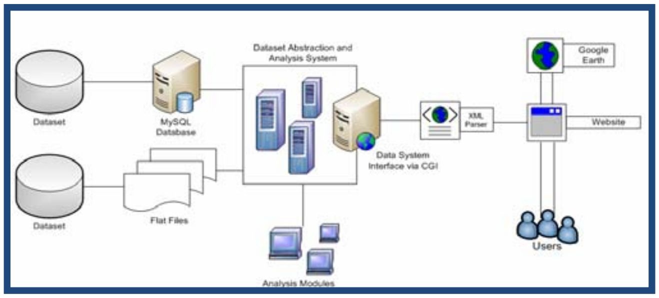


Figure 1. Design of the Data Extraction and Visualization Systems

The resulting system design (Figure 1) enables the display of regional and global environmental data in response to user queries. Users are able to select from a drop-down menu of available databases, including the NOAA ISD, from which they would like to use to obtain data in response to their query. The data is displayed using different tools such as Google Earth and XML/SWF charts, which are Flash-based and embedded into the website. Scripts using PHP are used to output the XML in other formats depending on how the data will be displayed on the website. PHP handles all the data and associated display, while generating a dynamic web page based on the user query.

Data analysis is an intermediate step (Figure 1), which occurs after the data is initially obtained from the very large database (VLDB). The VLDB data is archived locally in a format which supports data analysis. Mathematical functions and graphs, customized for anticipated environmental calculations, are also integrated into the website. These functions facilitate the user analysis of the data and include functions to support for new discoveries from the readings presented on the webpage. GIS sections are included for the user. The overall objective of the systems design effort was fast data retrieval, analysis and presentation.

III. Data Mining for Visualization

Once the data are obtained from the NOAA ISD dataset, data mining and analysis for presentation becomes the focus of the effort. The data mining process usually consists of three phases, or steps:

- Pre-processing (data preparation)
- Modeling and Validation
- Post-processing (or deployment)

The NOAA ISD dataset describes the behavior, location and identification of sensors all over United States. Visualization techniques are widely recognized to be powerful in this domain, since they take advantage of human abilities to perceive visual pattern and to interpret them.

The visualization method for the data used in this research begins from the user's query or request which is forwarded from the website. Once the user request is received, the XML parser process begins. The XML parser is integrated into the Google Earth script and transports the data from the central relational database to the Google Earth site. In this case, data was displayed data from the NOAA ISD dataset using Google Earth as a new method to display and interact with data.

One of the visualization tools is Google Earth which was used to render in 3D the mining outcomes. Google Earth displays satellite images of varying resolution of the Earth's surface, allowing users to see cities and houses looking perpendicularly down or at an oblique angle, with perspective. Also, Google Earth allows users to search for addresses in some countries, by entering their coordinates, or simply using the mouse to browse to a location [5]**.** For these reasons, Google Earth is an ideal candidate to display environmental data. The other visualization tool used was Flash Charts to provide advanced user interaction with the mining results. The key feature of Flash Charts is the ability to visualize both the data satisfying specific mining rules and the shape of the rules extracted. Charts allow the discovery of specific relationships between the shape of the initial dataset and the shape of specific rules [6].

A. Data Mining and Presentation

Identifying, managing and presenting data involves many processes. The different processes from the extraction of the data to the user's visualization require manipulation of the visualization tools, so that the large volume of data does not crash or overload the system. Also, managing real-time time-series data with these tools permits near real-time presentation to users which shows what is happening in real world.

Google Earth utilizes KML (Keyhole Markup Language) which is a file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard **[7]**. Using KML script language provides set features to Google Earth as place marks, images, polygons, 3D models, textual descriptions, etc.

PHP script language was selected for use in the development of the website. PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. PHP is the best at performing operations and provides interpretation of other script languages. PHP also provides support and helps to manipulate data from other sources easily, while producing program output on its standard output channel. All information from the SQL database is written using XML script language which is used as a XML parser to transfer data for display. The XML parser is preferable for large files.

