OSSES: An Online System for Studies on Evaluation of Systems

Catherine Mulwa, Mary Sharp, Vincent Wade, Seamus Lawless
Centre for Next Generation Localisation (CNGL),
Knowledge and Data Engineering Group,
School of Computer Science and Statistics,
Trinity College, Dublin,
mulwac@scss.tcd.ie, Mary.Sharp@scss.tcd.ie, Vincent.wade@scss.tcd.ie, seamus.lawless@scss.tcd.ie

Abstract: Web crawlers are a significant component in Web search engines. This paper describes a distributed, Web-based system consisting of 3 layers: presentation, business logic and data persistence. The educational benefit of this system is in the provision of an interactive reference tool to encourage the evaluation of systems that fulfill certain methodological requirements. The studies collected can be used as a basis of a searchable online database that provides an overview of the state-of-the-art to a scientific community and encourages scientists to evaluate their own systems. The system will also help students identify pitfalls in the planning process as well as in data analysis. The collaborative nature of the tool facilitates the sharing of information among research students. The architecture includes RSS Feed Management, Paper Subscription, SMART URL analysis and Document Downloading. The system has been tested and evaluated showing good performance, the results are presented below.

Keywords: Adaptive Systems, 3-tier Architecture, Object Oriented, RSS Feed, Search Engine, SMART URL and Web Crawler.

1. Introduction

Due to the enormous increase of linked information on the World Wide Web (WWW), it has become increasingly difficult for search engines to find appropriate information. The use of large scale search engines, such as Google, is very common when surfing the WWW. The capability of these search engines is impressive. Search engines typically have five components: a crawling module for content discovery, an indexing module for content representation, a relevance estimation and ranking module, a query-document similarity module and a content repository (Olston & Pandey, 2008). The crawling module is responsible for the process of discovering and downloading documents from the internet. This process is completed by 'Web crawlers'[1] which start with a set of seed URLs, download Web pages and extract links from the downloaded pages for further download in an iterative process. This paper introduces an '**O**nline **S**ystem for **S**tudies on the **E**valuation of **S**ystems (OSSES) which was implemented to provide a searchable, categorised set of evaluation studies. This will allow interested learners to get suggestions of experimental designs, criteria and other experimental issues. The system is composed of two components: two personalised Web crawlers (RSS Feed and Document Crawler) and an online database providing a list of evaluation studies on systems (e.g. each study has a title, author, published date, link content, citation and reference) and each evaluated system mentioned in these studies which is described in terms of (system name, the functions it fulfils, the purpose, application area, evaluation methods and criteria used and data type analysis). The Object-oriented approach to software engineering was adopted which involved six development activities; requirements elicitation, analysis, system design, implementation, testing and evaluation (Bruegge & Dutoit, 2009).

The goal of the research described in this paper was to: i) design and implement an online system capable of crawling and automatically retrieving the most recently published studies on the evaluation of systems that fulfill certain methodological requirements; ii) implement an online database which will be used to identify gaps in the state of the art and act as a guide for planning new evaluations; iii) encourage new researchers from different domains to perform research on the evaluation of systems; iv) provide evaluators and new PhD students a tool which will help reduce cost and time spent conducting literature reviews; and v) improve the learning process by providing a system which is personalised to each user's needs.

The main educational benefit of OSSES is to provide an interactive reference tool to encourage the evaluation of systems that fulfill certain methodological requirements. The studies collected can be used as a basis of a searchable online database that provides an overview of the state of the art to the scientific community and encourages scientists to evaluate their own systems. It will also help students identify pitfalls in the planning process as well as

[1] Also sometimes referred to as 'web robots' or 'web spiders'

in the analysis of collected data and also identify omissions in the state of the art. For people outside the community, the system will serve to teach them the strengths and weaknesses of certain types of systems. The system can also be used to inform student researchers of currently used methods and criteria as well as omissions and problems in current evaluations. The contribution of this research supports education because it provides new knowledge and a more complete view of the state of the art to researchers. The collaborative nature of the architecture enables the sharing of information among similar users. Currently, OSSES is being used to collect and analyze studies on user evaluations of adaptive systems in particular adaptive systems which combine Adaptive Hypermedia (AH) and Information Retrieval (IR) techniques.

The OSSES is a significant development for new researchers in the field of evaluation of systems since it provides an interactive reference tool to encourage the evaluation of systems that fulfill certain methodological requirements. For example if a paper has been published on the Web titled "Assessment on the Adaptivity of Adaptive Systems". Instead of manually searching for that title (e.g. Google scholar), the paper will be retrieved automatically and stored in a local repository. The system searches the first ten pages in Google and from each page; it retrieves the first four PDF documents and also eliminates duplicates. This is very significant because due to the large amount of available information, the Google search engine generates a set of documents dynamically, which results in a different result for every new search query, in our case, the query is an RSS feed. Each study is then manually analyzed to extract the system name, function, application area, evaluation method, criteria used, purpose of the system, and data type analysis. This information is then transferred to the online database. The database will help identify pitfalls in the planning process of evaluations as well as in the analysis of the collected data. The OSSES is better than Google scholar because Google scholar return query results by popularity but while this system focuses on most recently published papers. The OSSES system also allows users to query meta-data specially designed for specific users (e.g. system name, system functions, application area, evaluation method, criteria used, data type analysis and system purpose). It is important that evaluators avoid well-known pitfalls and that writer(s) of future evaluation reports increase their empirical value by reporting the methodology used and results in such a fashion that replication of the study is possible. We are convinced that the quality of evaluations will benefit and that, indirectly, the user will be served in the process. The system can also be used to serve as a reference for researchers in the field of system evaluation and assurance.

