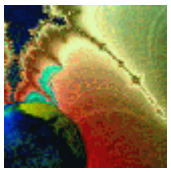
The University of Manchester

School of Informatics

HERACLITUS User Guide



Alexander Mikroyannidis and Babis Theodoulidis

{A.Mikroyannidis, B.Theodoulidis}@manchester.ac.uk

July 2005

Table of Contents

1 HERACLITUS Overview

HERACLITUS implements a framework for Semantic Web Adaptation. The Heraclitus framework[1, 2, 3] proposes the adaptation of the Semantic Web, based on web usage data. This approach aims to the adaptation of the web in order to assist the users in their browsing tasks. Web usage mining as well as text mining methodologies are employed. Both the physical and semantic structure of the web are targeted. The web site ontology is semi-automatically built and evolves through the adaptation procedure.

HERACLITUS is a suite of tools for the adaptation of a web site. Figure 1 shows the main screen of the application. The user can choose to perform the following actions:

- *Log Preprocessing*: The user can import the access log files into a database and perform data cleaning tasks on them. The output of this process is sessions of user visits.

- *Categorization*: The content categorization of web pages is prepared through this task.

- *Session Mining*: The sessions that have been derived during the access logs preprocessing are mined in order to produce *pagesets*. These are sets of pages that are frequently accessed together throughout the same session.

---

[1] Mikroyannidis, A. *Development of a framework for self-adaptive web sites*. School of Informatics, University of Manchester, MPhil Thesis, 2004.

[2] Mikroyannidis, A. and Theodoulidis, B. A Theoretical Framework and an Implementation Architecture for Self Adaptive Web Sites. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)* (Beijing, China, 2004), pages 558-561.

[3] Mikroyannidis, A. and Theodoulidis, B. Web Usage Driven Adaptation of the Semantic Web. In *Proceedings of the End User Aspects of the Semantic Web Workshop, 2nd European Semantic Web Conference (ESWC 2005)* (Heraklion, Greece, May 29-June 1, 2005), pages 137-147, http://kmi.open.ac.uk/events/usersweb/papers/12_mikroyannidis_final.pdf

- *Site Adaptation*: The pagesets and the content categorization of web pages are used to produce reports with proposed adaptations for the web site.

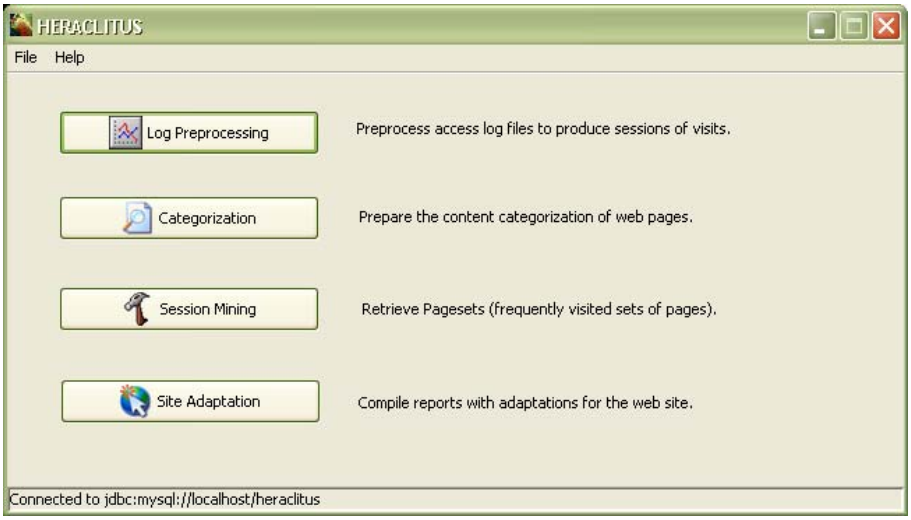


Figure 1. The HERACLITUS main screen

The functionality of the tools comprising the HERACLITUS platform is described in detail in the tutorial that follows. The usage of the tools is explained through a typical scenario of web adaptation.

2 Web Adaptation Tutorial

The sample data used in this tutorial are derived from the web site of the Informatics School of the University of Manchester. The Informatics web site (www.informatics.manchester.ac.uk) is comprised of approximately 2,500 web pages, including dynamic php pages, as well as supporting material for modules (pdf, zip files). The site contains information about the department's staff, facilities, research, postgraduate and undergraduate studies. It also supports various modules by providing frequent announcements and resources for them. Most of the traffic is generated by undergraduate or postgraduate students who access the supporting material of the modules.