The design goals emphasized simplicity, generality, and usability over the Internet. This was accomplished by using a textual data format with strong support via Unicode for the languages of the world. In addition, Adobe Flash, a multimedia platform, was selected to present the analysis side of the visualization design. Flash manipulates vector and raster graphics to provide animation of text, drawings, and still images, while supporting bidirectional streaming of audio and video. Flash can capture user input via mouse, keyboard, microphone, and camera. Flash contains an object-oriented language called Action Script.

B. Data Analysis

Datasets of time-series data are more complex and challenging to analyze than conventional datasets, which are usually composed of discrete data events, or transactions.

One of the reasons for the increased complexity of datasets containing time series data is that a large volume of data is more difficult to visualize and more complicated to display. The NOAA ISD dataset is particularly representative of this challenge as it has more than 8000 reporting sites and includes daily data going back to 1901. Charts are used to display time-series data and until now, it has been difficult to know which reporting sites have data in the different dates range. Several scripting languages were used display the data. For this reason, if the central database goes down, all of the tools can crash in the internal system easily. However, this single point of failure is currently necessary in the design to make sure that the data is appropriately analyzed and processed prior to visual presentation. Moreover, the data retrieval and the presentation of this data required integration of many tools for visualization as Google Earth and Flash Charts.

C. Visual Presentation

Two main approaches for the user's visualization were used as part of this research.

1) Visual Presentation with Google Earth

This approach displayed data from the NOAA dataset using Google Earth as the primary method of user interaction. Place marks were used to display each node in different states of the United States. These nodes are shown in the whole map (Figure 2). Currently, the visualization for the Google Earth side let users choose the dataset, country and states to visualize the nodes in 3D. Users can zoom in, zoom out, move the globe, and click into the node to see the location close and they can count the nodes available in a specific city. For now, the system is displaying the NOAA dataset, US and the 50 states.

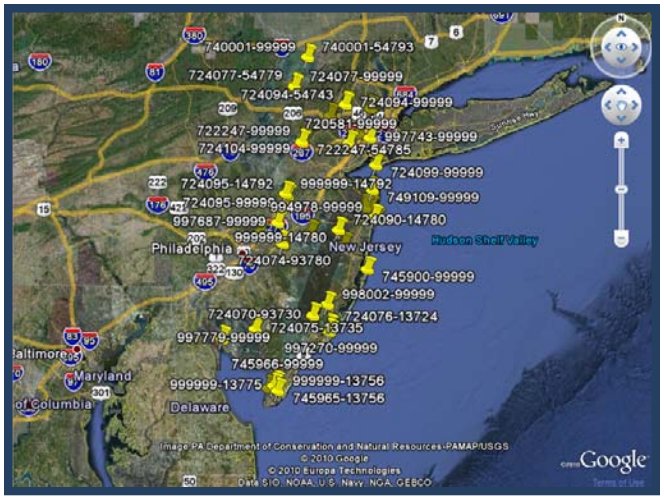
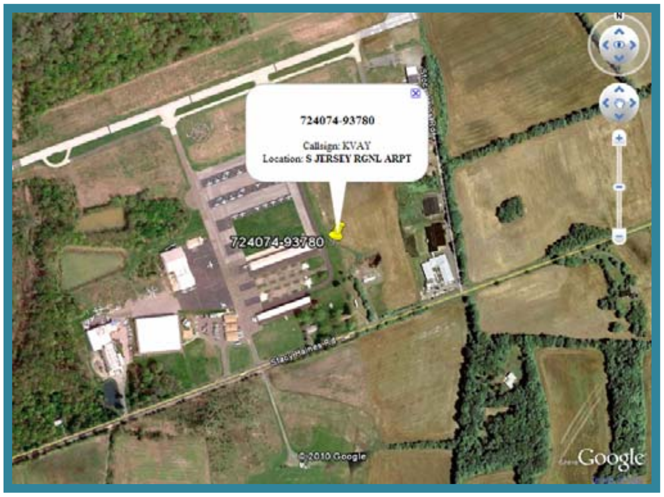


Figure 2. Google Earth displaying nodes from New Jersey.