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 describes the OSSES system implementation. This section also introduces the system methodology, architecture, crawling process, database design and system functions. Section 4 presents the performance testing and evaluation. Section 5 discusses the advantages and limitations of the OSSES. Finally, Section 6 concludes the paper and recommends future work.

2. Related Work

Evaluation is an important tool in software quality assurance. A typical software quality program involves: i) establishment, implementation, and control of requirements; ii) establishment and control of methodology and procedures and iii) software quality evaluation (Farooq, 2008). The software quality evaluation component is aimed at evaluating products (both in process and at completion), and methodologies (for appropriateness and technical adequacies). Evaluation of all systems is important. It is important to not only evaluate but also to ensure that the evaluation uses the correct methods (Brusilovsky, 2004). Evaluation is defined as the process of examining the product, system components, or design, to determine its usability, functionality and acceptability (Weibelzahl, 2003), which is measured in terms of a number of criteria essential for any software development project. Why should evaluation take place and who are the stakeholders? It is carried out by designing an evaluation plan which consists of: i) specifying goals, ii) decomposing those goals into evaluation questions, iii) setting criteria for deciding questions, iv) identifying data required to answer questions, v) selecting methods for collecting and analyzing data, vi) conducting experiments either in the experimental lab or in the field, vii) collecting data that can be qualitative and/or quantitative, or subjective and/or objective viii) analyzing and interpreting the data and xi) reporting and drawing conclusions based upon the established hypothesis. System evaluation places an emphasis on the comparison of the presented system with established criteria proposed by other researchers or other related systems.

It has been stated that the behaviour of a Web crawler (a program that automatically traverses the Web's hyperlink structure and retrieves some information for the user.) is a combination of policies: i) **A selection policy** that states which pages to download. If the set S is implemented as a stack, the algorithm visits the Web sites in a Breadth-first manner; ii) **A politeness policy** that states how frequently to visit the Web pages and what portions of sites to visit. The use of Web crawlers is useful for a number of tasks, but comes with a price for the general

community. The costs of using Web Crawlers include network resources and server overload. A partial solution to these problems is the robots exclusion file which indicates which Web sites (or portions of those sites) should not be accessed by crawlers; iii) **A parallelization policy** that states how to coordinate processes or threads in the downloading phase. Crawlers are used by search engines to make a local copy of the Web. This copy is then indexed for further fulfillment of user queries; and iv) A **re-visit policy** that states when to check for changes to the Web (Girardi, Ricca, & Tonella, 2006).

There is limited research on user evaluation of adaptive systems. An adaptive system tailors its output, using implicit inferences based on interaction with the user and explicit information provided by the user. The evaluation of adaptive systems is complex and difficult. Several researchers have pointed out potential pitfalls when evaluating these systems (Tintarev & Masthoff, 2009) (S Weibelzahl, 2005) including: i) Difficulty in attributing cause, ii) Insignificant results due to variance between participants, iii) Difficulty in defining the effectiveness of adaptation, iv) Allocation of insufficient resources, v) Too much emphasis on summative rather than formative evaluation and vi) Most importantly measures for adaptivity success have not been investigated systematically up to now. A few researchers have compiled a synopsis of studies published on the evaluation of adaptive systems; Weibelzahl compiled a synopsis of 43 studies published in the UMUAI journal and in the proceedings of User Modeling Conferences (1997 -1999), and considered the most important works in the area of adaptive systems. He accepted that about a quarter (11/43) of the studies examined, had either only a single user, hypothetical users or that the sample size was not reported. Actually, only 14 out of 43 studies were high quality in terms of sample size and statistical analysis. He also reported that the most frequent measures include accuracy, precision and recall, domain knowledge and duration of the interaction. Last year Van Velsen conducted a literature review study identifying 63 studies on user-centered evaluations of adaptive and adaptable systems (Van Der Geest T, Van Velsen, Klaassen, & Steehounder, 2008). Of the systems mentioned in these studies; 23 systems were adaptive systems, the authors fail to identify which studies were only on user centered evaluations of adaptive systems. During the process of conducting this literature review it was discovered that although a few researchers have researched the evaluation of adaptive systems, very little research has been conducted on user evaluation of adaptive hypermedia systems (AHS). AHS are systems composed of three main elements: user models, domain models and adaptation models. The user model is based on the user characteristics and can contain information such as prior knowledge among other things. The domain model can contain information such as detail of the subject area that the system is built around. Finally, the adaptation model describes how the navigation will be carried out and also how the material will be presented.