2.1 Database connectivity

HERACLITUS interacts with a MySQL database, which holds all the data gathered from the access logs plus the information about the visiting sessions. MySQL[4] is currently the most popular open source SQL (Structured Query Language) database management system. It delivers a fast, multi-threaded, multi-user and robust SQL database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. HERACLITUS has been tested with versions 4.1 and 5.0 of MySQL.

When starting the application, we are prompted with a dialog box, where the database connection details can be entered. In particular, we can specify the name of the machine where MySQL is running (MySQL host), the name of the database, as well as the username and password used to login to the database. We are then informed whether the connection to the database was established and the result is also displayed in the status bar of the HERACLITUS main screen.

[4] http://www.mysql.com/

If we choose not to connect to a database, then the tasks that require this connection (Log Preprocessing and Session Mining) will become unavailable in the HERACLITUS main screen.

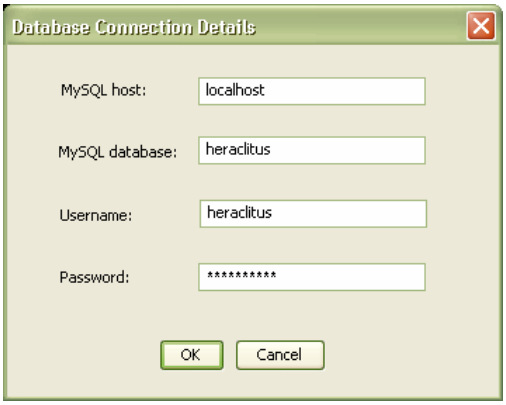


Figure 2. The database connection details dialog

2.2 Log Preprocessing

The first step in log preprocessing is importing the access log files to the database. This is done from **File ⇨ Import Log Files**. The files need to be in the Extended Log File Format. An example of this format is given in Table 1.

Table 1. Extract from an Extended Log File

remotehost	rfc931	authuser	[date]	"request"	status	bytes	"referer"	"user_agent"
123.45.67.89	-	jvb	[01/Jan/2004:12:57:45 -0600]	"GET /~test/ HTTP/1.0"	304	0	"http://www.msn.com/"	"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.2.1) Gecko/20030225"

The fields of this format are:

- *remotehost*: The remote hostname or IP address number if DNS hostname is not available or was not provided.
- *rfc931:* The remote login name of the user. If not available, a minus sign is typically placed in the field.
- *authuser:* The username as which the user has authenticated himself. This is available when using password protected WWW pages. If not available, a minus sign is typically placed in the field.

- *[date]:* The date and time of the request.

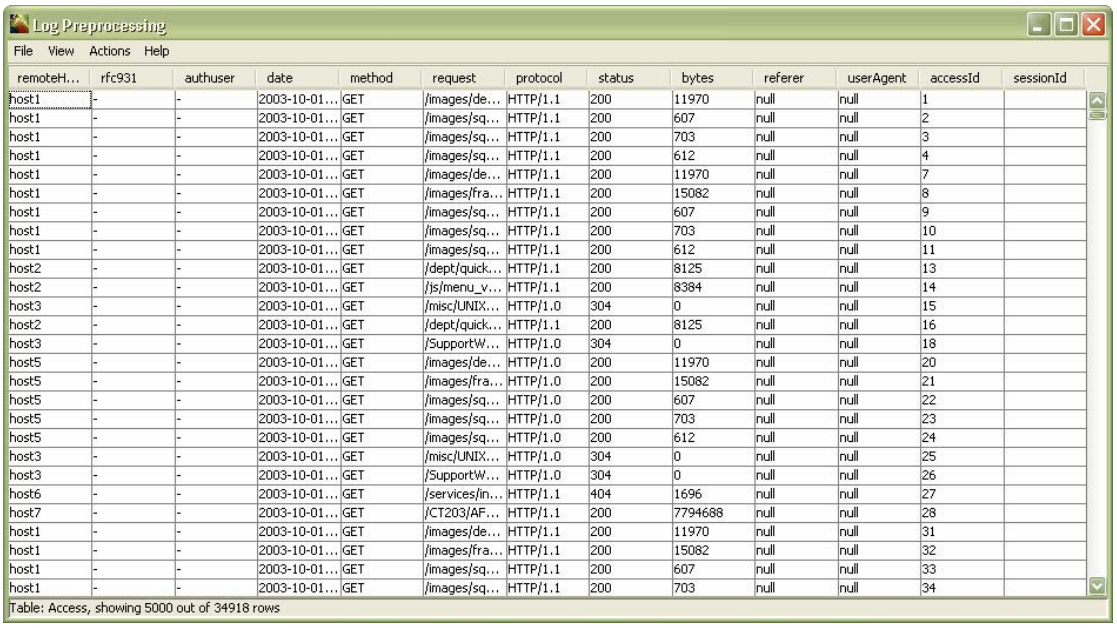
- *"request":* The request line exactly as it came from the client, i.e. the file name and the method used to retrieve it (e.g. GET, POST, HEAD).

