Using open source Geographical Information Systems and statistical tools to develop a web based *Spatial and Statistical Analysis and Interpretation (SASAI) Engine* for archaeological use.

**A GRASS + R + PYTHON + QUANTUM GIS + POSTGRESQL APPROACH**

Author : Ervin Kenan Aydin

Version : 1.0

Status : Final

Filename : osgistat_V1.0.doc

**Document History**

Version: 1.0

Status: Final
Contents
I. INTRODUCTION ..............................................................................................................................................4
II. PROJECT WORK FLOW .................................................................................................................................5
III. TECHNOLOGY AND DESIGN .......................................................................................................................5
   Python Technology ...................................................................................................................................7
   RPy (Python Interface for R) + R + QGIS + Python + GRASS + (Ka-Map) ..........................................................7
      R Server Module ..................................................................................................................................8
      Grass Server Module ............................................................................................................................8
      GRASS-Python library (grass.py) ..........................................................................................................8
      Python-SWIG-GRASS interface ............................................................................................................8
II. DATABASE LAYER ...................................................................................................................................9
III. INTERFACE LAYER ..................................................................................................................................9
IV. PRESENTATION LAYER ............................................................................................................................11
IV. ADVANTAGES OF THE MODEL ..................................................................................................................12
IV. SIMILAR PROJECTS/ TECHNOLOGIES .....................................................................................................15
V. CASE STUDY .............................................................................................................................................17
VI. CONCLUSION ...........................................................................................................................................17
VII. REFERENCES ...........................................................................................................................................17
I. INTRODUCTION

With enormous recent progress in all fields of science, physics, chemistry and biology have started being indispensable parts of archaeological research. Besides these, computer sciences and statistics are among other disciplines that are applied to archaeological surveys and excavations. Although statistics is being used by most of the disciplines today, archaeologists have just been discovering new applications of it into various branches of archaeology. Most of the important questions about the archaeological surveys and excavations can be answered and some of the unknowns can be predicted by use of technical analysis and having sufficient technical background.

Geographical Information Systems (GIS) is also a product of the recent developments in computer technologies. Although the use of GIS is still limited in archaeology, combining GIS and Statistics can lead to astonishing results.

Various detailed analysis can be done for archaeological sites by collecting appropriate data and deriving important parameters. However, still in most parts of the world, archaeologists neither have sufficient technical background to apply these newly emerging technologies in their fields, nor have funds to work together with technical consultants.

This project aims to contribute to today’s Archaeology by enabling archaeologists to better interpret and evaluate the outcomes of archaeological research by means of statistical and spatio-temporal components. To do so, a web based, platform independent, and easy to use spatial and statistical analysis and interpretation (SASAI) engine will be developed. This engine will allow archaeologists all over the world to technically analyse archaeological surveys and excavations, understand and use these methods, results and interpretations regardless of their knowledge of or technical background in statistics and GIS. The system is expected to be a major alternative to web based tools that offer similar services, an appropriate model for new research on computing in archaeology, and a solution to specific problems faced by archaeologists in statistics and GIS fields.

The locations of many archaeological sites usually follow a geographical pattern in a specific period of time. Although these spatial and temporal media are available and could be analysed by using statistics and GIS, most of this media have not being analyzed by means of statistics and GIS due to lack of basic technical and technological expertise. An easy to use system could fill the gap between archaeology and specialist technical expertise, and could quickly and hugely influence many ongoing projects by providing researchers with means to better interpret their findings. This project specifically aims to:

1. Develop and use an R-STAT (R Statistical Language), GRASS (Geographic Resources Analysis Support System), QGIS (Quantum GIS) based, platform independent Web Interpretation Engine to help non-technical users/archaeologists understand statistical and spatial analysis.
2. Increase the amount of contribution of Statistics, GIS and Computer Science to archaeology projects by helping the archaeologists to solve the technical problems in these projects by using SASAI Engine.

3. Create a web based spatio-temporal database for the usage of archaeology projects which can be accessed in all over the world.

4. Help archaeologists to understand the technical aspects of archaeology projects easier by easy to understand interpretations of spatial and statistical analysis.

II. PROJECT WORK FLOW

Archaeologist has the appropriate spatio-temporal data of the excavation or survey area. Now he needs to analyse the data but he is not an expert in either GIS or Statistics.