The past 16 years have seen notable progress in the field of AHS. Many systems were created and some of them brought new, interesting, and unusual ideas. Unfortunately a few were almost forgotten since their documentation was difficult to find; thus the importance of the OSSES system. A few researchers have implemented online databases with similar functions to the OSSES, but some of these databases are out of date (Weibelzahl & Weber, 2001) making it difficult to obtain clear and up to date metadata for the evaluation of adaptive systems. This system is a valuable tool for new PhD students since it will help reduce cost and time spent conducting literature reviews.

3. OSSES System Implementation

3.1 Our Methodology

The Rational Unified Process (RUP) Methodology was used. By adopting the RUP method the developer was able to manage the Object-oriented software development approach since it is extensible and has features such as: i) iterative development, ii) requirements management, iii) component-based architecture vision, iv) visual modeling of the system, v) quality management and vi) change control management.

3.1.1 System Architecture

The OSSES is designed as typical 3-tier architecture (see Figure 1) which consists of: **i) the presentation layer** which is the topmost level of the application that displays information related to services such as browsing. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network; **ii) The business logic layer** which is pulled out from the presentation tier and, has its own layer, it controls an application's functionality by performing detailed processing, and **iii) the data persistence layer** which keeps data neutral and independent from application servers or business logic. Giving data its own tier improves scalability and performance. The presentation tier is implemented using Java Server faces (JSF) technology. By leveraging the

Apache MyFaces library, two modules are implemented, namely: the Administration Interface Module and the End User Interface Module. The business logic tier consists of four modules namely: i) RSS Feed Management, ii) Self-Monitoring, Analysis, and Reporting Technology (SMART) Analysis, iii) Document Downloading, and iv) Paper Subscription. The data persistence tier consists of a MySQL database and Java Persistent API (JPA)

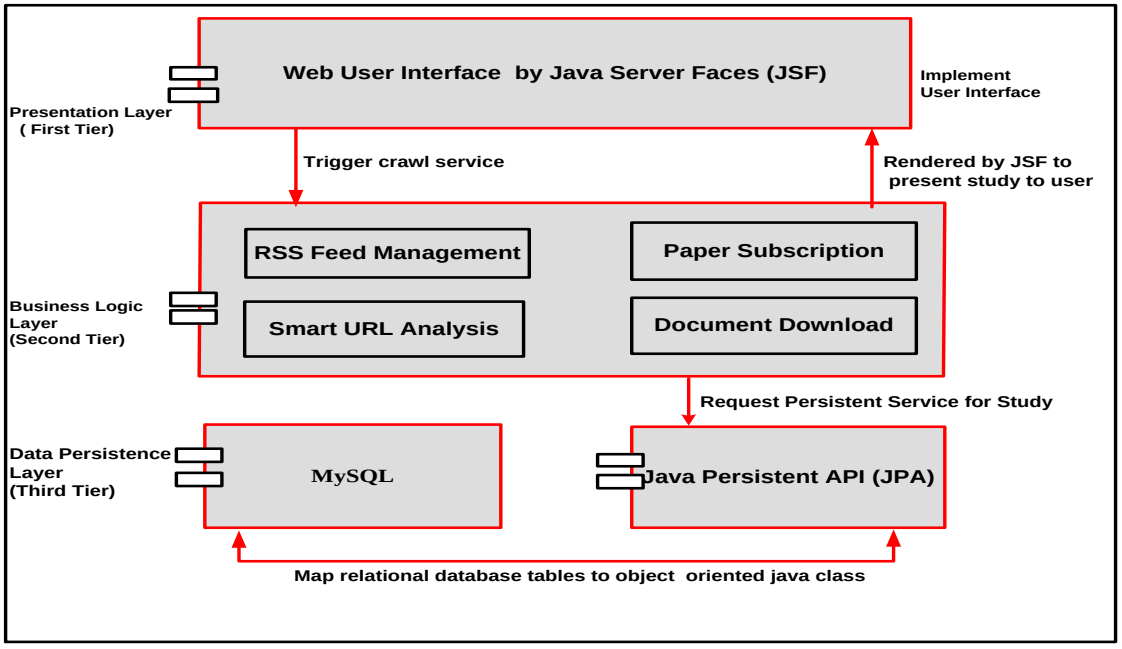

Figure 1 OSSES Architecture

The core of the system is the business logic layer in the middle (see Figure.1), the RSS Feed Management allows users i.e., system administrators, to manage a set of Web feed formats that will publish most recent papers to the user. As soon as a paper is published via RSS feed, the Paper Subscription module will create metadata for that particular paper, such as title, author, publication date, and most importantly the URL for the hard copied paper. Ideally, the URL is supposed to download a published paper document directly. But in the real world, the URL can point to an indirect location, such as a Web page containing a link to the actual document or a search result pointing to the document. To solve the indirect URL problem, the SMART URL Analysis Module is used to testify if the URL is directly pointing to document. If not, it will do a programmatic search in Google Scholar by entering the title of the paper and return the first downloadable link to Document Downloading module. Upon receiving the document link, the Document Downloading module will automatically retrieve the document using the Client URL (cURL) tool and store it to a local repository.

