Journal of Web Engineering, Vol. 6, No.3 (2007) 261-280
© Rinton Press

AN INTEGRATED TECHNIQUE FOR WEB SITE USAGE SEMANTIC ANALYSIS: THE ORGAN SYSTEM

JOHN GAROFALAKIS[1,2], THEODOULA GIANNAKOUDI[1,2] and EVANGELOS SAKKOPOULOS[1,2]

[1]Department of Computer Engineering & Informatics
School of Engineering, University of Patras
Rio Campus, 26500 Patras, Greece
[2]Research Academic Computer Technology Institute
Internet and Multimedia Technologies RU 5 and Telematics Center
N. Kazantzaki str. 26500, Greece
{garofala, gianakot, sakkopul }@ceid.upatras.gr

Received January 15, 2007
Revised March 26, 2007

In this work, a new log analysis system is proposed and implemented, called ORGAN (Ontology-oRiented usaGe ANalysis system). ORGAN aims to enhance and ease log analysis by using semantic knowledge.It is able to offer typical statistical analysis of Web usage logs taking into consideration at the same time site's underlying semantics. We evaluated ORGAN using Web site data for different cases to verify and exhibit its promising behavior. The experimental outcomes were encouraging and valuable conclusions for the Web site usage under analysis were reached. Consequently, we believe and show paradigms that ORGAN could become a useful tool for Web log analysts and assist the Web site managers in the decision-making for reorganization tasks. Finally, we discuss open problems to motivate further research efforts towards the incorporation of semantic Web technologies into Web site log mining analysis.

***Key words*:** Web Usage Mining, Web Traffic Analysis, Knowledge Acquisition, OWL ontology

1 Introduction

In the recent years, the research field of Web usage mining has gained notable consideration. A large number of log statistical analysis tools and applications have been developed either for commercial use or for research purposes (e.g. Web Trends [28], Analog [26], Weblogs [29] and SurfStats Log analyzer [27]). All these systems aim at revealing the knowledge hidden in the log files, so as to extract interesting patterns about the Web sites' visitors' preferences. However, their efficiency is limited, mostly, because of the initial unilateral approach of the problem, which used to take under consideration the Web sites logs or structure, but not its content. As it has been thoroughly analyzed in [6], the exploitation of the Web site content is considered a critical input to the pre-processing algorithms that can be used as filter before and after pattern discovery algorithms, and can provide useful information about usage patterns.

From that point on, Web site content has started to play a more important role in the Web usage mining process. In fact, some recent works have proposed new approaches that integrate site semantics with its usage data ([8], [16]). Actually, there is more in a Web site than its structure and content, the underlying semantic information. In [7], a survey of approaches for incorporating semantic knowledge into Web usage mining and personalization processes has been discussed. They discuss "*the issues and requirements for successful integration of semantic knowledge from different sources, such as the content and the structure of Web sites for personalization*". However, there are no integrated tools presented for Web site semantic log analysis that could be delivered as end-user applications to help the Web site administrator, designer or even an analyst.

In this work, a new log analysis system is proposed and implemented, called *ORGAN. Ontology-oRiented usaGe ANalysis system* aims to enhance and ease log analysis by exploiting semantic knowledge. ORGAN integrates a number of roles:

a) it facilitates traditional Web usage analysis,

b) it assists the detection of domain knowledge and its assignment on a well-known domain ontology for the Web site at hand and, finally,

c) it combines both of them in order to answer combined semantically enhanced queries about the Web site usage.

The proposed cycle of Web site traffic semantic analysis includes the Web log files and Web site content preprocessing tasks, the semantic annotation task, and the semantically-enhanced query mechanism. In particular, the Web site logs are cleaned and user sessions are extracted from them in order to detect user preferences. This information is used for the determination of a page's popularity taking under consideration the visiting frequency (page access).

Regarding the Web site content preprocessing and semantic annotation, mining techniques are applied to match site content with the domain knowledge, which is represented in the standard, semantic classes of an OWL ontology. Exalting to metadata descriptions contributes to the definition of semantic criteria in a standardized way for the web pages that will be log analyzed. Moreover, we support the integration of a related OWL based ontology in order to enable and take advantage of semantic inference during log analysis. Thus, the web traffic analysis will not concern anymore plain URLs, but Web entities with semantic descriptions that will give an insight in the users' preferences.

After preprocessing, a combined analysis of semantically enriched statistical queries on the pre-processed raw data may be performed. Our usage mining combines the visit ratio of web pages with their access scores and the underlying latent semantic topics in the web page in order to detect popular interconnected visits. Indications of web traffic reports that an analyst may get through this procedure are:

- Comparison of usage of pages with relevant subjects based on their semantics
- Detection of semantic groups of pages that should be linked in the site graph
- Detection of pages with increased popularity among users visiting different semantic groups of pages

ORGAN constitutes an integrated and standalone system, which is available for semantic log analysis and can be utilized for any Web site when combining the appropriate domain ontology (academic sites, e-shops, commercial sites etc). It has been experimentally tested using several datasets for different cases from the literature and academia in order to verify and exhibit its behavior.

The presentation of our proposed semantically enhanced log analysis tool is organized as follows: In section 2, our motivation for ORGAN and related work are presented. In section 3, the system architecture is described in order to provide a roadmap for the reader into system details. Next, in section 4, the Web site content processing methodology is further analyzed to highlight the details of our approach. In section 5, the semantic annotation process is presented. Following in section 6, the log preprocessing steps are described. In section 7, the system functionality and front-end are outlined. In section 8, the implementation details are presented. In section 9, the experimental use of ORGAN is discussed to evaluate system's efficiency. Finally, section 10 concludes the paper including future work directions.

2 Motivation and Related Work

User visits' analysis is the first step in any kind of Web site evaluation procedure either when it involves re-design and reorganization or not. Generally, the process of discovering useful information from Web logs is usually called log mining [17] or Web usage mining [6]. Log mining includes straightforward statistics methods, such as page access frequency, as well as more sophisticated forms of analysis, such as association rule mining, sequential pattern mining, clustering, etc. Our intention is to make a step forward beyond the available statistical analysis reports and to provide higher fidelity in the analysis results using the domain semantic underlying knowledge.

Our interest in this work focused on the inference extraction for the Web site visitors' preferences based on the semantics of the visited Web pages content, apart from the content itself and their navigation behavior. Considering all the potential information that a Web site may provide about its topics, its visitors and their needs, it is interesting to exploit the raw data, so as to give the possibility (especially for the Web site administrator) of finding answers to semantic queries involving more generalized notions and topic knowledge. Our aim is to provide a tool able to pose on logs semantically enhanced queries such as "What users ratio visits Web pages relevant to a specific topic x and Web pages relevant to a specific topic y?" to detect whether an important visit ratio exists between different semantic groups of Web pages within a single session. To orchestrate this goal, ORGAN integrates in a novel solution semantic Web technology onto a log analyzer tool to provide the ability of expressing semantically oriented queries together with typical log mining processing.

Researchers have already presented several approaches to the Web site traffic analysis. There has also been significant scientific work in the field of Web structure and link analysis, however the content based analysis attempts are much less. Some early work in the area has been presented by Chen et al [3]. Their work focuses on the extraction of user behavioral patterns from log files. Work on user path analysis in a Web site has been done by Berkhin et al [1]. Their goal was to understand visitors' navigation within the site. Later, Srikant et al [22] proposed an algorithm, which aimed to solve the problem by automatically discovering all pages in a Web site whose location is different from the location where visitors expect to find them. Several years ago, structural analysis of the hypertext graph and corresponding metrics have been presented by Botafogo et al [2]. Garofalakis et al have also early presented metrics [10] for re-organization as well as more and multiple enhancements in their approach of [4]. Recently, Jansen [15] presented critical terms for conducting transaction log analysis.

