Journal of Uncertain Systems
Vol.3, No.2, pp.95-107, 2009
Online at: www.jus.org.uk

A Web Content Management System for a Geo-Archeological Research Program

Salvatore Rampone*
Regional Centre on Information and Communication Technology (CeRICT)
and Dipartimento di Studi Geologici e Ambientali (DSGA)
Università del Sannio, Via Port'Arsa 11, I-82100 Benevento ITALY

Received 28 October 2008; Accepted 27 December 2008

Abstract

This work describes the structure and the functionalities of an information system developed in the framework of a Geological research program. It allows data input and integration in a distributed area, offering advanced capabilities as web based content management system to cooperative groups in distributed sites. The underlying workflow is defined by means of a learning algorithm. The algorithm identifies a set of rules, expressed as boolean functions, from a set of examples. The system is on line at the address *http://metaponto.sci.unisannio.it/*.

Keywords: information system, distributed computing, learning algorithm, content management system, web

1 Introduction

The coastal plain of Metapontum, Italy (see Fig. 1), is quite interesting to appraise the interference among sedimentary processes, tectonic events, volcanic events and high frequency climatic changes during the Late Pleistocene-Holocene time. Actually this zone has an high subsidence rate which favours the preservation in the sediments of geologic and climatic information. Moreover there are some Magna Graecia settlements which constitute an important stratigraphic marker for the recent deposits [19, 18, 27, 9, 6].

This is the scenario of a recent research program called "Geological evolution of the Metapontum and Sybaris coastal plains during the Late Pleistocene - Holocene. Climatic factors and catastrophic events". The study collects several kinds of data including detailed geologic and geomorphic surveys, boreholes, georadar profiles, seismic profiles and core borings. Many analyses are carried out on epiclastic and pyroclastic deposits, regarding sedimentology, micropaleontology, petrography, tephro-stratigraphy, and geochemistry. Finally all these data are integrated by radioisotope 14 Carbon-dating and by several other archaeological analyses regarding the stratigraphic level corresponding to Magna Graecia time [1, 2, 4, 20] (see Fig. 2).

The main aim of the program is the individuation of elements in the Late Pleistocene-Holocene deposits which testify high frequency climatic changes so allowing the reconstruction of a climatic curve regarding not only the analyzed coastal plain but, probably, the whole Mediterranean zone [20, 8].

The need of interchange and integration of several different kinds of data, in some cases of great size, developed by the research groups in far away sites (see Acknowledgment section) implied the definition of a workflow to achieve the project tasks. Many scientific problems can be represented as computational workflows of operations that access remote data, integrate heterogeneous data, and analyze and derive new data. Usually the design of the workflow requires the involvement of at least two domain experts for each task - one from the scientific field of interest (e.g., a geophysicist) to specify how scientific products may be derived from data and another from the computer science, to understand the process of composing a workflow and define an information system [26]. Even when the data access and processing operations are implemented as web or grid services, workflows are often constructed manually in languages such as BPEL. While this is certainly useful, it has been evidenced that is not enough to model and construct complex data workflows [3]. In the

*Corresponding author. Email: rampone@unisannio.it (S. Rampone)

described scenario we applied a machine learning algorithm [22, 23] to identify the workflow as a set of rules, expressed as boolean functions, from a set of examples.

Then a supporting digital structure has been developed, allowing data exchange and integration in a distributed area by means of the Internet, while maintaining data security and privacy [29]. We describe the structure and the functionalities of this digital system.

The core of the system is a Web Content Management System. A Content Management System (CMS) is a software system used for content management. Content management systems are deployed primarily for interactive use by a potentially large number of contributors with a collaborative aim. The content managed includes computer files, image media, audio files, electronic documents and web content. A Web Content Management System (WCMS) is a content management system with additional features to ease the tasks required to transfer/share/publish data on the Internet [15, 11].

This sharing modality required the integration of several functionalities, including document identification, storage and retrieval, tracking, version control, and presentation. Here the word document means every file format, e.g., images, spreadsheets, word processing documents, and complex, compound documents as detailed multilayer maps and images, sometimes of several gigabytes (see Fig. 3).