Each node in Google Earth has the capability to display the NOAA id number and any other information that system wants to provide (Figure 3). For now each node displays location, id and call sign which are useful if the system wants to display data from particular nodes then it will let the user to interact with this specific node too.


724074-93780

Callsign: KVAY
Location: S JERSEY RGNL ARPT

Figure 3. Displaying data of a specific node in Google Earth.

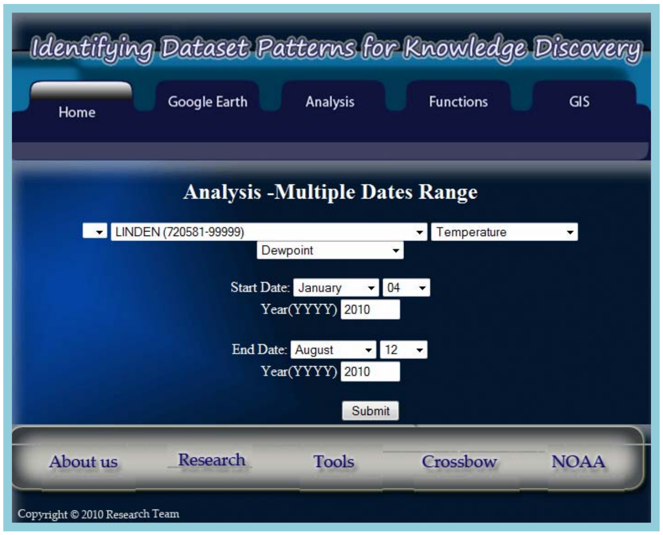
Google Earth makes it easy to understand the information of each node. Nodes are displaying in real time. For this reason, if the NOAA ISD dataset adds or deletes any node it will appear on the Google Earth side from the system because the information is updated.

2) Visual Presentation with Charts

The contrasting approach for the user's visualization is in XML/SWF Charts where the user can make a deeper quantitative analysis using multiple variables patterns and multiple dates range. Users can choose the dataset, country and states to be used for the site of the visualization.

In the initial prototype, the NOAA ISD dataset interface was the first one enabled, which was idea as it includes data from all 50 states in the United States. After users have selected a state, all of the nodes from the current state next to these will be displayed and the users will be able to see the id number from these nodes in the system.

Users are able to select multiple variables for display (Figure 4). This permits users to see a complete and detailed analysis from the specific node that they have selected with the variable patterns that they want to display in the chart. Furthermore, users can specify the dates range for the current nodes such as the start date (M/D/Y) and the end date (M/D/Y), inclusive, that they want to view data from the node selected. All of these options are available to users to provide them with a complete analysis in using all available environmental data. The NOAA ISD data includes real-time reports, so the information analyzed and displayed is in real-time and includes just-collected data.


Identifying Dataset Patterns for Knowledge Discovery
Home
Google Earth
Analysis
Functions
GIS
Submit
About us
Research
Tools
Crossbow
NOAA
Copyright © 2010 Research Team

Analysis -Multiple Dates Range

LINDEN (720581-99999) | Temperature | Dewpoint

Start Date: January 04 Year(YYYY) 2010

End Date: August 12 Year(YYYY) 2010

Figure 4. Multiple date range selections for state of New Jersey.

In Figure 4, a node from Linden, NJ is selected and the two variables patterns are measured in degrees Fahrenheit. These variables are *temperature* and *dew point*. The Start Date is January 4, 2010 and the End Date is August 12, 2010. As a result, when the user clicks the submit button the output will be a chart showing the temperature and dew point between those dates for the Linden, NJ sensor (Figure 5).