A Web usage analysis tool milestone has been set in the first years of WWW conference by Pitkow and Bharat with the WebViz tool [21], where a first Web usage analysis tool was presented by researchers to the global IT community. However, their log analysis could not result in the exact user paths and a client based enhancement, WebQuilt, was presented in [13] that introduced the notion of client side logging. Unfortunately, such an idea could not be and it is not widely adopted. On the other hand, ORGAN is an application that tries to enhance the Web usage analysis by the use of the underlying latent semantic topics in a Web site. The aim is to enable the site analyst to detect possible interconnected visits in "related" sections of his/her site. For instance, two courses, such as "Software languages and programming principles" and "OO programming", each consisting of several pages are possible to be both frequently co-visited. However, the standard analysis and mining is unable to find the relationships between these two topics. To overcome the above problems, one can utilize existing taxonomies, such as hierarchical clustering of content and site directory or even enhance them further in order to enable semantically enhanced log analysis queries.

An older solution that tried to take advantage of site context and taxonomies in order to provide Web site personalization is the SEWeP system [8]. Specific methodologies proposed in [8] are utilized by ORGAN, too, during the phase of initial Web site content processing. However, our system enhances the Web site content annotation with metadata, involving a domain ontology, instead of a site-specific terms taxonomy used in [8]. In ORGAN, any OWL ontology is supported in order to be used for the Web site at hand, suffice it to define relevant instances for the ontology classes. In this way, the correlations between the entities and their properties are taken into consideration. Moreover, in this way, we benefit from the use of semantic relationships by having the ability to detect semantic groups of Web pages that are related by more complex associations. For example, using a domain ontology for a computer science department for the annotation of the Web pages of a corresponding Web site, we may eventually detect clusters of pages that are visited together and refer to professors that teach courses of the "Software" sector. This kind of group of pages would never be detected through a classification method with taxonomies of terms and we would be limited to the detection of groups of pages that refer either to professors in general, or to software courses.

3 ORGAN Architecture

In this section, the ORGAN modules will be introduced. Our aim is to outline a roadmap of the available functionality and ease the readership. ORGAN analyses the usage of a Web site in relation to the Web site's semantic features, as they are expressed through an OWL ontology, relevant to the domain knowledge of the Web site. ORGAN utilizes knowledge, which is mined from three different data recourses: a) the Web site content, b) the usage data and c) an ontology representative of the Web site domain.

The Web site content provides the semantics of its Web pages and its structure, as well. The usage data contain information about the average number of hits from individual users visits to every Web page in proportion to the whole set of distinct users visits. The ontology is used for the assignment of the Web site content to standard classes, so as to ensure homogeneous annotation of the Web pages. This semantic annotation differs from the superficial information structuring of a Web Site. Ontologies are able to provide an objective specification of domain information by describing the kinds of entity involved in it and the relationships that can hold among different entities. Thus, they allow making explicit domain assumptions and support precise automated reasoning about the content they are used to annotate.

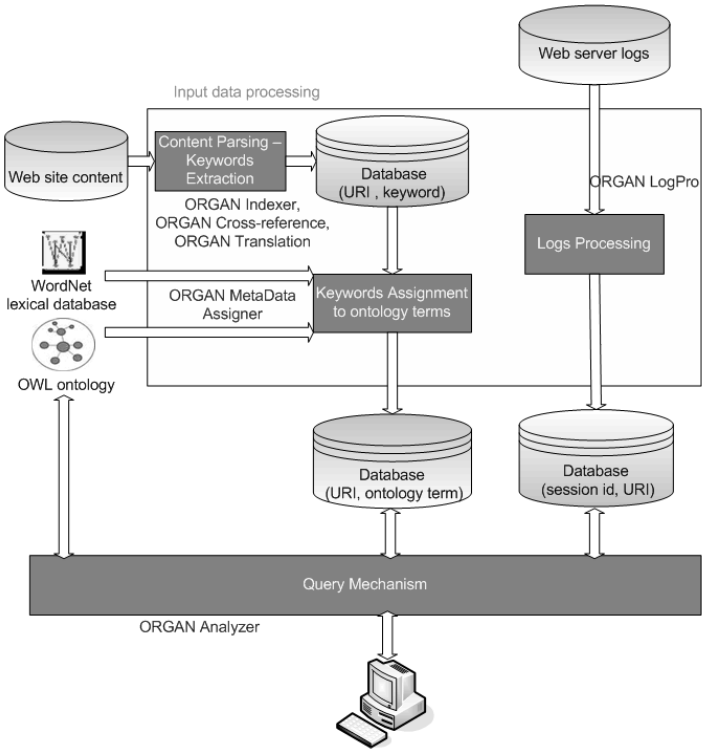


Figure 1 ORGAN Architecture

ORGAN consists of two different functional parts. The first part deals with the processing of the input data (Web site content and usage data), so as to build the appropriate knowledge base, that will be used from the query mechanism of the subsequent part. The initial component-which is utilized only once for a specific Web site and it will be re-used only if the Web site content changes- includes two independent sub-components. In the first sub-component, the functionalities that take place serially are: the Web site parsing, the extraction of keywords and the assignment of the keywords to the ontology classes and instances. In the second sub-component, the log files pre-processing and the sessions' extraction are carried out. In short, the modules dealing with the raw data processing are:

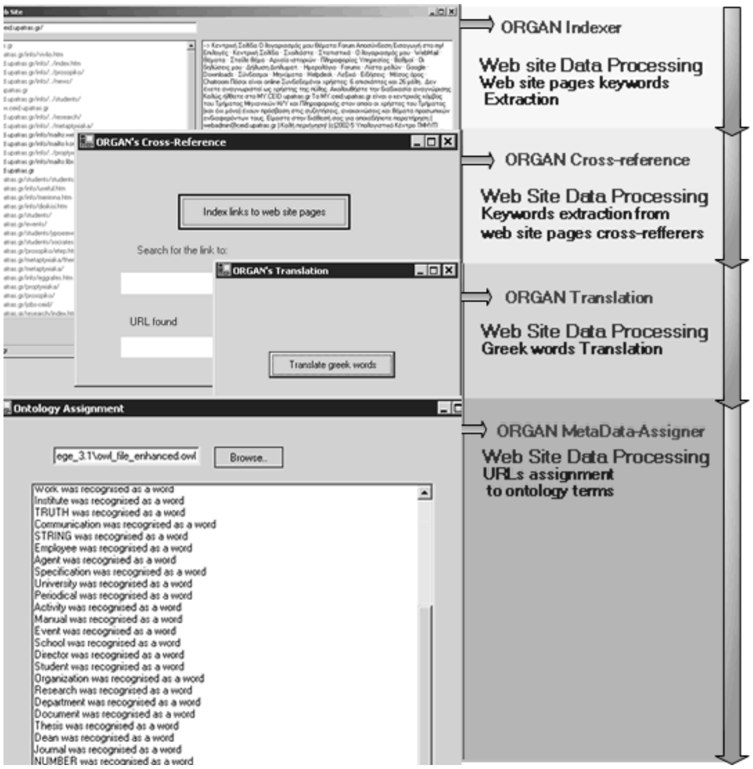
- ORGAN Indexer, process extracts keywords from the Web page body and from the URLs out of the Web site's domain to which this page provides links.
- ORGAN cross-reference, which extracts keywords from Web site pages popular cross-referrers
- ORGAN Translation, which translates greek words in the case of non English written pages
- ORGAN MetaData-Assigner, which applies appropriate mechanisms of word and phrases assignment, so as to semantically annotate the Web site pages - represented by keywords sets- with ontology terms.
- ORGAN LogPro, which processes raw usage data and extracts user sessions.

The second ORGAN part, which is called ORGAN Analyzer module, constitutes the system interface for the end-user. It enables a combined analysis of semantically enriched statistical queries on the pre-processed raw data.

The system architecture is presented in figure 1.

The modules, developed to deal with data processing functionalities, are presented in Figure 2.