- *status:* The HTTP response code returned to the client. Indicates whether or not the file was successfully retrieved, and if not, what error message was returned.

- *bytes:* The number of transferred bytes.

- *"referer":* The URL the client was on before making this request. If it cannot be determined, a minus sign will be placed in this field.

- *"user_agent":* The software the client claims to be using. If it cannot be determined, a minus sign will be placed in this field.



remoteH...	rfc931	authuser	date	method	request	protocol	status	bytes	referer	userAgent	accessId	sessionId
host1	-	-	2003-10-01...	GET	/images/de...	HTTP/1.1	200	11970	null	null	1	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	607	null	null	2	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	703	null	null	3	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	612	null	null	4	
host1	-	-	2003-10-01...	GET	/images/de...	HTTP/1.1	200	11970	null	null	7	
host1	-	-	2003-10-01...	GET	/images/fra...	HTTP/1.1	200	15082	null	null	8	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	607	null	null	9	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	703	null	null	10	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	612	null	null	11	
host2	-	-	2003-10-01...	GET	/dept/quick...	HTTP/1.1	200	8125	null	null	13	
host2	-	-	2003-10-01...	GET	/js/menu_v...	HTTP/1.1	200	8384	null	null	14	
host3	-	-	2003-10-01...	GET	/misc/UNIX...	HTTP/1.0	304	0	null	null	15	
host2	-	-	2003-10-01...	GET	/dept/quick...	HTTP/1.1	200	8125	null	null	16	
host3	-	-	2003-10-01...	GET	/SupportW...	HTTP/1.0	304	0	null	null	18	
host5	-	-	2003-10-01...	GET	/images/de...	HTTP/1.0	200	11970	null	null	20	
host5	-	-	2003-10-01...	GET	/images/fra...	HTTP/1.0	200	15082	null	null	21	
host5	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.0	200	607	null	null	22	
host5	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.0	200	703	null	null	23	
host5	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.0	200	612	null	null	24	
host3	-	-	2003-10-01...	GET	/misc/UNIX...	HTTP/1.0	304	0	null	null	25	
host3	-	-	2003-10-01...	GET	/SupportW...	HTTP/1.0	304	0	null	null	26	
host6	-	-	2003-10-01...	GET	/services/in...	HTTP/1.1	404	1696	null	null	27	
host7	-	-	2003-10-01...	GET	/CT203/AF...	HTTP/1.1	200	7794688	null	null	28	
host1	-	-	2003-10-01...	GET	/images/de...	HTTP/1.1	200	11970	null	null	31	
host1	-	-	2003-10-01...	GET	/images/fra...	HTTP/1.1	200	15082	null	null	32	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	607	null	null	33	
host1	-	-	2003-10-01...	GET	/images/sq...	HTTP/1.1	200	703	null	null	34	

Figure 3. Accesses imported into the database

During the import, the “remotehost” field is anonymized. Figure 3 shows how the database table **Access** looks like after the import of the sample log file **Anonymized_Sample.log**.

The imported accesses need to be cleaned before any sessions are identified. The cleaning of the accesses is a 3-step process. The following accesses are filtered out:

- *Accesses to multimedia content* (**Actions ⇨ Data Cleaning ⇨ Remove Multimedia Content Accesses**): These data are downloaded without an explicit request by the user, e.g. image and sound files. The list of file types that is used for this filtering is contained in the **MultimediaContent** table (**View ⇨ Browse Table ⇨ MultimediaContent**).

- *Robot accesses* (**Actions ⇨ Data Cleaning ⇨ Remove Robot Accesses**): Search engines use robot agents to crawl the web on a regular basis. Apparently, such accesses should be removed from the logs, as they do not constitute user behaviour. The list of robots that is used for this filtering is contained in the **Robot** table (**View ⇨ Browse Table ⇨ Robot**).

- *Erroneous accesses* (**Actions ⇨ Data Cleaning ⇨ Remove Erroneous Accesses**): Records with status codes that correspond to bad requests (e.g. missing pages) or unauthorized accesses must also be removed.

 *Tip*: There is no restriction on the order with which you can execute the data cleaning tasks.

We are now ready to identify the users' visiting sessions (**Actions ⇨ Session Identification**). Figure 4 shows the sessions derived from our sample data (table **Session**). We can see that each session has an id (**sessionId**), the number of accesses it consists of (**size**), as well as a starting and ending timestamp (**startDate**, **endDate**). We can view some statistics if we select **Actions ⇨ Session Statistics** (Figure 5).