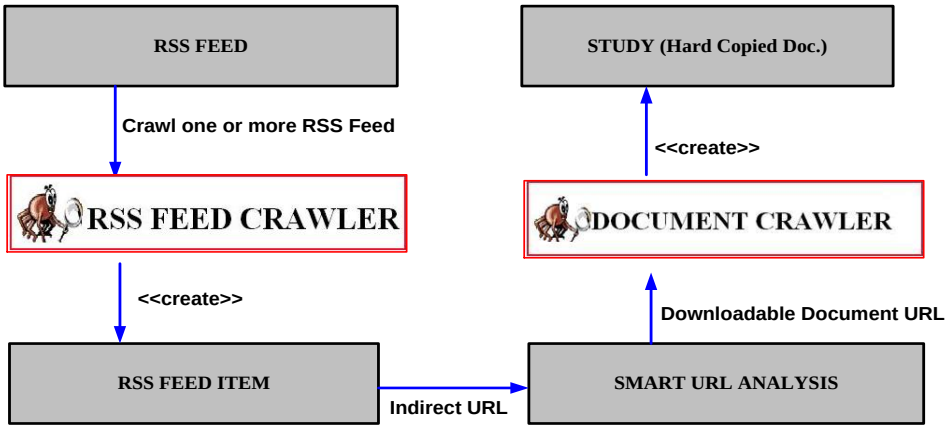

Figure 2 Crawling Process

3.1.2 Data Flow Process

The process of crawling to retrieve a study is shown in Figure 2. The user (e.g., System Administrator) triggers the crawling process, a RSS feed is retrieved by the RSS Feed Crawler. The RSS Feed Crawler sends a request to get the most recently published papers, and then creates one or more RSS Feed Item(s). An RSS Feed Item contains metadata about the published papers such as (title, author, published date time and a URL to the paper document) which is generated automatically. Subsequently, the URL to the paper document is passed to the SMART URL Analysis. If the URL is a downloadable document link, the analyzer will leave the URL untouched. Otherwise, the analyzer will try to ascertain the downloadable URL for the paper as discussed above. Finally, the downloadable URL is passed to the Document Crawler which uses Client URL (cURL) to retrieve the document and create a hard copy on the local repository.

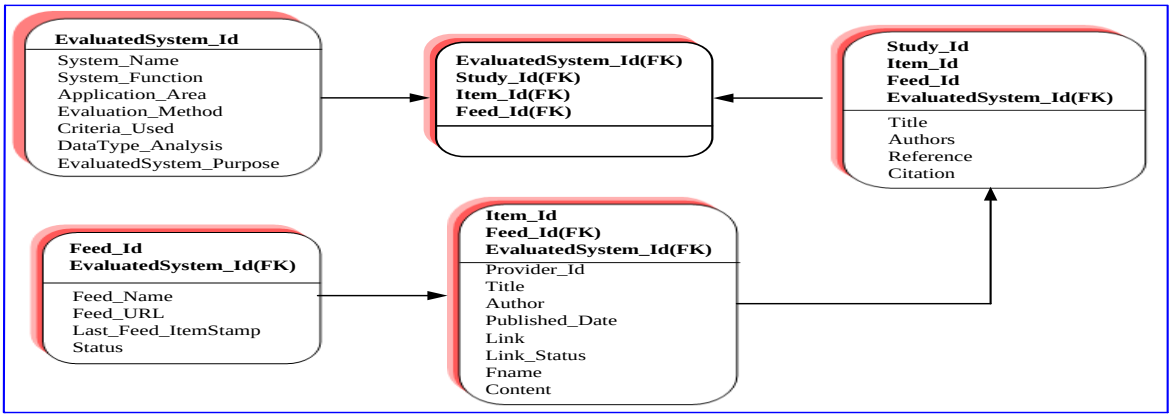

Figure 3: OSSES Database Design

3.1.3 OSSES Database Design

For each published study, the system automatically retrieves the study. Each study(s) has a title, author, published date, link content, citation and reference. Each evaluated system mentioned in this study(s) is described in terms of: system name, the functions it fulfils, the purpose, application area, evaluation methods and criteria used and data type analysis. If the Web crawler retrieves a study that is not relevant, the system administrator should delete it from the database (see figure 4). Administrator User Interface requires authentication at login. As a system administrator, the user can perform all the transactions (e.g., select study or system, view transaction, search for studies or system, modify system operations, delete or edit transactions, add system and study details) see Figure 4 and 5.