As it is shown in figure 1, the Web site content processing must follow a serial execution, whereas the log files pre-processing may be carried out independently.



```
Work was recognised as a word
Institute was recognised as a word
TRUTH was recognised as a word
Communication was recognised as a word
STRING was recognised as a word
Employee was recognised as a word
Agent was recognised as a word
Specification was recognised as a word
University was recognised as a word
Periodical was recognised as a word
Activity was recognised as a word
Manual was recognised as a word
Event was recognised as a word
School was recognised as a word
Director was recognised as a word
Student was recognised as a word
Organization was recognised as a word
Research was recognised as a word
Department was recognised as a word
Document was recognised as a word
Thesis was recognised as a word
Dean was recognised as a word
Journal was recognised as a word
NUMBER was recognised as a word
```

Figure 2 Web site data processing modules

Next, the different modules that constitute the system are discussed in detail.

4 Site Content Preprocessing

4.1. Keywords Extraction

During the initial Web site content analysis, ORGAN follows a methodology similar to the one proposed in [8]. In this work, keywords for every Web page were extracted from the Web page itself, the Web pages it references and the Web pages that cross-reference it. However, we refined this methodology in order to take into consideration an attribute that affects the keywords value, the Web site structure itself. The Web sites tested in our system and more contemporary Web sites, as well, give access barely to every Web site page from any node of the site. As a result, using the site pages that are referenced by the current Web page to determine the page's content is not efficient in this case. It leads to the extraction of keywords that characterize the Web site content in general, but not the content of the particular Web page. The utilization of the Web page's content, the content of Web pages out of the site domain that are cross-referenced by the Web page and the anchor text areas of Web pages that reference it seem to serve the system purpose well and characterize the Web page's content adequately.

Consequently, the set of keywords for every Web site page is extracted as follows:

1. keywords are extracted from the Web page's content
2. more keywords are extracted from the Web pages that are cross-referenced by the specific page and don't belong to the Web site's domain
3. additional keywords are extracted from the pages, which cross-reference the specific page

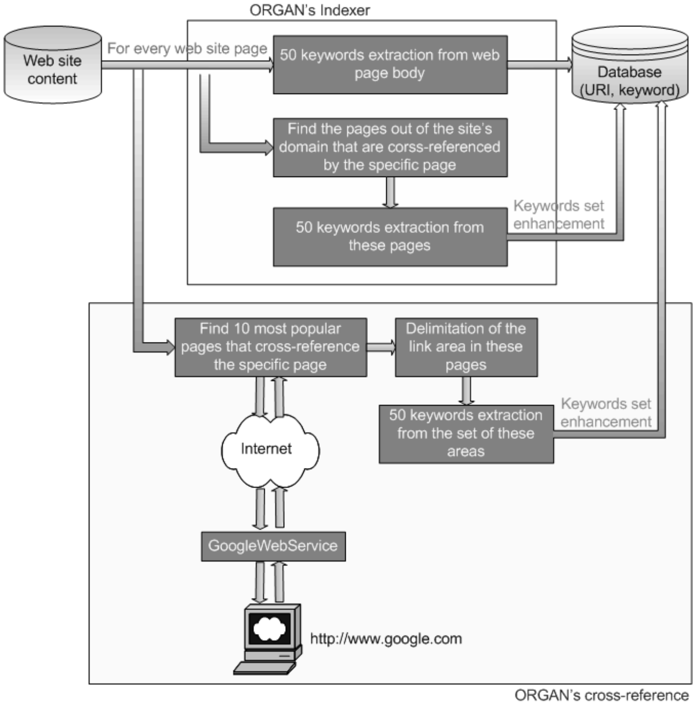

ORGAN's Indexer
Web site content
For every web site page
50 keywords extraction from web page body
Database (URI, keyword)
Find the pages out of the site's domain that are corss-referenced by the specific page
Keywords set enhancement
50 keywords extraction from these pages
Find 10 most popular pages that cross-reference the specific page
Delimitation of the link area in these pages
Keywords set enhancement
50 keywords extraction from the set of these areas
Internet
GoogleWebService
http://www.google.com
ORGAN's cross-reference

Figure 3 Keywords extraction modules architecture

Experimentation and fine tuning with ORGAN using the different Web site datasets resulted in the choice of fifty keywords for each set extracted. Further fine tuning can be performed as ORGAN can be reconfigured at any keyword list length.

The steps of the keywords extraction process are presented in Figure 3. Two distinct modules carry these tasks: the ORGAN Indexer and the ORGAN cross-reference.

4.1.1. ORGAN Indexer

In this module, the Web site is parsed and for every Web page the HTML tags are cleaned, the stop-words (very common words, numbers, symbols, articles) are removed, since they are considered not to contribute to the semantic denotation of the Web page's content, and at last, the top 50 most frequent keywords are extracted. The set of 50 keywords that represent a Web page is complemented by 50 more keywords which are extracted from the Web pages that are cross-referenced by the specific page.

The basis for the implementation of this module is the Web Indexer application [14], which traverses the Web starting from an initial URL, until each possible path has been exhausted. Its goal is to build a database of words cross-referenced by the Web pages containing those words.

4.1.2. ORGAN cross-reference

Next, the second module of the keywords extraction process is performed in order to find Web page references using Web searching. In the case of Web reference searching, Google [11] was utilized mainly because of its Web service programming interface. The first maximum 10 most popular Web sites, which reference the specific page, are considered. The HTML code of each of these pages is parsed and the area around the specific link is spotted. The margins of this area are anchored 100 bytes before the link and 100 bytes after the link. The text included in that character/byte window is accounted as representative of the referenced Web page topics, according to the results of paper [12], which was successfully verified in our case too. After removing HTML tags (and broken HTML tags from the edges) and the stop-words, the most frequent words are extracted. A set of the top 50 most frequent keywords is again kept for each Web site page from the set of keywords extracted from the set of 10 Web pages. Overall, after the compilation of these first two modules, appropriate keyword sets annotate every Web page of the Web site under analysis.

4.2. ORGAN Translation

To facilitate lexical processing in non-English written sites, Web page keywords should be translated. We have used the functionality of the WordNet [19] lexical database, for words of the English vocabulary only. To achieve automation in translation we built up a Web service, which posts the keywords to the Babel Fish translation engine (http://babelfish.altavista.com) and receives the translation results. In this way, the assignment process is performed. Other WordNet like solutions can be also easily incorporated such as the EuroWordNet (http://www.illc.uva.nl/EuroWordNet/) in order to achieve lexical analysis in other languages natively using a Web service interface.

5 Site Content Semantic Characterization

In this step, the content of the Web site pages is assigned to terms of an OWL ontology.

5.2. ORGAN Metadata-Assigner Module

The ORGAN Metadata Assigner module uses as a criterion the semantic similarity measure between every keyword and every term of the ontology to classify the URIs to classes and instances of the ontology. Each Web page is not represented by the three sets of keywords (within page keywords, referencing pages keywords and linked pages keywords) anymore, but by a very slighter terms set, which describe the Web page content.

The calculation of the semantic similarity measure between each keyword and each ontology term was accomplished using semantic similarity measures in combination with WordNet [19]. In WordNet, English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. The specific measure that was applied in this system is the Wu & Palmer one [31]. The specific measure calculates relatedness by considering the depths of the two synsets (one or more set of synonyms) in the WordNet taxonomies, along with the depth of the LCS.

$$score = \frac{2 * depth(lcs)}{(depth(s_1) + depth(s_2))}$$

This means: $score \in (0,1]$.

The score can never be zero because the depth of the LCS is never zero (the depth of the root of a taxonomy is one). The score is one if the two input synsets are the exactly same.

The successful assignment of Web pages to ontology terms is especially important for the reliability of the results that are retrieved from the queries on the ORGAN Analyzer module. For this purpose, the hierarchical organization of the ontology and the specific context of ontology's instances were taken under consideration. As a result, two distinct processes were implemented for the assignment of keywords to ontology classes and instances, correspondingly.