 *Tip*: You can empty the **Access** and **Session** table by selecting **Actions ⇨ Remove All Accesses** and **Actions ⇨ Remove All Sessions.**

Tip: Session identification adds the newly discovered sessions to the old ones (if there are any stored in the **Session** table).

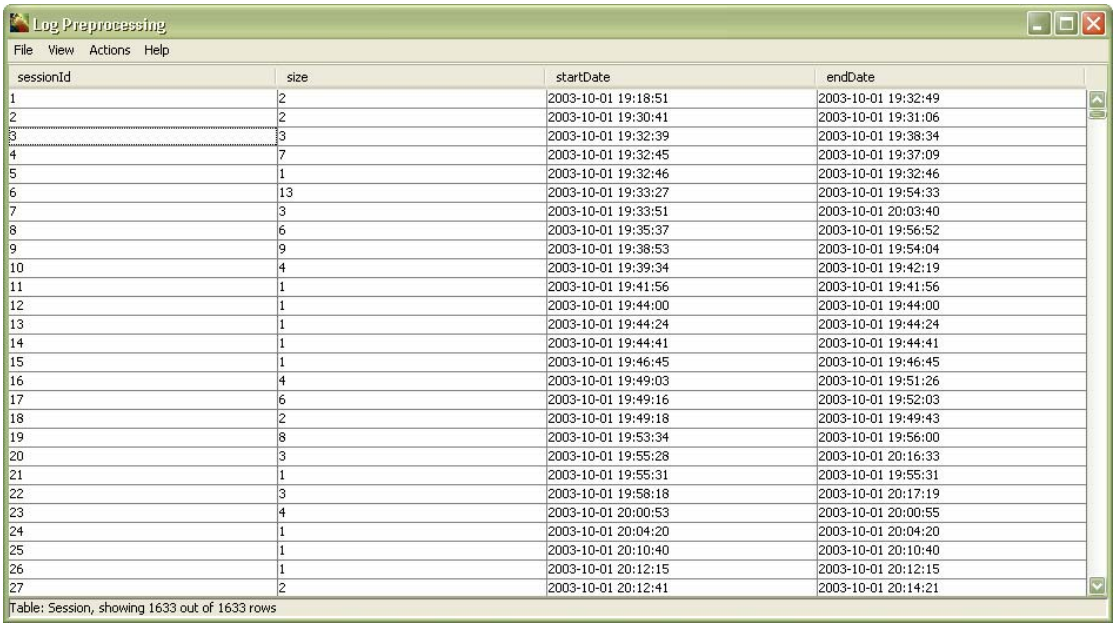


Figure 4. Visiting sessions

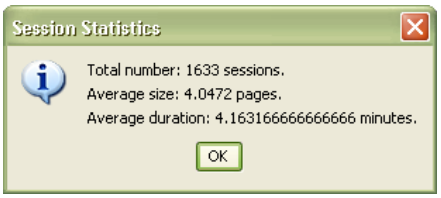


Figure 5. Session statistics

2.3 Categorization

The Categorization tool allows us to prepare the content categorization of the pages of a web site. This involves training the Support Vector Machines[5] categorization algorithm (SVM). Content categorization is used in the Session Mining and Site Adaptation tasks.

Figure 6 shows the Training panes, through which we can create the train set features and the categories' weight lists. The train set features are the most frequent terms used

[5] Cortes, C. & Vapnik, V. (1995) "Support Vector Networks", Machine Learning, vol. 20, no. 3, pages 273-297.

in the pages that comprise the training set. In order to calculate the train set features, we must choose the pages of the train set and enter the desired size of the feature list (default 250).