The system has been successfully applied in the research program and it is reachable on line[1].

This paper is organized as follows: we describe the workflow definition process and the used learning algorithm in Section 2, the software architecture in Section 3, and the system usage in Section 4.

2 Workflow Definition

2.1 Coding

The system users have been classified in groups, on the basis of their activity in the project. Namely the groups are *Napoli*, *Sannio*, *Bari*, *Parma*, and *Museo*. Two further users, forming separate groups, are the System *Administrator*, and the *Project Leader* (see Fig. 4).

Then, by the interaction between the domain experts, we defined a coding scheme which forms the basis of workflow definition. The coding, inspired by the work of Spooner *et al.* [28], represents tasks as sequences of operations, and operations as n-ary binary arrays of actions. Namely we adopt the following scheme:

- The *Workflow* is represented as a sequence of different tasks.
- A *Task* is a sequence of closely related operations that are executed by the groups.
- *Operations* are the basic jobs that are concurrently executed by the different groups. There is significant access to common resources, and conflicts may occur when operations from different groups access the same resource simultaneously.
- The access of an operation to common resources is described by a set of *Actions*. Actions may include, for instance, Delete, Modify, Move, Update, Download, Check, Monitor, View, Share and so on.

The coding scheme contains also flags expressing dependencies from other operations and ordering such that, for instance, an action executes earlier than another, despite being later in the coding.

By using this coding we collected a set of *positive* instances, each one indicated as the vector

$$\mathbf{w}_i^+ = w_{i,1}^+, w_{i,2}^+, \ldots, w_{i,n}^+, \quad w_{i,k}^+ \in \{0,1\} \tag{1}$$

and each one denoting a possible and admissible operation for a user c in a certain system status, where $c \in \{$*Napoli*, *Sannio*, *Bari*, *Parma*, and *Museo*$\}$.

At the same time we generated a set of *negative* instances, each one indicated as the vector

$$\mathbf{w}_j^- = w_{j,1}^-, w_{j,2}^-, \ldots, w_{j,n}^-, \quad w_{j,k}^- \in \{0,1\} \tag{2}$$

and denoting a possible but not admissible operation for a user c in a certain system status.

[1]The project site. http://metaponto.sci.unisannio.it/

Figure 1: Road map of the coastal plain of Metapontum, Italy

2.2 Boolean Rules

Given the previous coding, our problem has been posed as finding a set of Boolean classification rules f_c such that $f_c(\mathbf{w}) = 1$ if $\mathbf{w}$, a window of n binary values, called *instance* $\mathbf{w}$, corresponds to an admissible operation for a certain user c, and $f_c(\mathbf{w}) = 0$ otherwise. In the following we report some basic definitions related to these rules.

A *Boolean classification rule* (briefly *function*) f_c on n *variables*

$$x_1, x_2, \ldots, x_n, \tag{3}$$

each one taking only the values 1 or 0, is a mapping from the n-dimensional *instance space* $\{0,1\}^n$ to $\{0,1\}$

$$f_c : \{0,1\}^n \to \{0,1\}. \tag{4}$$

An *instance* can be viewed as a value assignment to the variables in f_c. An instance $\mathbf{w}_i^+$ is *positive* if $f_c(\mathbf{w}_i^+) = 1$, i.e. if $\mathbf{w}_i^+$ belongs to the class c. An instance $\mathbf{w}_j^-$ is *negative* if $f_c(\mathbf{w}_j^-) = 0$, i.e. if $\mathbf{w}_j^-$ does not belong to the class c.

A Boolean function is *consistent* with a set of instances if and only if it matches every positive instance and no negative instance in the set.

A variable is in *true form*, and it is indicated as the *literal* x_k, if it assumes value 1 when it is assigned to 1, and 0 otherwise. On the contrary, a variable is in *negated form*, and it is indicated as the literal $\overline{x}_k$, if it assumes value 0 when x_k is assigned to 1, and 1 otherwise.

A *term* m is the conjunction of any subset of the n variables $x_1, x_2, \ldots, x_n$, each one in true or negated form. A term is *made true* by an instance if each of its variables is 1 by assigning the corresponding instance

Figure 2: Doric temple of “Tavole Palatine” in Metaponto.