As regards the ontology classes, it is primarily essential to determine their meaning, before assigning them to URIs. The classes incorporate a representative ID, which is either a word recognizable by WordNet, or a string semantically comprehensible by the human, but not by a lexicon. For example, the second case involves classes, such as "AdministrativeStaff", "AssociateProfessor". In this case, the existence of the attribute <rdfs:label> is checked, which constitutes a description of the meaning of the specific class. In the case that the class "AdministrativeStaff" carries as a label the set of words "administrative staff worker", these are the words that will be semantically compared with Web site keywords instead of the class itself. Finally, the greatest score for each key-word is respectively used for the calculation of the similarity measure between the class and the keywords. If the class doesn't have a "label" attribute, it is examined whether the attribute <rdfs:eqivalentClass> exists. If it is found, the equivalent class is acquired for its ID/label recognition. In case, the class has not been recognized at all yet, the previous procedure is followed for its parent class considering it as the closest semantically relative one. The process will continue for the classes that are located on the path to the root-class, until a class is found that incorporates an ID or a «label» recognized by WordNet.

After class recognition phase, the keyword assignment process to classes starts. The similarity measure is calculated between every meaning of a keyword and a class, under the constraint that only identical word parts of speech are compared. For instance, there is no possible calculation of the semantic similarity measure between a verb and an adverb, or a verb and adjective, and so on. The greatest similarity measure of all the class meanings with all the meanings of a specific keyword is kept. This measure is numerically compared with the semantic measures of the rest of the classes for the certain keyword. Finally, the keyword is assigned to the class with the highest score.

Class instances are handled using a slightly different approach. The assignment of Web pages to an instance presumes that specific content related with the instance has been located to the page's keywords. As the instances may consist of one word or a phrase, they are tokenized for further examination. Every token is compared with all the page keywords in order to locate the word that is semantically most similar. In the case of phrases, tokens do not include stop words (i.e. common vocabulary words, which would never be used as a single-word query in a search engine, such as "to", "of", "the").

During our experiments, we found that the similarity measure of the closest keyword to the instance token should satisfy at least the threshold of 0.8, to have the word included in the keywords set that a Web page must contain in order to be accounted as relevant to the specific instance. The new terms set derives after the examination of every word-token of the instance and its assignment to a keyword, whenever the threshold is satisfied.

To handle exception cases, the instance tokens, which are not recognized in WordNet at all, are specially marked by the system with high importance. Such tokens are mainly entities, which can be found among the Web pages keywords. In this event, all those Web pages, which are characterized by the certain keywords set, will be assigned to the specific instance.

The assignment process is time-expensive, therefore we have implemented a caching policy to improve system response. The assignments of instances words are kept in cache, to minimize response time in case these words are met again.

As a result, the Web site pages will have been assigned to relevant classes and instances of the ontology after the completion of the metadata-assigner module. All assignments will be recorded in the ORGAN local database. In the following section, we present how our tool deals with the necessary log files pre-processing procedure in order to clean the Web usage trails of unnecessary non-textual information such as http errors or page icons.

6 Log Files Processing

A log file is a record of Web activity that automatically keeps track of client activity on a Web site server. It logs information such as the date, time, IP address[a], HTTP status, bytes sent, and bytes received [20]. The format of the log files is predefined for every Web application server and there is a number of different ones depending on the server technology. Some common log fields are:

```
Remotehost   rfc931   authuser   [date]   "request"   status   bytes
```

where *remotehost* is the remote hostname or the IP address if the DNS hostname is not available, *rfc931* is the remote login name of the user, *authuser* is username as which the user has authenticated himself (only in case of using password protected WWW page), *[date]* is the date and time of the request, *"request"* is the request line exactly as it came from the client (i.e., the file name, and the method used to retrieve it [typically GET]) and *status* is the HTTP response code returned to the client which indicates whether or not the file was successfully retrieved, and if not, what error message was returned and bytes is the number of bytes transferred. A log file example line with common log format looks like:

[a] Time, date and IP address are kept for both the requesting client and the responding web site server.

```
150.140.142.17 - - [01/Jan/2006:13:06:51 -0600] "GET /home.htm HTTP/1.0" 301 -4
```

A variant of the common log format has been lately proposed by W3C, which is called the “extended” log file format [9]. This format permits a wider range of data to be captured. This proposal was motivated by the need to capture a wider range of data for demographic analysis and also the needs of proxy caches. Some new fields are added, the most important of which are: the “referrer”, which is the url the client was on before requesting the current url, the "user agent" which is the software the client claims to be using and the “cookie”, in the case where the site visited uses cookies.

A log file example line with extended log format looks like:

```
150.140.142.17 - - [17/Jan/2005:09:56:31 +0200] "GET /index.htm HTTP/1.1" 200 669
"http://www.ceid.upatras.gr/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"
```

Overall, the main tasks undertaken during the log file processing are presented in the following figure.

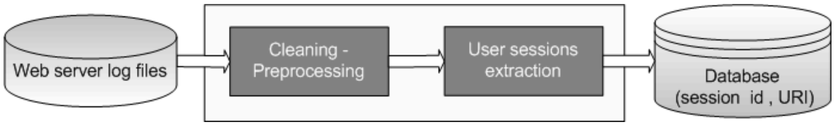


Figure 4 Log files processing

The preprocessing of the logs files follows two steps, as Cooley identifies them [6]: data cleaning, sessions’ identification and path completion. In our case, the log files are initially cleaned from records of requests for images, requests for non informational files, such as D-HTML parts (i.e. cascade stylesheets (css) & scripting code) and requests that were not successfully responded or were submitted by search spiders & robots.

In the following, the log files are parsed and user sessions are extracted. Sessions include the set of distinct pages that were visited. For the sessions determination, three aspects are taken into consideration: the user IP, the user agent and the time interval between subsequent requests. So, the couple IP-agent defines a user, however if time between hits exceeds half an hour, it is considered that a new session started, following the approach for the calculation of mean time between each user interface event that Catledge et al. [3] introduced. Next, path completion is performed and the page references that are missed due to local browsing caching mechanisms are filled in. The information extracted from the log files about the user sessions is stored in ORGAN local database.

7 ORGAN Analyzer

The ORGAN Analyzer is a knowledge analysis tool. It analyses the knowledge derived from three resources:

1. Web server raw log files
2. Database which contains metadata information of the site
3. OWL ontology with classes, instances and properties that correlate them

Particularly, this ORGAN module acts as a query builder that mines logs in relation to the Web site semantics. It uses the OWL ontology in order to reason about the Web site content in a machine-processable and domain-independent way. Next, it combines the pages popularity among users' visits as calculated from their sessions with the pages latent semantics, in order to provide deduction about the users' preferences and conceptually interpret detected interconnected visits. The primary parts of the query mechanism are presented in the Figure 5.

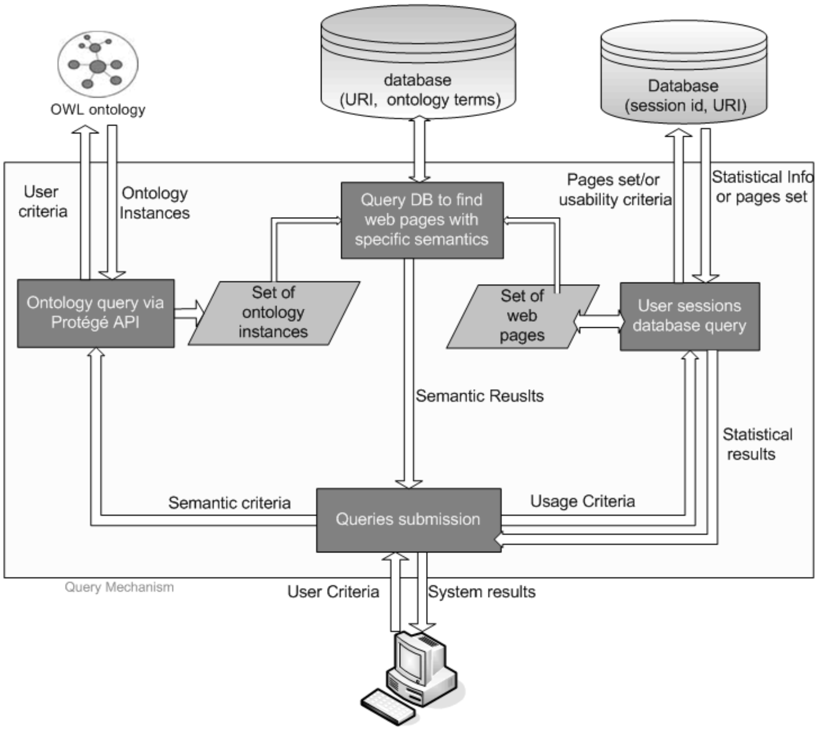

OWL ontology
database (URI, ontology terms)
Database (session id, URI)
User criteria
Ontology Instances
Query DB to find web pages with specific semantics
Pages set/or usability criteria
Statistical Info or pages set
Ontology query via Protégé API
Set of ontology instances
Set of web pages
User sessions database query
Semantic Reuslts
Statistical results
Semantic criteria
Queries submission
Usage Criteria
Query Mechanism
User Criteria
System results