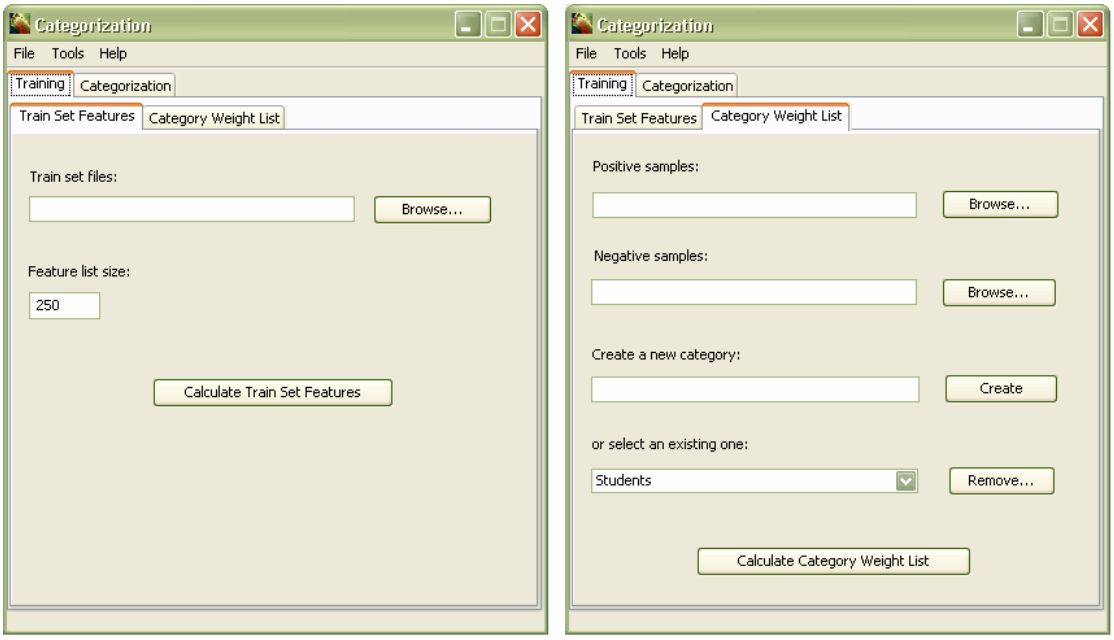


Figure 6. The Training panes

For the training to be complete, the weight list for each thematic category must be calculated. This list contains the features of each category together with their positive or negative weight. A feature has a positive weight if it belongs to the category; otherwise, a negative value is assigned to its weight. For the calculation of the weight list, a number of positive and negative samples are required. The positive samples are pages belonging to the category, whereas the negative samples are pages belonging to other categories. We can choose to calculate the weight list for an existing category or create a new one. We can also remove an existing category.

The tool has already been trained for the web site of our case study. The categories that we have defined are:

- *Students*: pages related to student issues, e.g. timetables.
- *Staff*: pages related to the staff of the School, e.g. CVs.
- *School*: pages related generally to the School, e.g. School news.
- *Research*: pages concerning the research done in the School.

We can test the algorithm by performing categorization on a selected page (Figure 7).

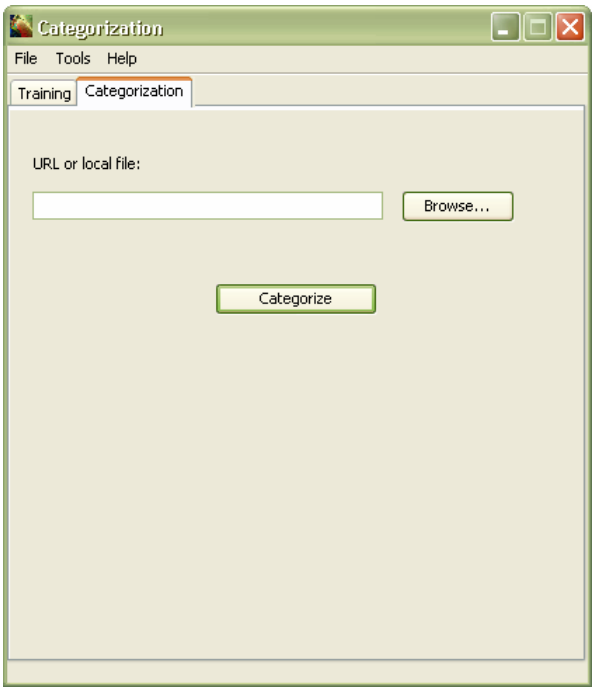


Figure 7. The Categorization pane

From **Tools ⇨ Options** we can change the tool's configuration (Figure 8). The stop word list contains terms that appear very often and that do not produce any distinctive meaning for any of the categories. These terms can be articles, pronouns, etc, such as "the", "an", and "he". Apart from these terms, and depending on the domain of the collection, words like "computer", or "dollar", etc. can be also removed.

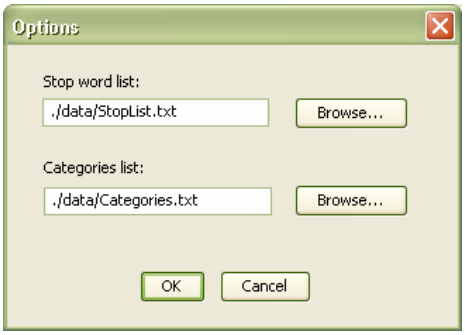


Figure 8. The Options dialog

2.4 Session Mining

In this phase, we will mine the sessions that were discovered during log preprocessing in order to produce pagesets (frequently visited sets of pages). Pagesets provide us

with an insight in the way the web site is browsed: what kind of pages are usually preferred by users and in what patterns these are visited.