Figure 3: An orthophoto of the site.

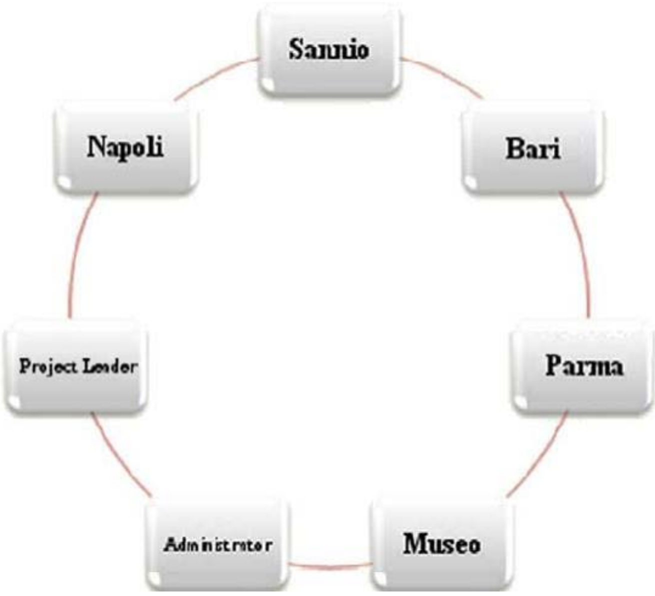

Figure 4: User groups.

values. In the same way a term is *not made true* by an instance if at least one of its variables is 0 by assigning the corresponding instance values.

A Boolean function is expressed as a *DNF* (disjunctive normal form) formula if it is expressed as a disjunction (or) of terms. Given an instance, it assumes value 1 if at least one term of the function is made true by the assignment, and it assumes value 0 otherwise. An L-term DNF function can be represented as

$$f_c = m_1 + m_2 + \ldots + m_L. \tag{5}$$

2.3 Learning Algorithm

Our goal was to find a rule which captures the underlying systematic aspects of the instances without overfitting on the instances at hand. To this aim we applied a known learning algorithm, called BRAIN, originally conceived for the prediction of splice site locations in human DNA [22, 23][2]. This algorithm given a set of training data, returns a DNF function which is consistent with all the training instances (provided that we have no contradictory instances), and, by applying the Occam's Razor [10], of approximately minimum size. The algorithm builds iteratively the function terms by means of a probability distribution over the literals derived from the training data.

Namely, let us consider p positive instances and q negative ones

$$\mathbf{w}_1^+, \mathbf{w}_2^+, \ldots \mathbf{w}_p^+, \mathbf{w}_1^-, \mathbf{w}_2^-, \ldots \mathbf{w}_q^- \tag{6}$$

and let us define the sets of variables

$$S_{i,j} = \{x_k | w_{i,k}^+ = 1, w_{j,k}^- = 0\} \cup \{\overline{x}_k | w_{i,k}^+ = 0, w_{j,k}^- = 1\} \tag{7}$$

that we collect as

$$S_i = \{S_{i,1}, S_{i,2}, \ldots, S_{i,q}\}. \tag{8}$$

[2]The optimized version FastBRAIN is available on-line. http://brain.sci.unisannio.it/

The $S_{i,j}$ sets represent constraints: to be consistent with a positive instance i and a negative instance j, a term must include at least a variable $v_k \in S_{i,j}$, and to be consistent with a positive instance i and with all the q negative ones, a term must include at least a variable for each $S_{i,j} \in S_i$.

In this way selecting a minimal set of terms whose union is consistent with all the instances, means selecting a minimal set of terms satisfying all the constraints S_i. To this aim the relative frequencies of a the variables in the $S_{i,j}$ sets are computed as

$$R(v_k) = \frac{1}{pq} \sum_{i=1}^{p} \sum_{j=1}^{q} R_{i,j}(v_k), \tag{9}$$

where

$$R_{i,j}(v_k) = \frac{I_{S_{i,j}}(v_k)}{\#(S_{i,j})} \tag{10}$$

and where $I_{S_{i,j}}(v_k)$ is the characteristic function of the $S_{i,j}$ set, that is 1 if $v_k \in S_{i,j}$ and is 0 otherwise. The $R(v_k)$ coefficients, called *relevances*, form a probability distribution.