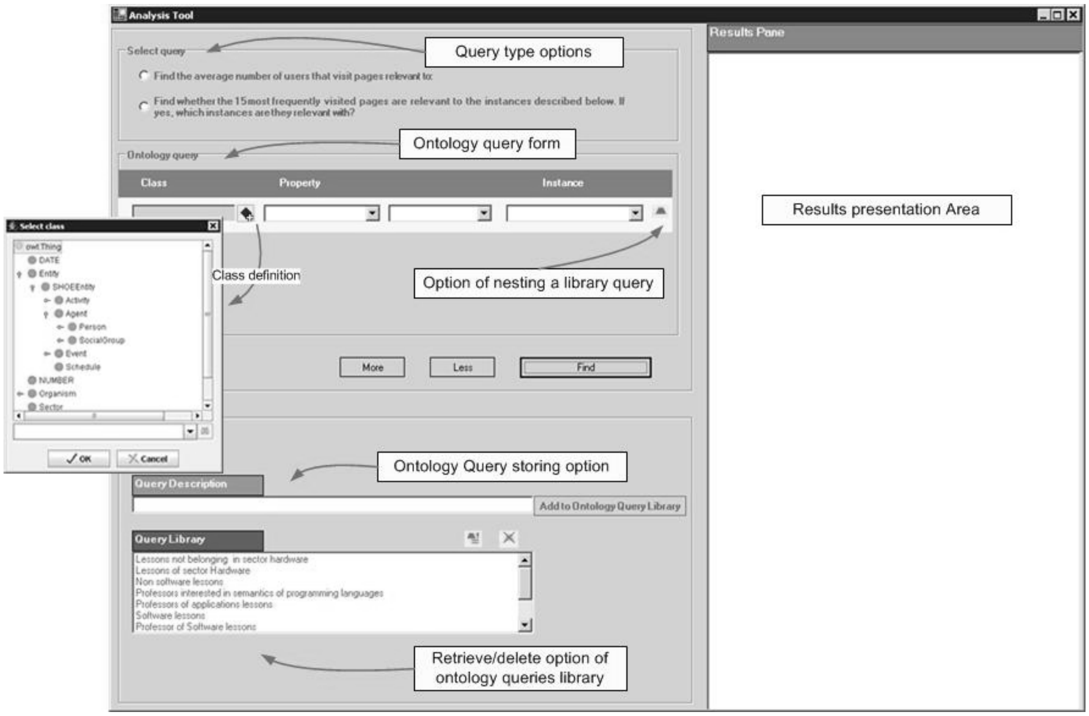
Figure 5 Query mechanism architecture

To facilitate knowledge acquisition functionalities, we have utilized the application programming interface of a very popular ontology management tool, the Protégé [24]. Without lost of generality any other ontology management tool could be utilized, however several reasons presented in the following subsections drove our decision to build upon it.

7.1. Ontology Management Integration

Protégé [24] is a tool which allows users to construct domain ontologies, customize data entry forms, and enter instance data. It is a free, open source ontology editor and knowledge-base framework. It provides a foundation for customized knowledge-based applications. Protégé supports XML Schema, RDF(S) and OWL. The Protégé API makes it possible for other applications to use, access, and display knowledge bases created with Protégé.

In our case, we integrated ontology management and querying facilities into our ORGAN tool in order to allow the analyst to specify semantics on the Web site that interest him/her. Next, details on the ORGAN Analyzer interface are presented.



○ Find the average number of users that visit pages relevant to:

○ Find whether the 15most frequently visited pages are relevant to the instances described below. If yes, which instances are they relevant with?

Lessons not belonging in sector hardware
Lessons of sector Hardware
Non software lessons
Professors interested in semantics of programming languages
Professors of applications lessons
Software lessons
Professor of Software lessons

Figure 6 ORGAN Analyzer Interface

7.2. ORGAN Analyzer Interface

The ORGAN Analyzer Interface is outlined in figure 6 and it provides access to a number of functions such as:

1. Form a semantic query (see figure 7):
   a. Select the statistical part queries. We call this static query part, because most Web log analyzers provide a list of predefined statistical questions. However, any possible query can be built and customized to mine in a statistical sense.
   b. Define the semantic criteria that have to be taken into consideration. We call this dynamic query part to underline the fact that any semantic notion can be chosen any time.
2. Store a query with a representative title in order to retrieve it more quickly later.
3. Recall a stored query as a condition during the building of a new query, achieving in this way the creation of a chain of nested queries.
4. Create union queries

The queries formed through our system are of the following format:

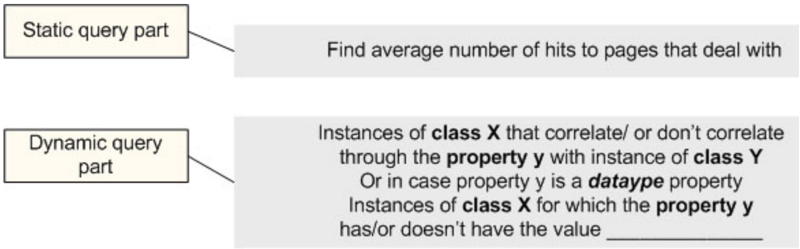

Static query part
Find average number of hits to pages that deal with
Dynamic query part
Instances of **class X** that correlate/ or don't correlate through the **property y** with instance of **class Y**
Or in case property y is a ***dataype*** property
Instances of **class X** for which the **property y** has/or doesn't have the value _______________

Figure 7 Query format

In particular, to submit a query, one has to select the statistical query type that needs. More possible query types are presented below and ORGAN supports any statistical query that can be transformed finally into a typical database SQL statement as all typical log analysis tools.

1. Find the average numbers of users that visit pages relevant to certain instances
2. Find whether the 15 most frequently visited pages are relevant to certain instances
3. Find the 15 most popular site subjects
4. Find the average number of user sessions that visit every one of the 10 most popular general thematic fields of the Web site

Henceforth, ORGAN delivers its new functionality compared to previous log analyzers. After selecting the query type, the instances of the pages that this query involves, can be defined in order to semantically enhance the query. The query part which is executed on the ontology may either be formed through the available form or it may be selected from a library storing favorite queries.

At this point, the capability of forming union queries has been incorporated. As a result, the user may define one or two groups of pages as criteria for the individual users' visits hits. This feature leads to the extraction of useful conclusions for the visitors' preferences. Via queries, such as "What users ratio visits Web pages relevant to hardware courses and pages relevant to software courses, as well?", the user may detect the pages that don't interrelate ostensibly, but interest some users groups.

Two kind of statistic results are calculated for the usage of pages that are semantically determined: the Overall Score(OS) and the Relative Score (RS). The *overall score* is the proportion of the distinct hits to the page in user sessions to the total number of user sessions. Let $U = \{U_1, U_2, ..., U_n\}$ be the set of users' sessions and P_i a Web page. We denote as a the number of distinct appearances of P_i in$U = \{U_1, U_2, ..., U_n\}$. The overall score of the page is expressed through the following relation.

$$OS = \frac{a}{n}\text{, such as } a = \left|\{P_i | P_i \in U\}\right|$$

The *relative score* of a Web page concerns only the case where union queries are performed. Let $B = \{B_1, B_2, ..., B_\ell\}$ and $C = \{C_1, C_2, ..., C_k\}$ be two semantic groups of pages. Let $P_i \in C$. Let

$u = \{u_1, u_2, ...u_m\}$ be a set of users' sessions where each u_i includes at least one page of the semantic group B. The relative score is the average proportion of users that visit at least one page of group B and access the examined page of group C in the same session. We denote with a_r the number of distinct appearances of P_i in u. The relative score of the Web page is expressed through the following relation.

$$RS = \frac{a_r}{m}$$

ORGAN displays these results in the right hand side of the interface pane. Intermediate results of the involved ORGAN modules can also be displayed for detailed analysis or debugging after user choice.