What he has to do is log into SASAI Engine from his desktop/laptop computer, and deploy spatio-temporal data into Engine.

After asking necessary questions to user and organizing the data, SASAI Engine will process data and complete the analysis. The results of these analyses will be interpreted in the SASAI Engine for the use of archaeologists.

SASAI Engine will send these interpretations back to the user’s computer and the archaeologists will able to use and evaluate the interpretations.

III. TECHNOLOGY AND DESIGN

In the recent years, with the increasing demand of the GIS field, new GIS programmes have been developed. Besides the commercial ones like Intergraph, ESRI and Mapinfo products, some open source ones are worth to consider. GMT, Grass, gvSIG, Jump/OpenJump, Quantum GIS, Thuban, uDig are some of these open source GIS programmes.

JUMP: The JUMP Unified Mapping Platform is a GUI-based application for viewing and processing spatial data. It includes many functions common to other popular GIS products for the analysis and manipulation of geospatial data. The JUMP also provides a highly extensible framework for the
development and execution of custom spatial data processing applications. For more information: [http://www.vividsolutions.com](http://www.vividsolutions.com)

**uDig:** uDig is an open source desktop application framework, built with Eclipse Rich Client (RCP) technology. The goal of uDig is to provide a complete Java solution for desktop GIS data access, editing, and viewing. For more information: [http://udig.refractions.net/](http://udig.refractions.net/)

**Thuban:** Thuban is an Interactive Geographic Data Viewer. It is extensible and multi-platform (GNU/Linux, Windows etc.). It is Free Software under the GNU General Public License (GNU GPL). For more information: [http://thuban.intevation.org/](http://thuban.intevation.org/)

**Quantum GIS (QGIS):** QGIS is a user friendly Open Source Geographic Information System (GIS) licensed under the GNU General Public License. QGIS is an official project of the Open Source Geospatial Foundation (OSGeo). It runs on Linux, Unix, Mac OSX, and Windows and supports numerous vector, raster, and database formats and functionalities.

Quantum GIS provides a continuously growing number of capabilities provided by core functions and plugins. You can visualize, manage, edit, analyze data, and compose printable maps. Get a first impression with some screenshots and a more detailed feature list. For more information: [http://qgis.org](http://qgis.org)

**Geographic Resources Analysis Support System (Grass):** Grass is a Geographic Information System (GIS) used for geospatial data management and analysis, image processing, graphics/maps production, spatial modeling, and visualization. GRASS is currently used in academic and commercial settings around the world, as well as by many governmental agencies and environmental consulting companies. GRASS is an official project of the Open Source Geospatial Foundation. For more information: [http://grass.itc.it/](http://grass.itc.it/)

Fig-1) Some of the open source GIS applications in relation to level of users (casual/intermediate/advanced). Quantum GIS and Grass combination is much more advanced compared to other applications.

According to Sherman (2008), GMT and GRASS can be considered as advanced GIS tools and gvSIG, JUMP, OPENJUMP, Quantum GIS, Thuban and uDIG can be considered as intermediate and basic tools.
QGIS is one of the most user-friendly Graphical User Interfaces (GUI) and it has good integration specifications to some of the other open source GIS tools.

Integrating Quantum GIS and GRASS by using appropriate plug-ins will result in a very strong tool with an easy to use interface.

Python technology can be used to develop this project.

In this part of the document, each step of the work flow is explained in details to make the Engine’s each process much clearer.

Python Technology

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

Python runs on Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines.

Python is distributed under an OSI-approved open source license that makes it free to use, even for commercial products.

RPy (Python Interface for R) + R + QGIS + Python + GRASS + (Ka-Map)

For this project, working with GRASS through QGIS and combining GRASS’s power with QGIS’s ease of use would be the best choice.