There is a number of parameters we can tune, getting different results in session mining. Figure 9 shows the corresponding pane, where the following parameters are displayed:

- *Sessions input source*: This is the database where the sessions will be retrieved from.

- *Session Size Threshold (accesses/session)*: This is the minimum number of accesses that a session consists of. If we set this factor to 2, we will retrieve only the sessions that contain at least 2 accesses, that is the sessions of 1 access will be disregarded.

- *Support Threshold*: This is the minimum support value used by the mining algorithm. As this value is increased, the number of generated pagesets decreases and vice versa.

- *Web Site URL*: During the process of pagesets creation, the web site is contacted in order to retrieve metadata of the pages participating in the pagesets.

When starting the mining process, we are asked where we want the pagesets file to be stored. The output is written in XML format. After the process is complete, the produced pagesets are displayed in an Internet Explorer window (Figure 10).

As we can see, at the beginning of the XML document general information about the pagesets are given, such as the title of the web site, the start and end timestamp of the analyzed access logs, the support threshold, the session size threshold and the number of sessions. For each pageset, the title, category and URL of its pages are provided. The category is retrieved with the use of the SVM algorithm that we trained before.

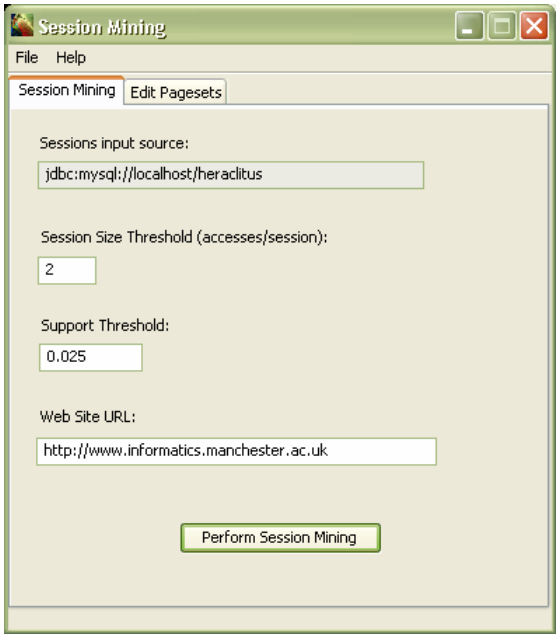


Figure 9. Session mining options

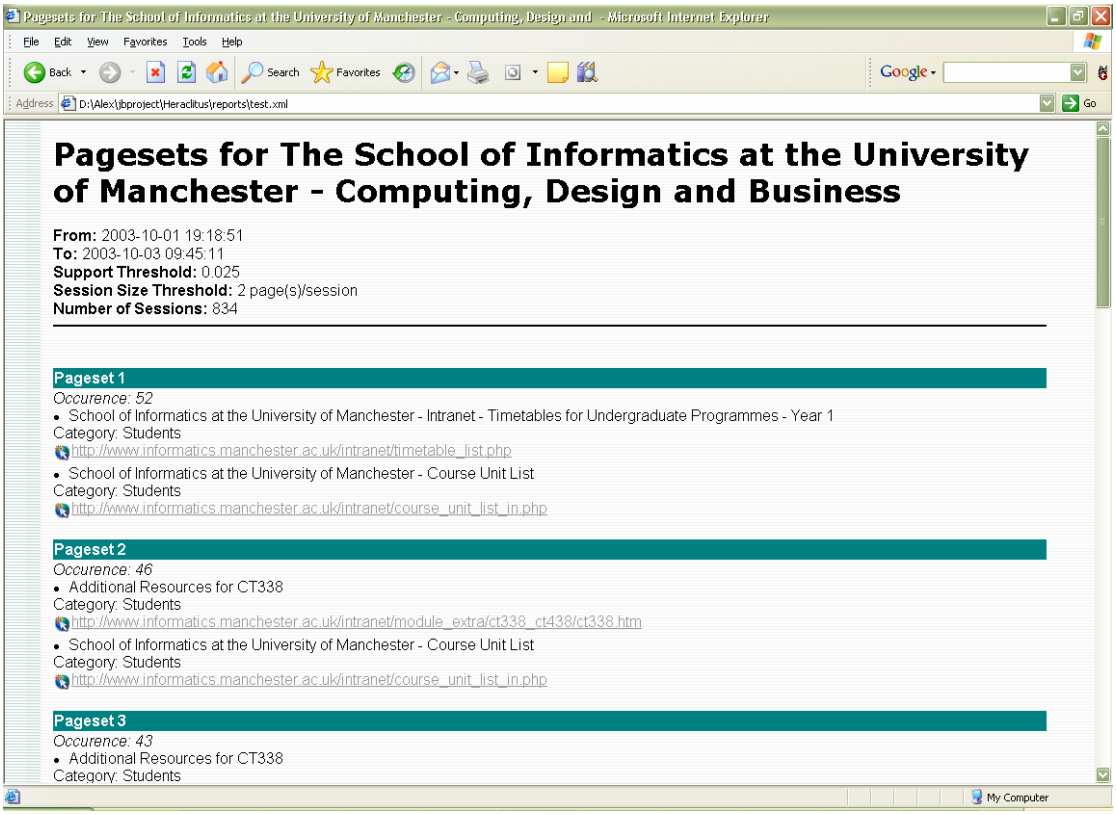


Figure 10. Sample pagesets

Since the output of session mining is a standard XML file, we can edit it with any XML editor. From the **Edit Pagesets** pane we can start the XMLPro[6] editor.