8 Implementation and Technological Details

ORGAN is a standalone application which was implemented in C#[25] in the Microsoft Visual Studio .NET environment [18] with a CPAN module, which applies semantic similarity measures on the WordNet lexical database.

Web Services [30] play an important role in the system. The Translation module is implemented though a Web service, using an online translation engine. Furthermore, the semantic similarity measures may be applied through a Web service, in case the CPAN module is not locally available or updated information on WordNet lexicon may be seek through the Web service. They were implemented using .NET and facilitate Web based loosely couple communication. In this way, ORGAN is expandable and supports scalability in terms of language lexical analysis and referencing pages discovery.

9 Case Study

In order to prove our tool effectiveness, we have applied it to two Web sites, which concern the Web site of the Computer Science and Informatics Department of the University of Patras (C.E.I.D site hereafter-http://www.ceid.upatras.gr) and a books e-shop: a pilot Web site (http://thaleia.westgate.gr/lettres_et_mots/ categories/categories.htm).

The C.E.I.D site is hosted in a Sunfire 280r with a single ULTRA SPARC processor at 900MHz, using OS Solaris 8 and Web Server Apache 1.3.26. We obtained a web log covering 6 months (January 2005 – May 2005) from which a set of 25938 sessions (users visits) for 4212 unique visitors though the LogPro module.

This site constitutes the official Web site of the Computer Science Department in Patras, has 4 levels and publishes a number of 253 web pages. Its content provides information about department administration, undergraduate and postgraduate studies, staff, research projects, news and interests, and the university campus. It was designed with HTML, apart from some dynamic pages implemented with PHP. There are two available versions (the Greek and the English one). The Web site's structure graph may be considered coherent. Its content was preprocessed by the ORGAN Indexer, the Cross-reference, the Translation and the Metadata Assigner modules. The ontology that was used was initially based on a DAML ontology [23] that describes a computer science university department. We transmuted this ontology to OWL format and instantiated it in Protégé.

In order to provide extra validation for our system functionality, we applied it to the pilot e-shop Web Site, whose logs are in Web Server IIS 6.0 format. We obtained a Web log covering 3 months (March 2005 – May 2005) from which a set of 193 sessions (users visits) for 64 unique visitors though the LogPro module.

The site consists of 100 unique pageviews and its maximum navigation depth is 5. Its content provides information about the world literary and its language is the Greek one. It was designed with PHP and MySQL. Its content was preprocessed by the 4 aforementioned modules. The ontology that was used in this case was an OWL ontology (http://ebiquity.umbc.edu/ontology/publication.owl) that describes publications.

After having completed the input data processing, the ORGAN Analyzer module was used in order to retrieve answers to queries that combine Web pages usage with their semantic features, which concern ontology terms and how they correlate with each other.

A large scale of experimentation was undertaken for each one of the examined case studies and some interesting results were obtained about the users' preferences and the possible re-arrangements actions in these web sites for facilitating the user's task. In the following, we present representative results of our semantic log analysis.

Example1

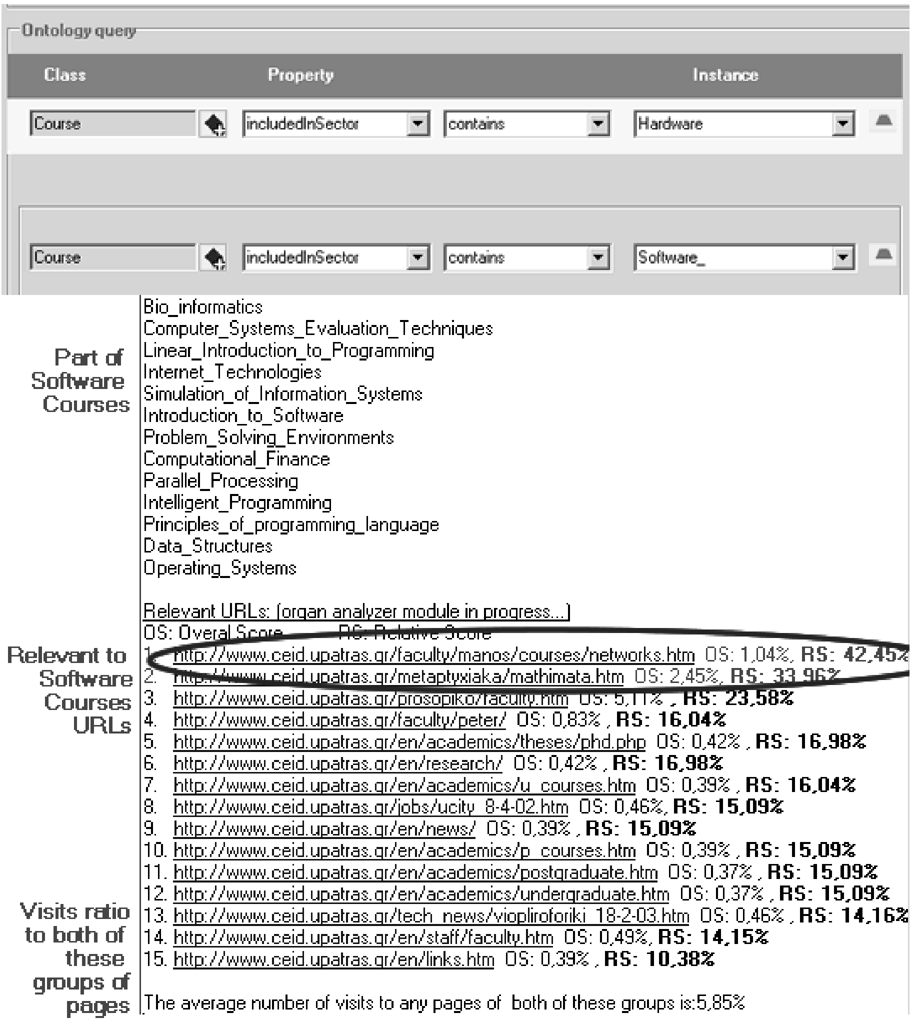
In the C.E.I.D case, we aimed to find ratio of users that visit pages relevant to hardware courses and pages relevant to software courses during a single visit session. As, it is shown in Figure 8, the specific groups of courses are retrieved from the ontology and returned to the results set. Furthermore, the web pages that have been assigned to each group of web pages and the average usage ratio of these groups of pages are returned, as well.

The queries that concern the ontology are expressed through the relations "Course :: IncludedInSector :: Hardware" and "Course :: IncludedInSector :: Software". A number of 13 web site pages were found to be relevant to the first instances and a number of 15 web pages were found to be relevant to the second instances. In the results pane, a part of instances of software courses, the relevant URLs sorted by their relative scores, with their overall and relative score and the average ratio of hits to both of these groups of pages in the same individual user sessions (5.85%) are displayed. The fact that the two first URLs have the greatest relative scores means that the users that visit both hardware courses and software course related pages present an increased preference to those two pages. This is a logic conclusion, because the first URL concerns the network courses, which includes networking programming matters and the second URL concerns the postgraduate courses. The curriculum of our department postgraduate studies requests from students to attend both hardware and software courses. The site structure could be re-arranged in order to access more easily these two links from pages relevant with hardware courses.

Example 2

In the e-shop case, we aimed to find ration of users that visit pages relevant to crime books and pages relevant to comics within a single visit session. The queries that concern the ontology are expressed through the relations "Book::belongsToCategory::Crime" and "Book :: belongsToCategory:: Comics".