The BRAIN algorithm uses the relevance as a greedy criterion to build DNF function terms. Starting from an empty term, it select the most relevant variable, and add it to the term, erasing all the satisfied constraints $S_{i,j}$ and all the incompatible S_i sets, until at least a set S_i of constraints is satisfied (empty). This greedy choice aims at the same time both to cover greatest number of positive instances, and to select the less possible number of variables. The process is iterated until there are no more constraints.

An algorithm scheme follows. In the reported scheme the original genomic bias has been removed.

BRAIN Algorithm

Step 1: *Input:* n= variable number,
$G = \{\mathbf{w}_1^+, \mathbf{w}_2^+, \ldots \mathbf{w}_p^+, \mathbf{w}_1^-, \mathbf{w}_2^-, \ldots \mathbf{w}_q^-\}$.
the set of training instances.
p = positive instance number.
q = negative instance number.
Initialization: Set $f_c = \emptyset$.

Step 2: While there are positive instances in G

2.1 *$S_{i,j}$-sets:* Build from G the sets $S_{i,j}$ and collect them in S_i's.
2.2 *Start a new term:* Set $m = \emptyset$.
2.3 *Build the term :* While there are $S_{i,j}$ sets
2.3.1 *Relevances:* Compute the relevances $R(v_k)$.
2.3.2 *Add variable:* Select the variable v_k such that $R(v_k)$ is maximum. $m \leftarrow m \cup \{v_k\}$.
2.3.3 *Update sets:* Erase the S_i sets not including v_k. Erase the $S_{i,j}$ sets including v_k.
2.4 *Add the term:* $f_c \leftarrow f_c + m$.
2.5 *Update instances:* Erase from G the positive instances satisfying m.

Step 3: *Output:* f_c.

In this way we generated the rules of thumb for each group of users.

3 Architecture

The main functions of our system are to put to disposition and to share via Internet files of varied nature to heterogeneous clients and to consent a collaborative work. So the application has been conceived as a client server one [25] (see Fig. 5).

Since the end user machines are quite heterogeneous, to the aim of reaching the system by the Internet without any particular client, the structure is centred around a *Web Server* [13], i.e. a computer that delivers (serves up) Web pages. By both financial and reliability reasons, it is based on *Apache*[3] [12, 16], a public-domain open source web (HTTP) server. It is actually the world's most widely employed web server.

[3] Apache Software Foundation. http://www.apache.org/

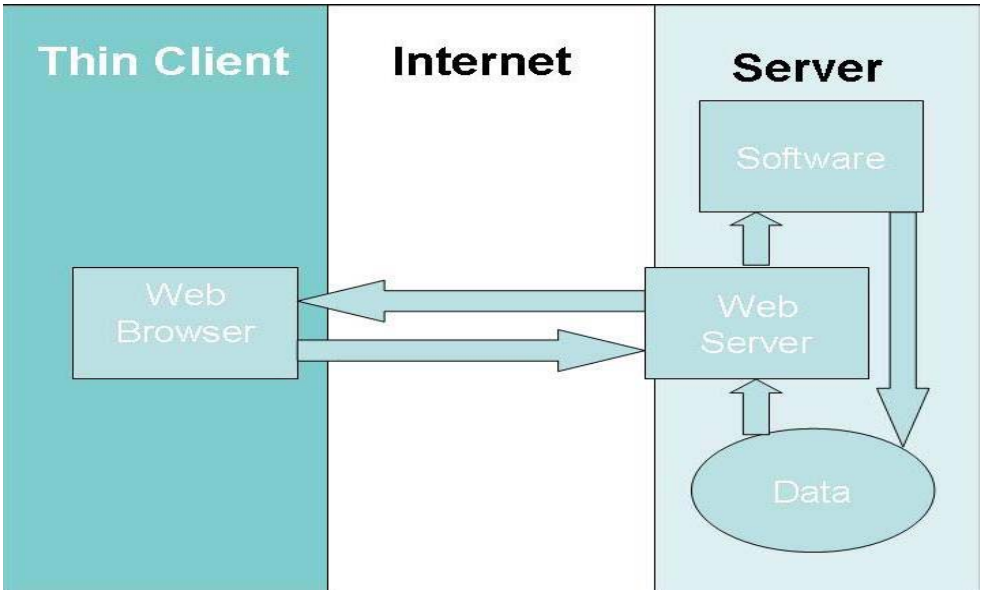

Figure 5: Client/server schema.