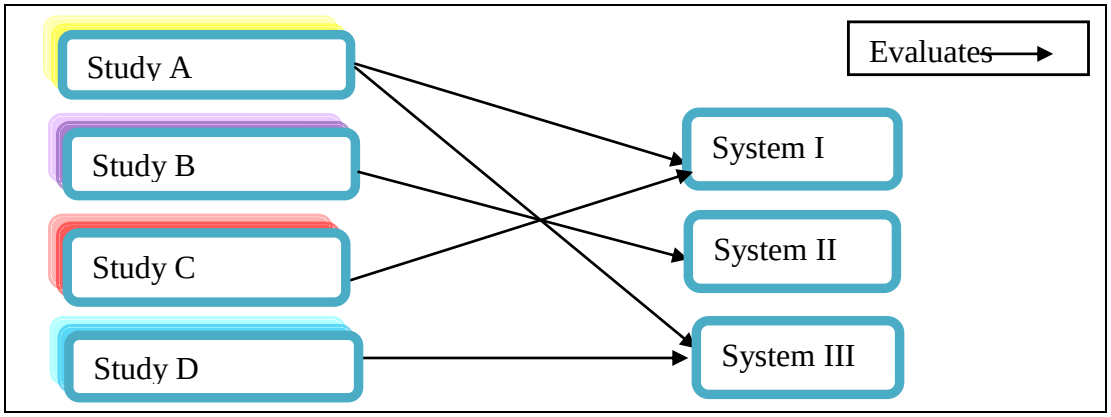

Figure 3.1: Relation of evaluation studies and evaluated systems

The relationship between evaluation study(s) and evaluated system(s) is demonstrated in figure 3.1 (e.g. while one study might evaluate one or more systems (e.g. Study A), a system might be evaluated by one or more studies (e.g. System III).

3.1.4 Software Components

The OSSES system is implemented by integrating current technologies and software: (**Eclipse** - platform, **Apache-Openjpa** - To store and Retrieve data from database, **Apache-Tomcat** - server, **Client URL** (cUrl) tool-Downloading Tool for PDF, **Myfaces-core** - Java Server Faces (JSF) used to display data in the Web, **MySql-win32** - MySql database server, **MySql-connector-java** - Connector for java to communicate to mySql, **Json** - To parse RSS Feed and **Self-Monitoring, Analysis, and Reporting Technology** (SMART) URL analysis – a system for monitoring computer hard disks to detect and report on various indicators of reliability) which has led to an increase in performance and a better User Interface.

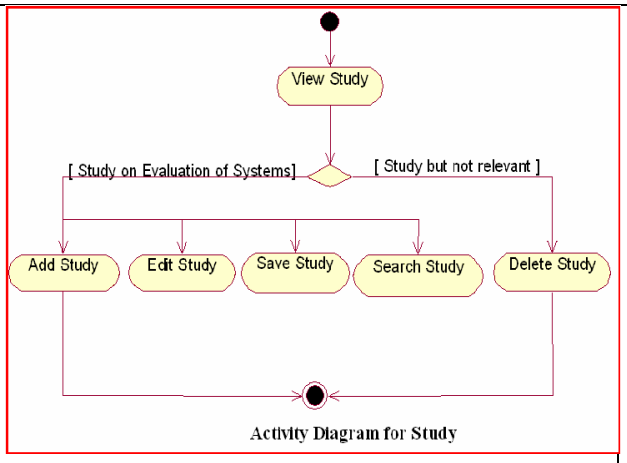

Figure 4: Activity Diagram

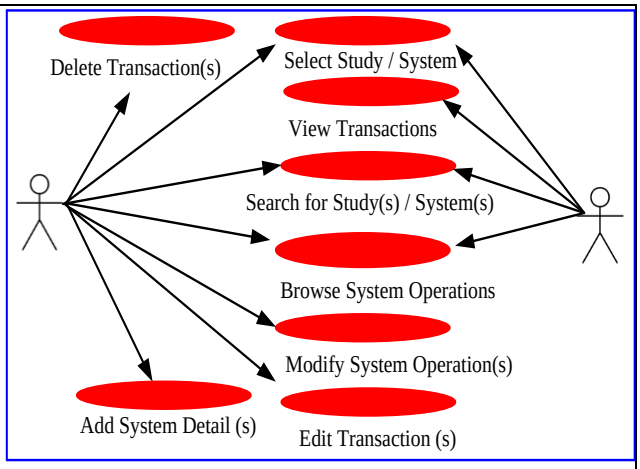

Figure 5: Use Case Diagram

3.2 The OSSES system Functions

The system is composed of two user interfaces[2]: system administrator and end user (see Figure 6 and 7)