I. BUSINESS LOGIC LAYER

The first layer in the Python Model is the Business Logic layer. Business Logic layer includes the business servers. The layer should be modular so that any other servers can be added to the engine easily. Basically, the layer consists of two parts which are both open source technologies: R Server Module and GRASS GIS Server Module.
R Server Module

R, which is used for statistical computing and graphics, is a programming language and a complete environment. R is very similar to the S and can be considered as a GNU version of it. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. For more information on R; [http://www.r-project.org/](http://www.r-project.org/)

Grass Server Module

GRASS GIS (Geographic Resources Analysis Support System) is an open source geographical information system (GIS) capable of handling raster, topological vector, image processing, and graphical data. It is released under the GNU General Public License (GPL), and it can be used on multiple platforms, including Mac OS X, Microsoft Windows (natively or with optional Cygwin tools) and Linux. For more information on GRASS; [http://en.wikipedia.org/wiki/GRASS_GIS](http://en.wikipedia.org/wiki/GRASS_GIS)

GRASS-Python library (grass.py)

Python script can contain simple module description definitions which will be processed with g.parser. In this way with no extra coding a GUI can be built, inputs checked, and a skeleton help page can be generated automatically. In addition it adds links to the GRASS message translation system. For code which needs access to the power of C, you can access the GRASS C library functions via the SWIG interface.

Python-SWIG-GRASS interface

GRASS-SWIG (Simplified Wrapper and Interface Generator) interface wraps both raster and vector data C functions plus the general GIS functions. SWIG is:

- A compiler that turns ANSI C/C++ declarations into scripting language interfaces.
- Completely automated (produces a fully working Python extension module).
- Language neutral. SWIG can also target Tcl, Perl, Guile, MATLAB (try PyLab+Matplotlib from python), etc...
- Attempts to eliminate the tedium of writing extension modules.
- The engine will have both GRASS and R servers which will be integrated by SpGrass6 library.
- GRASS and R will be in communication using SpGrass6.
#1) SpGRASS6 Library
By using this library, Grass commands will be used in R.

FigX.x: Using R and Grass in QGIS enables a statistical component which uses QGIS GUI to reach the clients.

II. DATABASE LAYER
Database layer is the layer where all the spatial and non-spatial data will reside. One of the most important processes is to choose the most convenient database for the Engine which should have spatial component. Spatial databases allow users to store and display features, process geo processing and analysis using the spatial functions. In a spatial database, a spatial column contains the geometry of the feature, a row represents feature and a table represents layer.

Although there are too many commercial and open source spatial databases like MySql, PostGreSQL Oracle etc., as this project relies on the open source technologies, spatial database should be open source too. MySql and POSTGreSQL are the opens source spatial databases but it would be more convenient to choose PostGreSQL with PostGIS as it is much more reliable than MySQL spatial database. Firstly, MySql can be used only for storing spatial data but not geo processing them. There have been still ongoing developments for this spatial database. In the other hand, PostGreSQL with PostGIS is the oldest and most powerful open source spatial database which performs exactly reliable geo processes. Moreover, PostgreSQL fully implemented the Open GIS Consortium’s Open GIS Simple Features Specification for SQL.

SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database management systems (RDBMS), database schema creation and modification, and database object access control management.

SQL is a querying language for querying and modifying data and managing databases. It was standardized first by the ANSI and later by the ISO. Most database management systems implement a majority of one of these standards and add their proprietary extensions.

PostGreSQL’s procedural language is PL/pgSQL and it needs PL/pgSQL in order to implement spatial types and functions.

III. INTERFACE LAYER
The interface layer is the layer which gets the request client sends. After getting the request, the interface follows these steps;
1. **Understand the request:** whether the request is about statistics, Geographical Information Systems or project management (PM) server.

2. **Compile the request & Send the command:** Compile the request to form an understandable command for the related server (R, GRASS or PM). The command does not need to be only one in number. The interface can create more than one command to execute the exact request sent by the client.

3. **Receive the Response:** The Interface receives the responses from R, GRASS and Project Manager and sends them back to the clients.

**Quantum GIS**

Quantum GIS is free GIS software and it has been developed using C++, and its graphical user interface (GUI) uses the Qt library, which is used in KDE. Quantum GIS allows integration of plugins developed using either C++ or Python. In addition to Qt, required dependencies of Quantum GIS include GEOS and SQLite. GDAL, GRASS GIS, PostGIS, and PostgreSQL are also recommended.

Quantum GIS can also be used as a graphical user interface to GRASS (QGis - Grass Plug-in: OSGeo4W Setup, Lib Folder). Quantum GIS has a small file size compared to commercial GIS’s and requires less RAM and processing power; hence it can be used on older hardware or running simultaneously with other applications where CPU power may be limited.