As it is presented in the figure 9, these groups of books are popular for the site visitors. Moreover, it is noticed that there is a group of users that is highly interested in both of these groups. So, it is very likely that people who will visit pages about crime books will also visit pages about comics' books. This remark would be very useful for a site designer who would aim at improving the Web site's navigational options. The direct linking of these two book categories in the site structure would facilitate the user while searching for books of his interest.


Ontology query
Class
Property
Instance
Part of Software Courses
Relevant to Software Courses URLs
Visits ratio to both of these groups of pages

Course | includedInSector | contains | Hardware

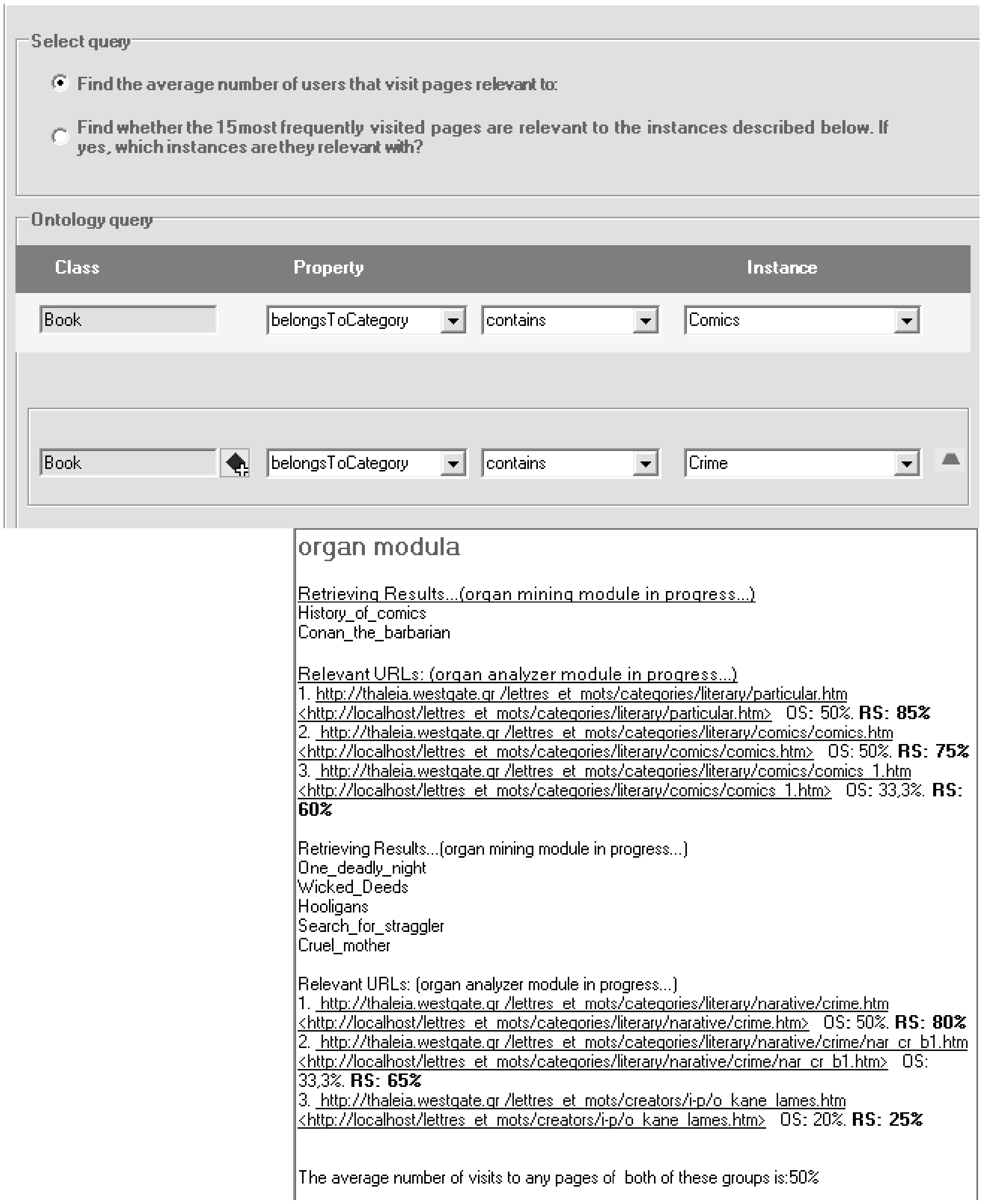
Course | includedInSector | contains | Software_

```
Bio_informatics
Computer_Systems_Evaluation_Techniques
Linear_Introduction_to_Programming
Internet_Technologies
Simulation_of_Information_Systems
Introduction_to_Software
Problem_Solving_Environments
Computational_Finance
Parallel_Processing
Intelligent_Programming
Principles_of_programming_language
Data_Structures
Operating_Systems

Relevant URLs: (organ analyzer module in progress...)
OS: Overal Score          RS: Relative Score
1.  http://www.ceid.upatras.gr/faculty/manos/courses/networks.htm  OS: 1,04%, RS: 42,45%
2.  http://www.ceid.upatras.gr/metaptyxiaka/mathimata.htm  OS: 2,45%, RS: 33,96%
3.  http://www.ceid.upatras.gr/prosopiko/faculty.htm  OS: 5,11% , RS: 23,58%
4.  http://www.ceid.upatras.gr/faculty/peter/  OS: 0,83% , RS: 16,04%
5.  http://www.ceid.upatras.gr/en/academics/theses/phd.php  OS: 0,42% , RS: 16,98%
6.  http://www.ceid.upatras.gr/en/research/  OS: 0,42% , RS: 16,98%
7.  http://www.ceid.upatras.gr/en/academics/u_courses.htm  OS: 0,39% , RS: 16,04%
8.  http://www.ceid.upatras.gr/jobs/ucity_8-4-02.htm  OS: 0,46%, RS: 15,09%
9.  http://www.ceid.upatras.gr/en/news/  OS: 0,39% , RS: 15,09%
10. http://www.ceid.upatras.gr/en/academics/p_courses.htm  OS: 0,39% , RS: 15,09%
11. http://www.ceid.upatras.gr/en/academics/postgraduate.htm  OS: 0,37% , RS: 15,09%
12. http://www.ceid.upatras.gr/en/academics/undergraduate.htm  OS: 0,37% , RS: 15,09%
13. http://www.ceid.upatras.gr/tech_news/vioplirofoRiki_18-2-03.htm  OS: 0,46% , RS: 14,16%
14. http://www.ceid.upatras.gr/en/staff/faculty.htm  OS: 0,49%, RS: 14,15%
15. http://www.ceid.upatras.gr/en/links.htm  OS: 0,39% , RS: 10,38%

The average number of visits to any pages of  both of these groups is:5,85%
```

Figure 8 Union query about the C.E.I.D web site usage. Two semantic groups of pages are examined (hardware and software courses). There is not any remarkable interconnection visit ratio (5,85% visited both groups of pages). However, we notice that the software-related pages of postgraduate and network courses are of high interest by users visiting pages relevant with hardware courses (42,45% and 33,96% relative scores, correspondingly).



Select query

● Find the average number of users that visit pages relevant to:

○ Find whether the 15most frequently visited pages are relevant to the instances described below. If yes, which instances are they relevant with?