3.1 Application Requirements

Our application has been conceived as divided in two parts: a *public area*, accessible to every Internet user, and including informative HTML pages, and a *restricted area*, accessible to the research groups, to be used as a content management system. For this last part we evidenced the following requirements:

1. Provide a consistent and predictable information structure, and navigational mechanism, ensuring information is up-to-date, and enabling management, indexing, revision and previous version maintaining of the materials;
2. Give a user friendly customizable interface;
3. Facilitate non-technical authors in the content management from any web browser;
4. Enable workflow between groups supported by the roles, responsibilities, and access control defined by our DNF formulae;

One further goal was developing a low-maintenance information system, easily managed by end users. Responsibilities are delegated and dispersed to each group.

3.2 Content Management System

The application required a web based CMS engine [7]. Our choice is OWL[4], a multi user content management system written in PHP [17]. It satisfies the specified requirements with the further advantages to be released under GNU General Public License (GPL), and to be independent system, since it is written in an interpreted language.

The CMS requires an underlying external database management system (DBMS), supporting storage and retrieval. To this aim we used MySQL[5] [24]. MySQL is an open source *RDBMS,* short for *Relational DataBase Management System* [5]. Its speed, scalability and reliability make it the right choice for our application. It

[4]OWL documentation. http://owl.sourceforge.net/
[5]MySQL documentation. http://www.mysql.com/

is actually the world's most popular open source database, since it is an attractive alternative to higher-cost database technology.

Given the potential high size of our data, and for performance and scalability reasons, the data persistence is maintained also by the server file system.

A scheme of the overall organization is reported in Figure 6.

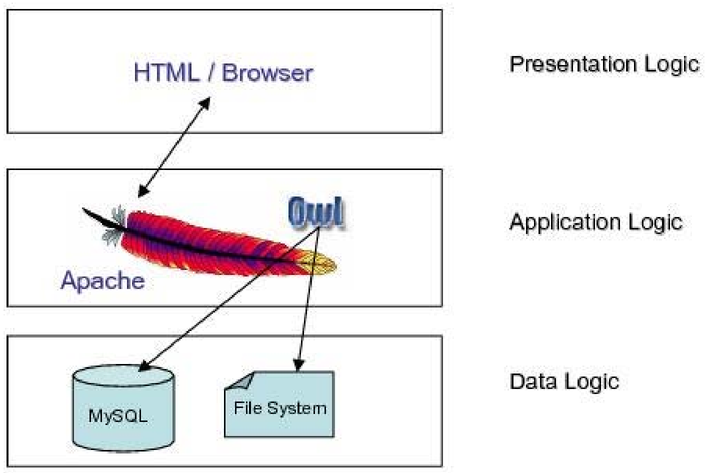

Figure 6: Overall logic organization.

4 Using the system

4.1 Access Management

The resulting system is reachable via net by any Internet browser. In order to use the CMS facilities a login procedure (Fig. 7) is required.

The CMS access is restricted as follows :

- The user of the system must be registered by the administrator (although a limited access as an anonymous user can be allowed)
- Every user belongs to at least one group, the Primary group
- Every user can also belong to additional groups, Member Groups

The users and groups are actually the research groups indicated as *Napoli*, *Sannio*, *Bari*, *Parma*, and *Museo*, respectively.

4.2 System Administration

The administration of the system (Fig. 8) is also web based (remote administration) and demanded to the *System Administrator* and *Project Leader* users. The administrative options include

- Appearance Settings, used to adjust html preferences of the system and to change the visual appearance.

- System Preferences, to set the configuration of the main tasks and to enable/disable features. - Logging, to examine what the users are doing on the system installation.

Figure 7: Login screen.