**Project Management (GUI) Module**

Project Management (PM) module is the Graphical User Interface and will be used for all the client side communication. By using PM in any kind of web browser, client will able to send commands and receive result sets. The responses from Grass and R servers will be integrated to be seen in PM module, so user will deal with only this module not the actual user interfaces of R or Grass, so that, PM should be very easy to use and it should be designed for the non-technical archaeologists.

PM will be a part of Apache Tomcat Server which will have connections on both R and GRASS Servers. PM should organize the whole project management procedure, spatial and non-spatial data.

**#2) QGIS-Grass Plugin** Quantum GIS has GRASS plug-in so that it is possible to use GRASS features with QGIS GUI. QGIS 1.0 includes QGIS-GRASS plug-in (#2 in FigX.x) by default and after activating it, QGIS and GRASS will be able to communicate with no problem.

**#3) QGis-Python Plugin(RPy)** capability and R installation with Rpy module will make QGIS and R work together. RPy is a very simple, robust, Python interface to the R Programming Language. It can manage all kinds of R objects and can execute arbitrary R functions (including the graphic functions). All errors
from the R language are converted to Python exceptions. Any module installed for the R system can be used from within Python. The main goals of RPy are:

1. To have a very robust interface for using R from Python.
2. the interface should be as transparent and easy to use as possible.

it should be usable for real scientific and statistical computations.

For more information: http://rpy.sourceforge.net/index.html

The integration between Interface and Project Management (PM) will be done by using Python programming Language. In the server-side, Python will be used as the programming language.

#5) **Python Web Processing Service (PyWPS)** is implementation of Web Processing Service standard from Open Geospatial Consortium which is written in Python programming language. Processes can be written using GRASS GIS, but using other programs is also possible. It can be integrated for R, GDAL or PROJ tools. For more information: http://pywps.ominiverdi.org/

**IV. PRESENTATION LAYER**

Presentation layer will present the engine in any kind of platform, regardless of the operating system or hardware. Linux, Windows, Unix, Desktops, Laptops, PDAs, mobiles are some of these platforms. To do so, appropriate technology will be used. In the client side, **Ajax** and **JavaScript** will be used to detect the minimum requirements of the operations planned to be held. To do so, **Intelligent Analysis Module (IAW)** will be located in the presentation module.

**Easy to Use:** Presentation module should be easy to use. Archaeologists will be able to use the GUI easily, regardless of the technical knowledge. Interpretations should take place in the presentation module coded by **Java**, **JavaScript** and **Ajax**.

**Client Side Scripting**

**HTML**

HTML, an acronym for Hyper Text Mark-up Language, is the predominant mark-up language for web pages. It provides a means to describe the structure of text-based information in a document by denoting certain text as links, headings, paragraphs, lists, etc. HTML is written in the form of "tags" that are surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code such as JavaScript that can affect the behaviour of Web browsers and other HTML processors.

**AJAX**
Shorthand for asynchronous JavaScript and XML is a group of interrelated web development techniques used to create interactive web applications or rich Internet applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behaviour of the existing page. The use of Ajax has led to an increase in interactive animation on web pages. Data is retrieved using the XMLHttpRequest object or through the use of Remote Scripting in browsers that do not support it.

**JAVASCRIPT**

JavaScript is a scripting language used to enable programmatic access to objects within the other applications. It is primarily used in the form of client-side JavaScript for the development of dynamic websites. JavaScript is a dialect of the ECMAScript standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

"JavaScript" is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape Communications and current entities such as the Mozilla Foundation.

**IV. ADVANTAGES OF THE MODEL**

1. **Extensibility:** One of the most important parts of the project is its extensible structure. Anyone who knows Python programming language will be able to code new analysis methods so that the Engine will gain new specifications so that new analysis will be held. The base structure will be developed first and Predictive Modelling, Cluster Analysis and Density Analysis packages will be developed after the base structure which will be able to communicate through all of the packages. The new packages will be in the form of Python Plug-in and it will be possible to activate them through a check box in the GUI’s Menu. This structure will be similar to the Quantum GIS Plug-In structure where the user can check all the plug-ins those he wants to use during the use of application and developers can develop new plug-ins, add into the Quantum GIS structure easily.