Ontology query

Class	Property		Instance
Book	belongsToCategory	contains	Comics
Book	belongsToCategory	contains	Crime

```
organ modula

Retrieving Results...(organ mining module in progress...)
History_of_comics
Conan_the_barbarian

Relevant URLs: (organ analyzer module in progress...)
1. http://thaleia.westgate.gr /lettres_et_mots/categories/literary/particular.htm
<http://localhost/lettres_et_mots/categories/literary/particular.htm>  OS: 50%. RS: 85%
2.  http://thaleia.westgate.gr /lettres_et_mots/categories/literary/comics/comics.htm
<http://localhost/lettres_et_mots/categories/literary/comics/comics.htm>  OS: 50%. RS: 75%
3.  http://thaleia.westgate.gr /lettres_et_mots/categories/literary/comics/comics_1.htm
<http://localhost/lettres_et_mots/categories/literary/comics/comics_1.htm>  OS: 33,3%. RS:
60%

Retrieving Results...(organ mining module in progress...)
One_deadly_night
Wicked_Deeds
Hooligans
Search_for_straggler
Cruel_mother

Relevant URLs: (organ analyzer module in progress...)
1.  http://thaleia.westgate.gr /lettres_et_mots/categories/literary/narative/crime.htm
<http://localhost/lettres_et_mots/categories/literary/narative/crime.htm>  OS: 50%. RS: 80%
2.  http://thaleia.westgate.gr /lettres_et_mots/categories/literary/narative/crime/nar_cr_b1.htm
<http://localhost/lettres_et_mots/categories/literary/narative/crime/nar_cr_b1.htm>  OS:
33,3%. RS: 65%
3.  http://thaleia.westgate.gr /lettres_et_mots/creators/i-p/o_kane_lames.htm
<http://localhost/lettres_et_mots/creators/i-p/o_kane_lames.htm>  OS: 20%. RS: 25%


The average number of visits to any pages of  both of these groups is:50%
```

Figure 9 Union query about the books e-shop. Two semantic groups of pages (related to comics and crime books) are examined regarding their popularity and interconnection visit ratio. These groups of pages are extremely popular (high overall scores OS). Moreover, they prove to be co-visited very often, since relative scores are noticeably high 60-85%). Interconnection links are proposed for re-organization issues.

10 Conclusions and Future Steps

We introduced the ORGAN Web log analysis tool that offers an integrated solution of building and performing analysis, taking into consideration both the site content semantics and the Web site page visits. Information about the user preferences in relation to the Web site topics may be extracted, which will constitute ORGAN as a valuable application during the site administrator's decision-making about the Web site reorganization.

ORGAN only needs as input: the URL of the Web site to be examined, the Web site logs for the period of interest, and a corresponding OWL ontology, relevant to the Web site's domain knowledge, which will have been instantiated through an ontology editor. Considering the variety and simplicity of the available free ontology editors, this is a task easy to be performed by a non ontologies-familiar user. ORGAN integrates the facilities of the popular Protégé and it is able to process all this data using its modules in order to construct an appropriate knowledge base. This base will be utilized by the ORGAN Analyzer module to answer semantic queries about the site usage.

Concluding, ORGAN is an integrated tool, taking advantage of extra online services through a service-oriented architecture. WordNet-based similarity measurement, term translation, OWL ontology querying constitute a set of individual services that our system unifies, to achieve its final goal; the Web site usage log semantic analysis. ORGAN aims to be a useful application ready to serve any Web site administrator, Web log miner and site analyst towards a semantically enhanced interpretation of their logs. We also hope that ORGAN will boost researchers' efforts towards semantic Web site usage analysis.

Future steps include works for a number of different directions. First of all, a crucial point for the system reliability comprises the phase of keywords extraction and their assignment to ontology terms. We dealt with cases, where substantial words had been neglected and ORGAN was set in debugging. This aspect could be enhanced using further study of linguistic rules and patterns possible to be applied. Such patterns should detect these words that are usually nominal entities, so as to assign them a special weight. To this direction, the research may include the (semi-)automatic instance extraction from the Web pages themselves.

Furthermore future work includes, the log files processing enhancement research. We consider the impact of taking into account structural information such as the Web pages depth, during the discovery of the most popular pages. The integration of different log analysis structure metrics [5] and session building methods could ameliorate the Web pages value calculation. Moreover tool expansions in order to support Web site reorganization tasks semi or fully automatically are particularly interesting. Particularly we see that the integration of the tool with a Web design suite could be a powerful solution for Web designers and site analysts to the refinement of Web site towards the improvement of visitors browsing experience. Nonetheless we hope that ORGAN will help researchers to the deeper understanding of user behavior over a particular Web application.

References

1. Berkhin, P., Becher, J.D. and Randall, D.J. Interactive path analysis of Web site traffic. Proceedings of KDD01, 2001, pp. 414-419.
2. Botafogo, R.A., Rivlin, E. and Shneiderman, B. Structural Analysis of Hypertext: Identifying Hierarchies and Useful Metrics. ACM Transactions on Information Systems, April 1992, vol. 10, no 2, 142-180.
3. Catledge, L. D. and Pitkow, J. E. "Characterizing browsing strategies in the World Wide Web. Computer Network and ISDN Systems, 1995, vol. 27, 1065-1073.

4. Chen, MS, Park, JS, Yu, JS, Seoul, K. Data mining for path traversal patterns in a Web environment. In Proceedings of the 16th International Conference on Distributed Computing Systems, 1996, 385-392.
5. Christopoulou, E., Garofalakis, J., Makris, C., Panagis, Y., Psaras-Chatzigeorgiou, A., Sakkopoulos, E., Tsakalidis, A. Techniques and Metrics For Improving Website Structure. Journal of Web Engineering, 2003, 2(1-2): 90-104.
6. Cooley, R. The use of Web structure and content to identify subjectively interesting Web usage patterns. ACM Transactions on Internet Technology, 2003, portal.acm.org.
7. Dai, H. and Mobasher, B. Integrating semantic knowledge with Web usage mining for personalization. Web Mining: Applications and Techniques, A. Scime (Ed.), Hershey: Idea GroupPublishing, 2004, 276-306.
8. Eirinaki, M., Vazirgiannis, M., Varlamis, I. SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process. In Proceedings of the 9th SIGKDD Conference, 2003.
9. Extended Log File Format Specification. http://www.w3.org/TR/WD-logfile.html.
10. Garofalakis, J., Kappos, P. & Mourloukos, D. Web Site Optimization Using Page Popularity. IEEE Internet Computing, ,1999, 3(4): 22-29.
11. Google Web Apis Home Page. http://www.google.com.gr/apis/.
12. Halkidi, M., Nguyen, B., Varlamis, I., Vazirgiannis, M. THESUS: Organizing Web Document Collections Based on Link Semantics, in VLDB Journal, special issue on Semantic Web, 2003.
13. Hong, J.I., Heer, J., Waterson, S., Landay, J.A. WebQuilt: A proxy-based approach to remote Web usability testing. ACM Transactions on Information Systems, 2001, Vol 19, no 3, 263-285.
14. Index the Web with .NET. http://www.vsj.co.uk/articles/display.asp?id=407.
15. Jansen, B. J. Search log analysis: What is it; what's been done; how to do it. Library and Information Science Research, 2006, 28(3), 407-432
16. Jin, X., Zhou, Y. and Mobasher, B. Web usage mining based on probabilistic latent semantic analysis. In Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2004, 197–205.
17. Kosala, R. and Blockeel, H. Web Mining Research: A Survey. ACM SIGKDD, July 2000.
18. Microsoft Visual Studio. http://msdn.microsoft.com/vstudio/.
19. Miller,GA. WordNet: A lexical database for English. Communications of the ACM, 1995, 38(11):39--41.
20. Pfizer Glossary. http://www.pfizer.com/pfizer/privacy/ mn_privacy_glossary.jsp.
21. Pitkow, J.E., Bharat, K.A. Webviz: A Tool For World-Wide Web Access Log Analysis. In Proceedings of 1st World Wide Web Conference (WWW1), Geneva, Switzerland, May 1994, 271–277. Elsevier Science BV, Amsterdam, 1994.
22. Srikant, R. and Yang, Y. Mining Web logs to improve Website organization. In Proceedings of the WWW10, Hong-Kong, May 2001, 430-437.
23. The DARPA Agent Markup Language Web Site. http://www.daml.org/ontologies/64.
24. The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/.
25. Visual C# Developer Center. http://msdn.microsoft.com/vcsharp/.
26. Web reference:Analog. http://www.analog.cx.
27. Web reference:SurfStats. http://www.surfstats.com.
28. Web reference:Web Trends. http://www.Webtrends.com.
29. Web reference:WebLogs. http://www.cape.com.
30. Web Services Activity. http://www.w3.org/2002/ws/.
31. Wu and Palmer, M. Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994, 133–138.