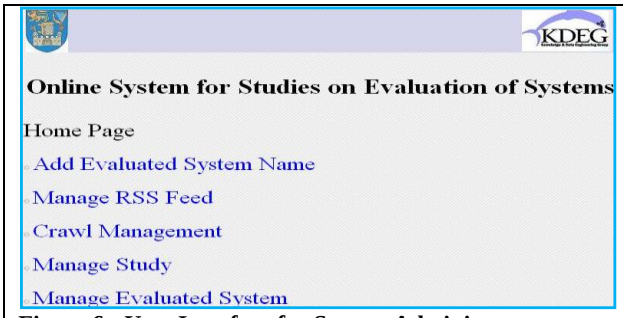

Figure 6: User Interface for System Administrator

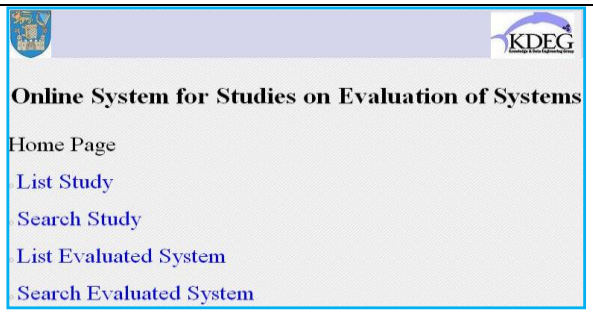

Figure 7: User Interface for the End User

The system administrator has full control of all transactions (see figure 4, 5, 6 and 7). Other users can only view study(s), search for study(s), view evaluated system(s) details and also search for evaluated system(s) (see figure 5 and 7). Following is a brief description of the system functions and activities the user performs:

[2] http://kdeg-vm-10.cs.tcd.ie:8080/OSSES/faces/admin/

A. Administration Functions

1) **Add Evaluated System Name:** This function allows the system administrator to quickly; add, edit, and view each evaluated system details (e.g, system name, functions, application area, evaluation method, criteria, system purpose).

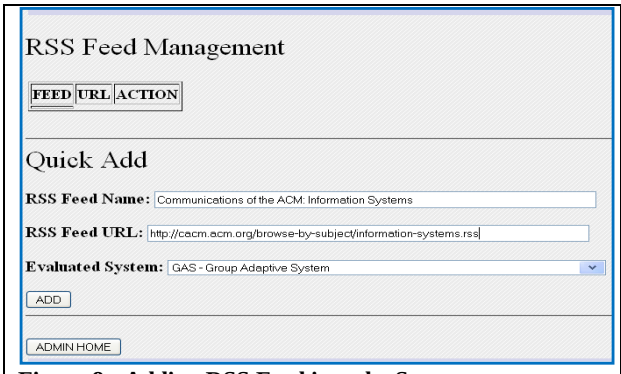

Figure 8: Adding RSS Feed into the System

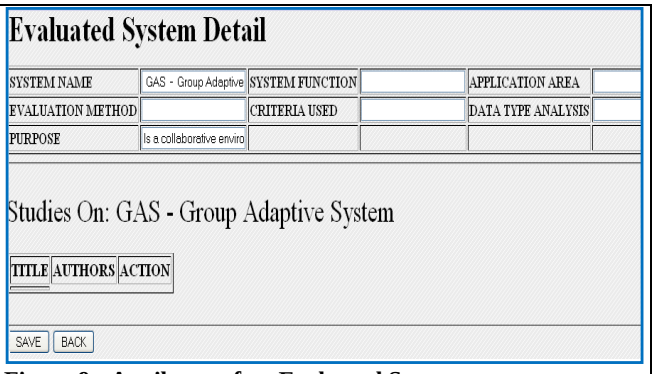

Figure 9: Attributes of an Evaluated System

2) **Manage RSS Feed:** Once a system name has been added, the RSS feed management function allows the administrator to: add feed name and the URL link. It also provides a drop down list of Systems which were added in function 1. (see figure 8). Once the feed has been added the user is able to see the feed title and the URL link (see figure 10)

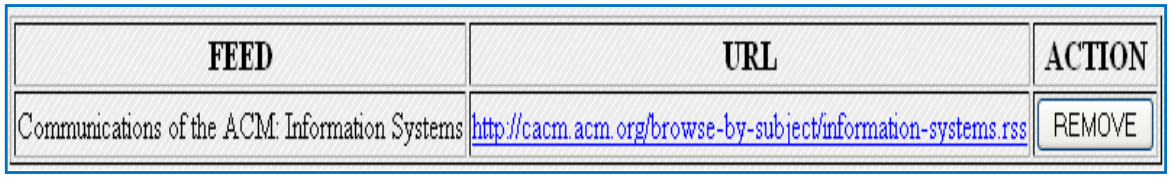

Figure 10: Crawled Feed and the URL link

3) **Crawl Management:** The crawl management function performs two tasks: first the system administrator has to click "crawl studies" before clicking "crawl documents". This task can only be performed when function 2 has been completed (see figure 11).