Three packages, predictive modeling, Cluster Analysis and Density Analysis packages will be developed in this project.
Predictive modeling is a good example. Conolly and Lake (2006) state that predictive modeling in Archaeology is to predict the probability of archaeological settlements occurring in the landscapes according to the location characteristics of these settlements in a surveyed area.

The predictive modeling has some important steps which are sometimes difficult to identify and apply for non-technical archaeologists. Lack of GIS knowledge and expertise sometimes leads not to apply them into the survey area. After deploying the data, the archaeologist will activate the engine by clicking the appropriate buttons in the menu and predictive modeling package will do the rest for the study area. Firstly, Grass GIS will process the layers one by one to identify the parameters which will be important for the predictive model. Layers can include a DEM, ancient settlements coordinates, and reclassified maps like Rock, Soil etc. where, in all the layers, the projections will be changed to UTM first, aspect and slope maps will be obtained from DEM, ancient settlements will be laid over each layer independently and the parameters for each settlement will be identified and written into POSTGIS database. After taking the results, R will be activated to deal with the spatial statistics. The content of both the spatial and statistical analysis can differ according to situation but, in each case, archaeologist/user will click just one button and will send one command. The command will be interpreted in interpretation engine and it will separate to more than one command which will be understandable by R and Grass. After the processes have been completed, the result, directed to interpretation engine, will be interpreted again and the user will see the final interpreted result, which is planned to be
easy to understand.

One of the most complex processes held in the engine is predictive modeling as it is more than one step. So, in this case, the interpretation engine will separate the command to appropriate commands for R and Grass, Grass should do the appropriate work and the results should be transferred to R for spatial statistical analysis. R, will transfer all the result sets, by not doing any change and all of the interpretation and changes should be done in interpretation engine.

Compared to Predictive Modelling Package, density analysis package and cluster analysis package would be easier to develop as they have less separate steps. One of the most important points is to develop the ability to communicate not only with the Base Structure, but also with the packages themselves. So that, any new developed package should be able to use one others’ abilities to finalize its process. As some of the packages will use the other ones’ processes, the structure will not need to develop same packages again and again so that a very complex structure will be created and it will have a very fast progress.

2. **Web based:** The system is designed to be web based so that anyone would be able to use it in all over the world. When the user signs into the Engine once, the engine would remember the information about and by using the same password and User ID, he/she would be able to continue his saved analysis.

3. **Cross platform interoperable:** Engine will be used with any kind of operation system, Linux, Unix, Windows and any hardware, desktops, laptops, mobile tools, PDAs etc. This will make the system more practical and easy to use.

4. **Solving uncertainty problem:** One of the basic problems Raster GIS having is the uncertainty problem which can be defined as holding only one value for each pixel but it is impossible for the real world. As there are some solutions to this problem like fuzzy logic way, another solution to the problem can be R solution. Although the matrix way of resembling raster model has not been designed on purpose to solve this problem, it can hold more than one value for each pixel and can result in different kinds of new Raster GIS models to overcome this problem. Such kind of a new approach would be very useful for this project as GRASS and R are being used together.

5. **Module based structure:** The module based structure will help Engine to plug and use new R and GRASS Server Modules. Some other modules could be plugged easily to the Engine as well. With the module based structure, each module can be developed independently from the other one by means of both hardware and software, so that whilst using the Engine and the modules, the new module could be developed and then plugged in with no problems.

6. **User friendly & easy interpretation:** The model will be easy to use, so that all the archaeologists will be able to use the system without technical knowledge. Even the complex operations will be available
with pressing a single button, and the user does not need to know the back processes having being done.

For example, when an analysis with both statistics and GIS parts are planned to be done, one button press will be enough for the Engine so that the request will be sent to Interface first, and the interface will change this request to the appropriate R command. After identifying which R Server is more available by load balancing software, R command will be sent to that server. During the process, the GRASS Server can be used according to needs and the result will be sent back to Interface.

Thus, easy interpretation is one of the key points in this project. The interpretation would be done by Java in PM and Ajax and JavaScript codes in client side, according to the information sent back from interface, so that the user will get detailed interpretation about the processes by not using the resources of any server.

6. Detailed Reporting: After getting the detailed interpretations and the results of the operations done, an appropriate printing tool can be used.