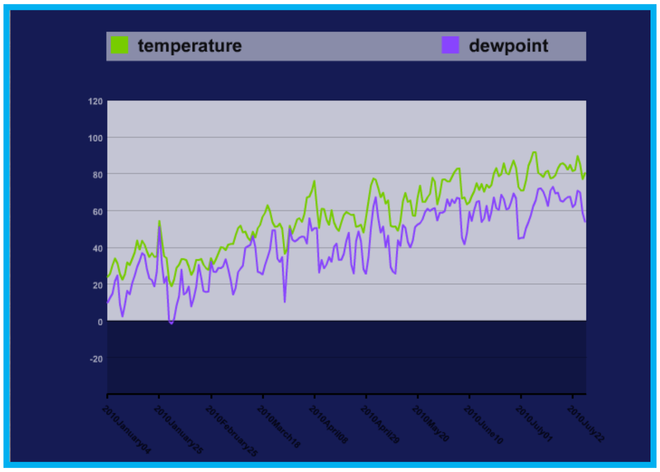


Figure 5. Display analysis of multiple variable patterns and date ranges.

In the left side, the chart shows the degrees Fahrenheit by dates for the current node. Also, it displays readings from January 4, 2010 to July 22, 2010. Even if the user puts the end date range to August 12 the NOAA dataset will show the data that they have for the current node. It is difficult to know when the current node has data between specific dates because the NOAA dataset has a large volume of data and the data is displayed in real time. The significant point is that the chart processes time series and real-time data from 1901 to 2010 to make a deep analysis of the variables patterns from a specific node.

IV. System Design Challenges

When developing this project the first goal was to collect data from a larger database and get it into the MySQL database. One of the problems encountered was that larger datasets need to be broken down into smaller parts for presentation to the web. Module datasets perform best in the relational database. From the larger datasets, both useful and useless data was found. The useless data was identified and discarded, leaving the useful data for further analysis and presentation on the website. The resulting dataset was put into Perl scripts to be sent to Google Earth.

Importing the data for use in Google Earth was challenging. Usually, data is provided for web visualization through the use of an HTTP GET request to a URL. The return result is data in XML format, which is later send to Google Earth. The problem in this system initially was converting the XML format to KML for Google Earth to use. KML is an XML-like language for Google Earth but our system wanted only to display the data and not to process it in Google Earth. The easiest way to add custom data is to add it directly to Google Earth as Extended Data. It will preserve the data but not process it, using namespace prefix so is easy to indicate the data, which is in XML format [9, 10]. The problem encountered was that the data in XML format did not using namespace prefix, which makes hard to transfer the data without using another language similar to XML to convert the data into KML. A similar problem was found while working with the chart software, but the difficulty was now migrating from one XML presentation to XML presentation required by the XML/SWF chart software. The chart software required that the XML tags appear in a specific sequential order, in order for the data to be correctly read by the chart software for presentation. The re-ordering of tags from one XML presentation to the XML presentation required by the chart software was dictated by the Perl scripts. The chart data and all the features of the charts are included inside a tag name chart (<chart>). The tags will have the data source indicated as well as all the features in HTML. The challenge was getting this data into the PHP Scripts. The process of obtaining or pulling the data from the dataset for analysis needed to be as simple and efficient as possible. When uploading the resulting scripts to test that they were working on the server, a bigger problem presented itself: how will the user query access the scripts in order to obtain the data needed for analysis? How should the pages be linked in order to present either chart analysis display or the Google Earth data?

V. Data and Information sharing Approaches

The initial effort was spent getting Google Earth working in response to user queries. Once the design team became familiar with KML [10], an approach was needed to get the

XML data into Google Earth. There are several ways to get data into KML, most of which require converting from XML to some other formats and then into KML or HTML. For example, to get our data into KML so Google Earth was able to display the data, XSLT (Extensible Stylesheet Language Transformation) was used to transform the data into KML [11]. To transform the data from XML, a XSLT style sheet was created. XSLT is an XML-based language that is used for transformation of XML Document into another XML type of document which in this project was KML to overlay our data into Google Earth [10]. To create the XSLT style sheet the data needed to be grouped. Different grouping techniques were used to facilitate grouping and then applying the style sheet to the XML data in PHP [12, 13]. This was the easiest approach found, however, the process had a high maintenance cost. Every time the data was updated it needed to be regrouped and more work was required. Multiple Perl scripts were designed to output all the data from any dataset, and transfer it to the relational database selected, mySQL. After the data is in the mySQL database it was analyzed and only the useful data was taken forward and put up on a server. The PHP CURL Function was used to solve this problem and parse the XML data.