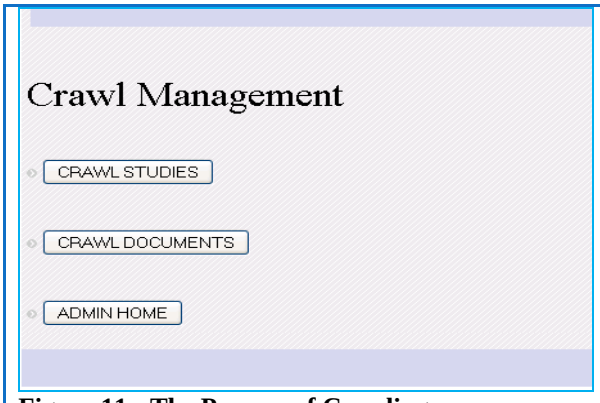

Figure 11: The Process of Crawling

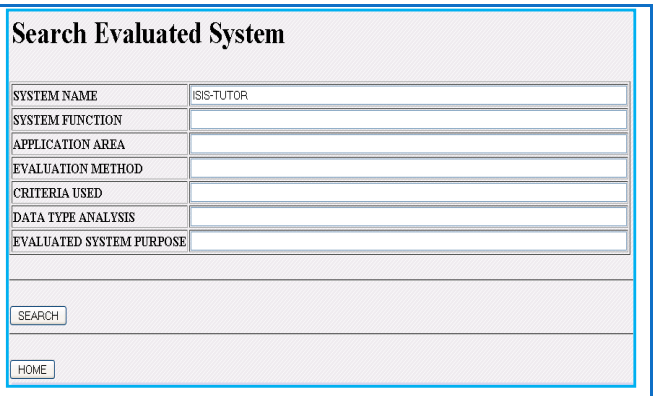

Figure 12: Searching for a System

4) **Study Management:** This function allows the system administrator to view all crawled studies from function 3 (e.g., title, authors) and perform actions such as: editing the study details (e.g., title, authors, reference, citations) and also it provides a quick link to the drop down list of all the Evaluated Systems (see figure 13).

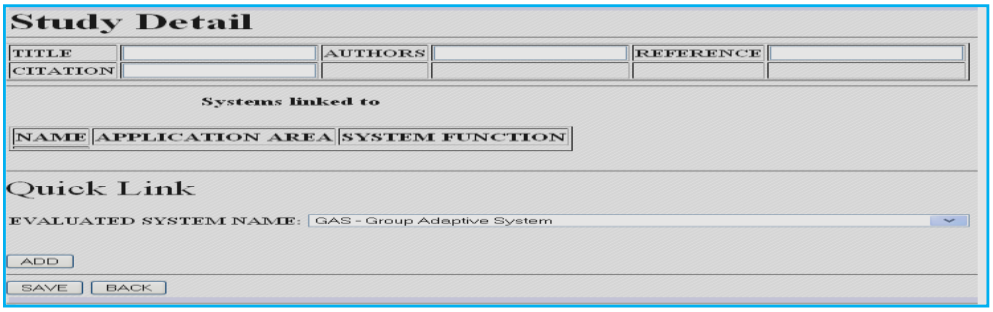

Figure 13: Attributes of Published Paper

5) **Evaluated System Management:** This function allows the administrator to enter details of each evaluated system (e.g., name, function, application area, evaluation method, criteria and purpose), perform actions such as editing and viewing which study(s) are linked to that particular system (see figure 9).

B. Usage Functions

6) **List of Studies:** The user can: (e.g. view or search for study(s) title, author, reference, citation, systems linked to that study(s)).
7) **List of Evaluated Systems:** This function provides a list of all the evaluated systems described by each study(s). It also allows users to perform actions such as: i) view details of the evaluated system and , ii) the studies (e.g., title and authors) describing that particular system.

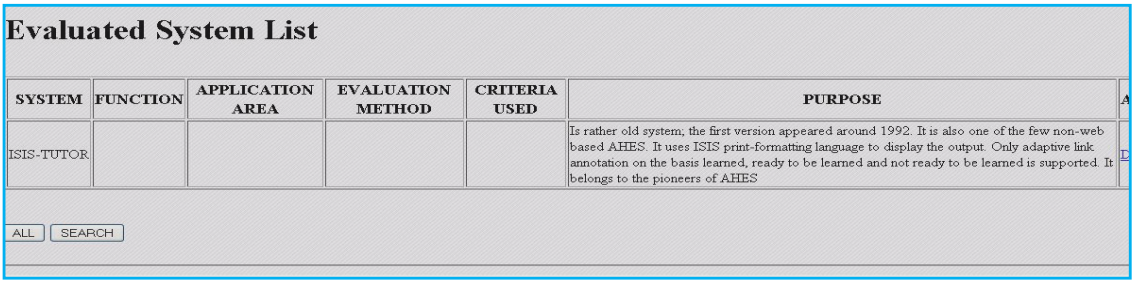

Figure 14: Retrieved System after Search Query

8) **Search Evaluated System:** The user is capable of searching for existing systems by using the following search terms(i.e, system name, function, application area, evaluation methods, criteria used, data type analysis, evaluated system purpose). For example if a user is searching for a system named "ISIS-TUTOR" (see figure 12) and it exists in the database, it will be displayed with all other details relating to that system (see figure 14).