- Statistic System allows to view the statistics related to the system use.
- News Administrator, to publish news to the users and change already published news.
- Trash Collector helps to recover deleted files.

4.3 User Interface

The system main screen is the File Explorer Interface (Fig. 9). Here the user can browse all the files and folders which he/she has access to. There are also four basic operations:

- Read - the ability to have access to a document within the system.
- Write - the ability to update or change a document within the system such as uploading a new version, checking it out, checking in a new version, editing its properties such as title or keywords.
- Delete - the ability to remove a document from the system.
- Download - the ability to transfer the document to the user local system.

4.4 Operations with Files

The system offers the user multiple operations to do with the files, and everyone uses a very simple dialog, common to the different operations. The drop down list "Permissions" defines what a user in the system can do once a file has been submitted. Namely the operations are of Adding Files, Move, Update, Properties and Mail Files.

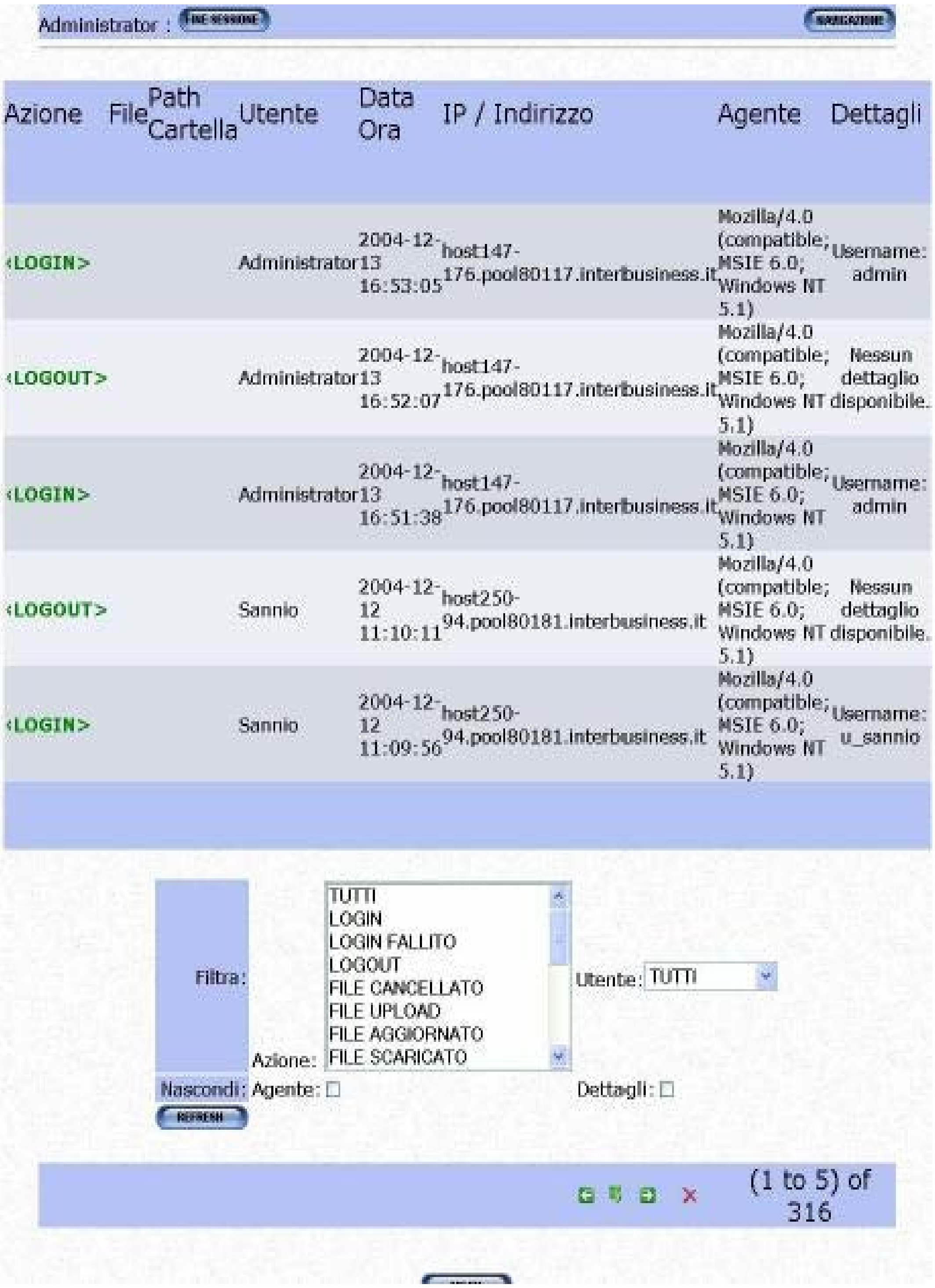

Figure 8: Statistic system screen.

Figure 9: File explorer screen.

5 Conclusions

In this work the structure and the functionalities of a file server supporting a research program workflow have been described. The server allows data input and integration in a distributed area and offers an easy way to use web based content management system with a flexible and powerful set of user, group and permission options.

We reviewed the workflow definition, the architecture choices and the daily usage of the system, as an user or as an administrator, with a description of the main options. The system has a very simple user interface and it is very easy to use. Further details can be obtained from the content management system on line documentation.

While the system has been developed and tailored for a specific geoarchaeological research program, it appears suitable to all the cases when the research activity requires data input, manipulation and integration in a distributed area.

The almost all open source software choices, based on the classical LAMP (Linux, Apache, MySQL, and PHP) environments as base, on the BRAIN algorithm to define the system rules, and on the OWL CMS make the system attractive also from a budget point of view.