IV. SIMILAR PROJECTS/ TECHNOLOGIES

1. PyWPS Project:

A WPS can be configured to offer any sort of GIS functionality to clients across a network, including access to pre-programmed calculations and/or computation models that operate on spatially referenced data. A WPS may offer calculations as simple as subtracting one set of spatially referenced numbers from another, or as complicated as a global climate change model. The data required by the WPS can be delivered across a network, or available at the server. This interface specification provides mechanisms
to identify the spatially-referenced data required by the calculation, initiate the calculation, and manage the output from the calculation so that it can be accessed by the client. This Web Processing Service is targeted at processing both vector and raster data. For more information: http://pywps.wald.intevation.org

Some PyWPS applications:

http://apps.esdi-humboldt.cz/klasifikace/

http://geo.sazp.sk/

http://dev.bnhelp.cz/inspire/client/

http://pywps.ominiverdi.org/demo_google.html

http://pywps.ominiverdi.org/ka-map/htdocs/index_wps.html

http://pywps.ominiverdi.org/ka-map/htdocs/index_wps_winman.html

http://pywps.ominiverdi.org/demo/embrio/ka-map/htdocs/index_wps_qgis.html

**PYWPS Raster Application:** This process is used to found the number of trees in an input raster map. You'll get the number of trees detected and a point for each one. Future version, will allow to upload raster on the server.

GRASS Minor cost path module (using r.walk): Find shortest path between 2 points using slope factor as cost value. [http://pywps.ominiverdi.org/subversion/trunk/web/embrio/raster/r_walk/r_walk.php](http://pywps.ominiverdi.org/subversion/trunk/web/embrio/raster/r_walk/r_walk.php)

GRASS Routing module (v.net.path): Find shortest path on vector network.

GRASS Buffer module: r.los generates a raster map output in which the cells that are visible from a user-specified observer location are marked with integer values that represent the vertical angle (in degrees) required to see those cells (viewshed).

GRASS Buffer module (v.buffer): Create a buffer around features of given type (areas must contain centroid).

2. Kamap
Ka-Map is an open source project that is aimed at providing a JavaScript API for developing highly interactive web-mapping interfaces using features available in modern web browsers. ka-Map has a number of interesting features like; interactive, continuous panning without reloading the page, keyboard navigation options (zooming, panning), zooming to pre-set scales, scale bar, legend and key map support, optional layer control on client side.

V. CASE STUDY

VI. CONCLUSION

VII. REFERENCES
Books

Basham, B., Sierra, K., Bates, B., 2008, Servlets & JSP, O’Reilly Media Inc.


A Grass + R + Python + Quantum GIS + PostgreSQL Approach

Bivand, R., 2009, Spgrass Library; Interface between GRASS 6 geographical information system and R


Crawley, M. J., 2005, Statistics; An Introduction using R, John Wiley & Sons Ltd


Eernisse, M., 2006, Ajax Web Applications, Sitepoint Pty Ltd

Heiman G. W., 1992, Basic Statistics for the Behavioural Sciences, Houghton Mifflin Co

Lewis, J., Loftus W., 2009, Java Software Solutions; Foundations of program design, Addison-Wesley/Pearson

Lutz, M., 2006, Programming Python, O’ Reilly Media Inc


Peng, Z. R., Tsou M. H., 2003, Internet GIS; Distributed Geographic Information Services for the Internet and Wireless Networks, John Wiley & Sons Inc.


Wheatley, D., Gillings, M., 2002, Spatial Technology & Archaeology, Taylor & Francis Group


Web References & Similar Projects

CASA Blog for GIS and Agent Based Modelling, http://gisagents.blogspot.com/

Enterprise Java Beans (EJB), http://java.sun.com/products/ejb

Version: 1.0

Status: Final
Grass Help Files, http://grass.itc.it/

Interface-PM Integration (Python), http://rpy.sourceforge.net/index.html

Java Server Pages (JSP), http://java.sun.com/products/jsp


OpenMap, http://openmap.bbn.com/

PyWPS (Python Web Processing Service), http://pywps.ominiverdi.org/

PYWPS Project, http://pywps.wald.intevation.org

PYWPS Project Documentation,

http://pywps.wald.intevation.org/documentation/how_it_works.html

QGIS Page, http://www.qgis.org

R Blog, http://romainfrancois.blog.free.fr/