2.5 Site Adaptation

We can now classify the pagesets according to their features and generate reports with adaptations for the web site. The pane shown in Figure 11 lets us specify the classification criteria. The first criterion (topology) refers to the relations that the pages of each pageset have, according to the site topology. The key factor is whether the pages contained in a pageset, are directly linked to each other or not. Pagesets of unlinked pages suggest the insertion of shortcut links between these pages, in order to achieve shorter navigation paths. From the pagesets of linked pages, changes in the appearance of existing links can be extracted. For example, if an index page and some of its links comprise one or more pagesets, then by highlighting these links in the index page, valuable help would be provided to first time visitors. The second classification criterion is based on the content of the pages contained in the pagesets. This classification can reveal inconsistencies in the organization of the site's thematic categories.

 *Tip*: You can define combinations of classifications, e.g. pagesets of unlinked pages that belong to the same content category.

We can also generate reports proposing shortcut or highlighted links (Figure 12). For example, by selecting the shortcut links option, we will come up with the report of Figure 13 (the displayed page titles may vary depending on changes made to the corresponding web pages). The third proposed adaptation regards the page containing the Course Unit List. This is the list of all courses taught in the School. Shortcut links to the resources of popular courses have been proposed to be inserted in this page.

The report is in XML format. We can edit it by pressing the **Edit Report** button, which starts the XMLPro editor.