The system is on line at the address *http://metaponto.sci.unisannio.it/*. In the first few months of experimentation the overall throughput remains stable without excessive decay in stressing conditions [14, 21].

Acknowledgments

This work has been developed in the framework of the research program “Geological evolution of the Metapontum and Sybaris coastal plains during the Late Pleistocene - Holocene. Climatic factors and catastrophic events”, sponsored by Ministero dell’Istru-zione, dell’Università e della Ricerca, finanziamento Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale (PRIN) Anno 2003.

The following research institutions were involved in the program: Dipartimento di Scienze della Terra - Università di Napoli Federico II, Dipartimento di Studi Geologici e Ambientali - Università del Sannio, Dipartimento di Geologia e Geofisica - Università di Bari, Dipartimento di Scienze della Terra - Università di Parma, Soprintendenza per i Beni Archeologici della Basilicata - Museo Nazionale Archeologico di Metaponto

I would like to express my sincere appreciation to the project leader, Prof. T. S. Pescatore who is retiring from the Università del Sannio after many years of outstanding activity, for his invaluable support and encouragement.

I would like to thank Eng. V. Morfino for his help during the CMS configuration and Dr. M. Franco for useful discussions.

References

[1] Abdullayev, N.R., Seismic stratigraphy of the Upper Pliocene and Quaternary deposits in the South Caspian Basin, *Journal of Petroleum Science and Engineering*, vol.28, pp.207-226, 2000.

[2] Abrams, M.A., and A.A. Narimanov, Geochemical evaluation of hydrocarbons and their potential sources in the western South Caspian depression, Republic of Azerbaijan, *Marine and Petroleum Geology*, vol.14, pp.451-468, 1997.

[3] Ambite, J.L., and D. Kapoor, Automatically Composing Data Workflows with Relational Descriptions and Shim Services, *Proceedings of ISWC*, 2007.

[4] Anderson, A.L., and W.R. Bryant, Gassy sediment occurrence and properties: Northern Gulf of Mexico, *Geo-Marine Letters*, vol.10, pp.209-220, 1990.

[5] Atzeni, P., S. Ceri, *et al.*, *Database Systems - Concepts, Languages and Architectures*, McGraw-Hill, Boston, 1999.

[6] Auriemma, R., G. Mastronuzzi, and P. Sansò, I siti archeologici costieri pugliesi come markers delle variazioni del livello del mare, *In: Ambiente e paesaggio nella Magna Grecia, XXXXII Convegno di Studi sulla Magna Grecia, Taranto 5-8 ottobre 2002, Istituto per la Storia e l’Archeologia della Magna Grecia, Taranto*, pp.515-521, 2003.