The CURL Functions are a library which allows connection to different servers and types of protocols [14]. Any PHP script in the website needs to be able to connect to the Perl Scripts to obtain data and that is the functionality that the CURL Functions support. These Functions will connect to the Perl script and download all the output into memory for me to parse.

The CURL functions only solve part of the problem. The hard part is to parse the volume of data from the XML file. One approach is to use the PHP XML libraries to parse the downloaded output into an array and use two functions which are xml to array and get_value_by_path [15]. Later on, for greater functionality, XML DOM parsing was used instead of the XML libraries as XML DOM is simpler. These types of functions and objects support the loading of an XML document into an XML DOM object. The XML DOM defines a standard way for accessing and manipulating XML documents. After the document is loaded it is save into an array which will hold the data for the PHP scripts to use.

Most of the scripts built for the website have the same procedure: using CURL functions and parsing the XML data. The only differences in all the scripts are the purpose of the script.

For Google Earth, the KML output was written in a String with the data as variables taken from the arrays. Google Earth includes a place mark with all the basic data and location of the nodes for the user to visualize. The Place mark includes detail about the location, the node-id and its description. The importance of Google Earth is that with KML, the Place mark can be modified in order to have a better design and look. To add more features and better visualization, only the KML String was modified to improve the visualization in Google Earth. The description permits the addition of data and any type of media to improve visualization, such as the inclusion of videos in the future. After PHP outputs the KML document, the Google API reads all the information and outputs the data into Google Earth, in the same manner as any Google Earth program.

For the XML/SWF charts [16, 17], the same procedure is used to obtain the data. The only problem encountered is when handling the data range (starting date to final date) for data acquisition. The Perl script outputs the data in the format YYYY:MM:DD:HH:MM:SS. In PHP or any other language this causes a problem when reading the data, as this type of numeric order is not recognized. The solution was when parsing the dates to store the data in different arrays. For example, the year is stored in an array called *years* and the day in *days*. In order for the script to know which year belongs to what day, it must know the position of the day within the array. The year for node one in the nodes array will be in position one of the array years. The rest of the scripts put all XML data in a string, which is output for chart presentation.

The same approach is used for the rest of the website. In order for one page to go to the next one by passing all the data, the same process is used. The user can specify the data to be used through a drop-down menu which is completed and past from page to page. This was managed by using a PHP form shared by web pages. This approach provides support for the entire website, permitting data to be passed seamlessly from one page to the next.

A prototype demonstration of the research site (http://trinity:81/Research10/index.php) illustrates the two main approaches for the user visualization. Both the Google Earth presentation site and the Data Analysis –Charts site offer advanced analysis of specific environmental information from sensor nodes reporting from numerous sites, archived in the NOAA ISD.

Details about the analysis and queries supported are provided at the website. This approach can be used in many other areas in addition to environmental monitoring, such as health care and public safety applications.

VI. EVALUATION AND PEFORMANCE

The resulting system in entered by an initial webpage menu. This menu provides a chart to assist in navigating the website and data retrieval. From the initial screen, users can select what they wish to view.

The Google Earth application entry point will ask for

- the dataset
- the country and
- the state or regional division.

From there it is possible to go to the API to view all the node locations, provided with some basic information and a quick description. All the pages have link to all other pages, which makes easier for the user to go back and access the data.

The data analysis application entry point will ask for

- the specific Node ID
- variables and
- the date range

After entering this information, the user will be taken to chart display screen. The chart and any associated analysis will be displayed immediately after the user submits a request. Additional functionality is being added to this screen as it is developed. A GIS section is also included. All the data presented is time-series sequential information, in contrast to more discrete-event transaction oriented presentations of conventional data mining systems, and this system loads quickly for display and use.

VII. CONCLUSION

This research integrates the NOAA ISD dataset, Google Earth and chart software with data analysis and visual presentation in a geographic context. This provides a foundation for the use of predictive algorithms based on environmental measurement using two main approaches. An understanding of the information gathered by large datasets will permit patterns and trends to be identified. For this reason, visualization is very important as it lets users understand the data displayed. By providing users with current information to predict or avoid environmental situations which might have adverse consequences, public health and safety can be supported.