[6] http://www.vervet.com

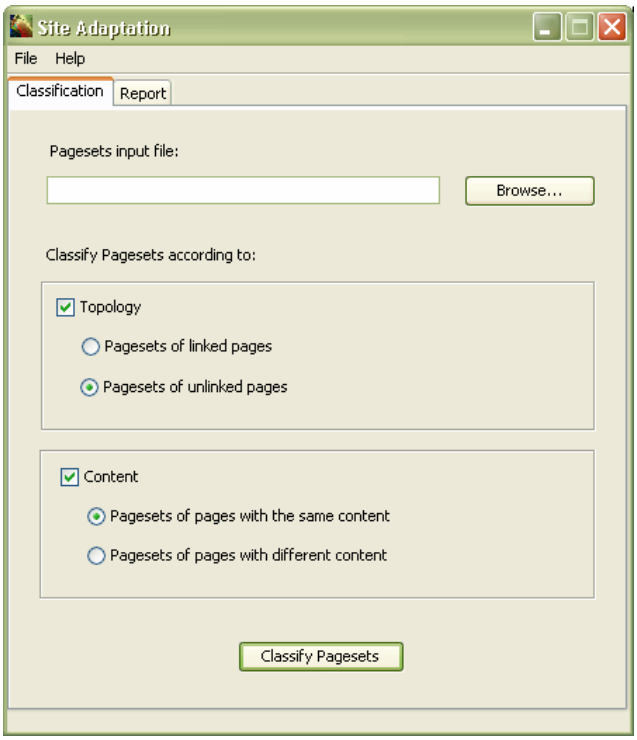


Figure 11. Pagesets classification options

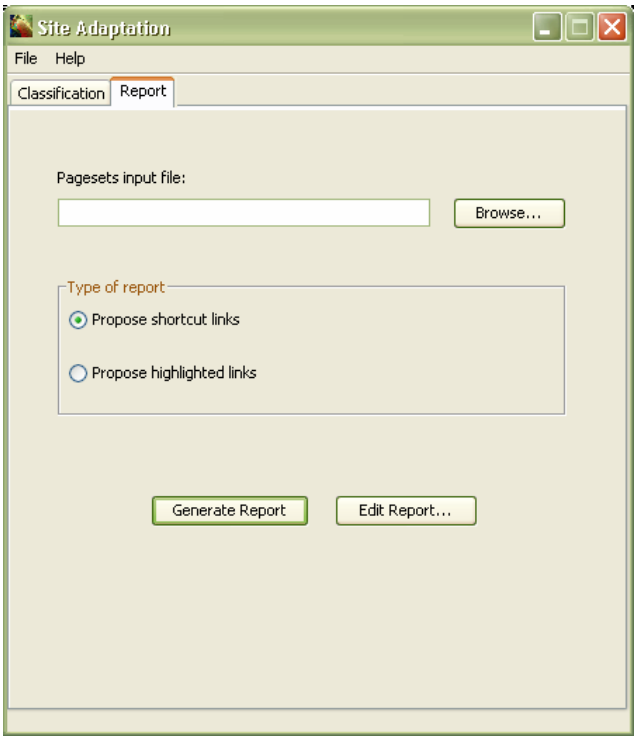


Figure 12. Reporting options

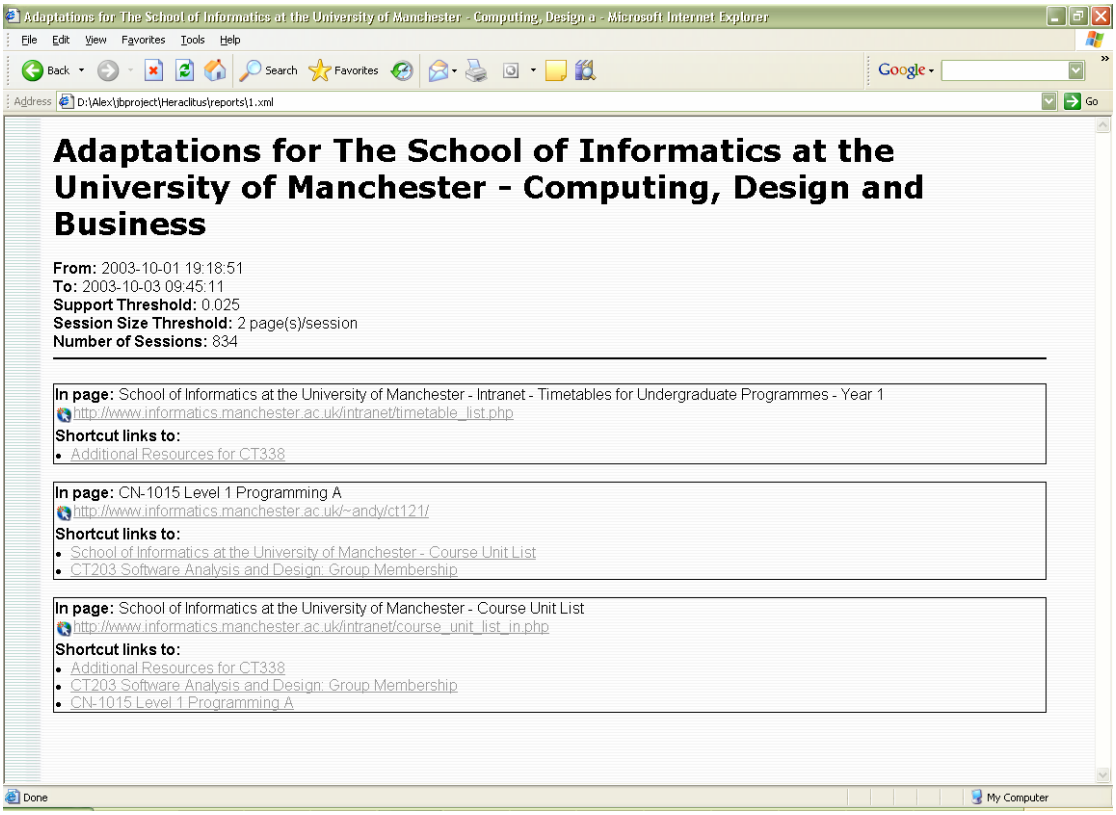


Figure 13. Sample report

It is worth mentioning that the results we will get after the analysis of the sample dataset (**Anonymized_Sample.log**) are quite limited, due to the amount of the recorded accesses. This dataset is provided just for familiarization with the HERACLITUS platform. Due to confidentiality restrictions, we are unable to provide a larger dataset. The results of the adaptation that was carried out for this web site using access data of a whole year are reported in the publications cited in section 1 of this guide.

3 Acknowledgements

The SVM categorization algorithm was implemented by Ilias Kapoutsis (ekapoutsis@mycosmos.gr) as part of his MSc in the School of Informatics, University of Manchester, 2003.