[7] Balasubramanian, V., and A. Bashian, Document management and web technologies: Alice marries the Mad Hatter, *Communications of the ACM*, vol.41, pp.107-115, 1998.

[8] Becker, J., D. Menzel, *et al.*, High-frequency climate interactions between the Mediterranean and the Atlantic during Marine Oxygen Isotope Stage 100, *Geophysical Research Abstracts*, 5, 08891, 2003.

[9] Bianchini, G., E. Marrocchino, *et al.*, Chemical-mineralogical characterization of historical bricks from Ferrara: an integrated bulk and micro-analytical approach, *Geological Society*, vol.257, pp.127-140, 2006.

[10] Blumer, A., A. Ehrenfeucht, *et al.*, Occam's razor, *Inf. Process. Lett.*, vol.24, no.6, pp.377-380, 1987.

[11] Cleveland, G., Overview of document management technology, *Network Notes*, vol.9, pp.1-4, 1995.

[12] Coar, K., and R. Bowen, *Apache Cookbook*, O'Reilly, 2003.

[13] Comer, D.E., *Internetworking with TCP/IP*, Prentice Hall, Englewood Cliffs, 1995.

[14] Curtin, D.P., K. Foley, *et al.*, *Information Technology - The Breaking Wave*, McGraw-Hill, Boston, 1998.

[15] Kampffmeyer, U., *ECM Enterprise Content Management, Definitions, Scope, Architecture, Components and ECM-Suites in English, French, and German*, Hamburg, Project Consult, 2006.

[16] Laurie, B., P. Laurie, *Apache: The Definitive Guide*, O'Reilly, Sebastopol, 2003.

[17] Lerdorf, R., K. Tatroe, *Programming PHP*, O'Reilly , Sebastopol, 2002.

[18] Marchesini, L., A. Amorosi, *et al.*, Sand composition and sedimentary evolution of a Late Quaternary depositional sequence, northwestern Adriatic coast, Italy, *Journal of Sedimentary Research*, vol.70, pp.829-838, 2000.

[19] Morabito, S., Late Pleistocene-Holocene paleoclimatic and paleoenvironmental changes in the Ionian Sea (ODP Hole 964B) as revealed by planktonic foraminifera and calcareous nannofossils assemblages, Geophysical Research Abstracts 9, SRef-ID: 1607-7962/gra/EGU2007-A-05258, 2007.

[20] Pescatore, T.S., Storia geologica delle piane costiere di Metaponto e di Sibari durante il Pleistocene superiore - Olocene, Fattori climatici ed eventi catastrofici, http://metaponto.sci.unisannio.it/, 2004.

[21] Raj, J., *The Art of Computer Systems Performance Analysis*, John Wiley and Sons, New York, 1991.

[22] Rampone, S., Recognition of splice-junctions on DNA sequences by BRAIN learning algorithm, *Bioinformatics*, vol.14, pp.676-684, 1998.

[23] Rampone, S., An error tolerant software equipment for human DNA characterization, *IEEE Transactions on Nuclear Science*, vol.51, no.5, pp.2018-2026, 2004.

[24] Reese, G., *My SQL Pocket Reference: SQL Functions and Utilities, Pocket Reference*, O'Reilly, 2007.

[25] Renaud, P.E., *Introduction to Client/Server Systems*, Second Edition, John Wiley & Sons, 1996.

[26] Salayandia, L., P. Pinheiro da Silva, *et al.*, A model-based workflow approach for scientific applications, *Proceedings of the 6th OOPSLA Workshop on Domain-Specific Modeling*, Portland, Oregon, USA, 2006.

[27] Scarponi, D., and M. Kowalewski, Sequence stratigraphic anatomy of diversity patterns: Late Quaternary benthic mollusks of the Po plain, Italy, *Palaios*, vol.22, pp.296-305, 2007.

[28] Spooner, D.P., J. Cao, *et al.*, Performance-aware workflow management for grid computing, *The Computer Journal*, vol.48, no.3, pp.347-357, 2005.

[29] Tanenbaum, A.S., *Computer Networks*, Prentice Hall, Englewood Cliffs, 2003.