Google Earth and Charts are tools which are easy to integrate. In addition, these tools add features, such as changing node colors in the map to warn users of environmental situations and warning or show the sensors as different icons or in color. The integration of environmental sensors with data analysis will help users to understand the environment currently and in the future. The research site provides a visual display of time series data in geographic and temporal context. The knowledge discovery in large datasets, such as those publicly available through NOAA or NASA sites, is vital. Time -series data from the NOAA dataset is displayed in real-time for users to understand and can be used in any type of environmental application, including future predictive algorithms.

There are many organizations that collect data, including environmental data as discussed here, but these large volumes of data are infrequently used or analyzed. Visual presentation of significant results is rare. The objective of the research project was to collect data from large datasets and visually display the data in a meaningful context for the user. This process has been demonstrated, with widely available tools for analysis.

Future plans include using our approach to other large datasets provided by NOAA, as well as other government and public groups and provide visualization and analysis to a larger area of the earth.

REFERENCES

[1] A. Anastasia, V. Kantere, and D. Dash. "Managing Scientific Data." Communications of the ACM , Vol. 53, No. 6 , 2010, pp. 68-78.

[2] P. Morreale, F. Qi, P. Croft, R. Suleski, B. Sinnicke, and F. Kendall. "Real-Time Environmental Monitoring and Notification for Public Safety." IEEE Multimedia, Vol. 17, No. 2, April -June 2010, pp. 4-11.

[3] J. Heer, M. Bostock, V. Ogievetsky, and Stanford University. "A Tour through the Visualization Zoo." A Survey of Powerful Visualization Techniques, from the Obvious to the Obscure: 1-21.

[4] P.Compieta, S.Di Martino, M.Bertolotto, F. Ferrucci, T. Kechadi."Exploratory spatio-temporal data mining and visualization."February 2007, pp.255-279.

[5] "Google Earth." Wikipedia, the Free Encyclopedia. Web. 07 July 2010. <http://en.wikipedia.org/wiki/Google_Earth>

[6] National Oceanic and Atmospheric Administration (NOAA) Integrated Surface Dataset (ISD) http://www.ncdc.noaa.gov/oa/climate/isd/index.php

[7] J.Falby,M. Zyda, D.Pratt and R. Mackey"NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation."In Computers & Graphics,Vol 17, No.1, pp.65-69

[8] "Keyhole Markup Language." Wikipedia, the Free Encyclopedia. Web. 07 July 2010. <http://en.wikipedia.org/wiki/Keyhole_Markup_Language>

[9] Miles, Jake. "Overlay Data on Maps Using XSLT, KML, and the Google Maps API, Part 2: Transform and Use the Data." IBM - United States. 09 Sept. 2008. Web. 06 July 2010. <http://www.ibm.com/developerworks/library/x-geomap2/>.

[10] Google. "Adding Custom Data - KML." Google Code. Web. 06 July 2010. <http://code.google.com/apis/kml/documentation/extendeddata.html>

[11] "XSLT." Wikipedia, the Free Encyclopedia. Web. 06 July 2010. <http://en.wikipedia.org/wiki/XSLT>.

[12] "PHP: Introduction - Manual." PHP: Hypertext Preprocessor. Web. 06 July 2010. <http://www.php.net/manual/en/intro.curl.php>.

[13] "PHP: Xml_parse - Manual." PHP: Hypertext Preprocessor. Web. 06 July 2010. http://www.php.net/manual/en/function.xml-parse.php

[14] "PHP." Wikipedia, the Free Encyclopedia. Web. 06 July 2010. <http://en.wikipedia.org/wiki/PHP>.

[15] " Perl." Wikipedia, the Free Encyclopedia. Web. 29 July 2010. <http://en.wikipedia.org/wiki/Perl>.

[16] "XML/SWF Charts Introduction." Maani.us. Web. 29 July 2010.

[17] <http://www.maani.us/xml_charts/>.