4. OSSES Testing and Evaluation

The validation of the OSSES system prototype was subdivided into two distinct tasks; functional verification and efficiency evaluation. Functional verification was used in order to verify all the functional requirements and the efficiency evaluation to ensure that user satisfaction was accomplished. We used different software testing elements: i) **Methods and techniques** which included information retrieval techniques, interviews, expert reviews

and log file production during downloading the crawler produces a log file containing some information on the PDF documents of the downloaded studies, ii) **Process,** iii) **Empirical knowledge,** and iv) **Tools** including Self-Monitoring, Analysis, and Reporting Technology (SMART) Analysis which is a monitoring system for computer hard disks to detect and report on various indicators of reliability, in the hope of anticipating failures during the process of crawling.

Two different types of evaluations were conducted, **formative** and **summative** evaluations. Formative evaluation was conducted during the implementation process. A range of evaluation methods were used: i) interviews with domain experts and ii) task analysis and iii) retrieval accuracy. To evaluate the crawler, we considered features such as completeness, robustness, time and controlling the number of downloaded documents and the graphical User Interface. The OSSES system was evaluated internally by: i) Our research supervisors; ii) Presentation of the deployed system to a group of researchers in knowledge and data engineering research group (KDEG). KDEG is pioneering research into the fundamental challenges and application of knowledge-driven systems. The group combines innovative technology research in knowledge discovery, representation, reasoning, data management and intelligent systems engineering. The following list contains a selection of questions which were asked by evaluators and the relevant responses to these questions:

- *Why should we use your system while we can use Google search engine?* It will save the end users time and also encourage research in the area of evaluations of systems.
- *Why the use of different technologies and software?* These technologies and software solutions were used in order to increase the system performance and also for a better user interface.
- *What are your future plans for this system?* The future plan is to add more functions that are specific to user evaluations of adaptive systems and continue to populate the database.
- *How did you test the system?* The system was tested by functional verification and efficiency evaluations.
- *How do you deal with retrieved studies that are not relevant?* The system administrator deletes irrelevant studies, as demonstrated by the activity diagram (see figure 4).
- *Can new functions be added that are relevant to our research?* Yes.

One of the research students interviewed indicated that they wanted to start using the system immediately.

In addition, summative evaluation was conducted to provide information on the system's ability to perform better. In order to determine how well the system performed, several **evaluation criteria** were used:

- Evaluation of input data (e.g., objectivity of data assessment, retest-reliability),
- Evaluation of adaptation decision (e.g., retrieval accuracy, precision and recall,
- Amount of help required, computational time; number of navigation steps,
- Task success, user satisfaction), usability satisfaction, effectiveness,
- Reliability, functionality, performance, time, robustness,
- Downloading limiting and completeness.

5. Discussion

5.1 OSSES Advantages

- The system provides services that assist researchers in their Web navigation by automating the task of link traversal, creating a searchable index of the Web and fulfilling searchers' queries from the index.
- Our Web crawler's simplification of the Web experience makes the Web a more friendly and useful tool.
- This system is useful because it can provide some context for a searcher's particular query: by issuing a well-formed query, a searcher can find the breadth of information about that particular topic and can use that information to further refine his goal.

5.2 OSSES Limitations

- All the retrieved study(s) do not provide author name(s), results returned are "Unknown author".
- Once a study(s) is retrieved, the system administrator has to process the information and manually enter this information in the database.

6. Conclusion and Future Work

This research will support PhD students by encouraging new researchers from different domains to research in the area of the evaluation of systems which fulfill certain methodological requirements. It will serve as a reference for researchers in the different fields of evaluations of any kind of system; for example research on user evaluations of adaptive systems especially those that combine AH and IR techniques. The online database will help to identify gaps and pitfalls in the planning process of evaluations as well as in the analysis of collected data. It is crucial that evaluators evade well-known pitfalls and that writers of future evaluation reports increase their empirical value, by reporting the used methodology and results in such a fashion that replication of the study is possible. A user who wants to use a Web crawler has two choices: building it from scratch or downloading one from the internet. The second option has some drawbacks such as the user deciding which one to choose? Which is the best for the task at hand? Which is most complete? Which is the most robust? We provide a fully functional personalised Web crawler which is capable of automatically retrieving recent published studies. We are convinced that the quality of evaluations will benefit and that, indirectly, the user will be served in the process.

In future extra functions will be added that are specific to research on user evaluations of adaptive systems. Find an algorithm which will be used to automatically extract references and finally modify the crawler to enhance the performance capability.

Acknowledgements

This research is based upon works supported by Science Foundation Ireland (Grant Number: 07/CE/I1142) as part of the Centre for Next Generation Localization (www.cngl.ie). The authors are grateful for the suggestions of the reviewers for this paper.

References

Bruegge, B., & Dutoit, A. (2009). *Object oriented software engineering*: Prentice Hall.

Brusilovsky, P., Karagiannidis, P., and Sampson, C. (2004). Layered Evaluations of Adaptive Learning Systems. *14*, 402-421.

Farooq, A., Dumke, Reiner R. (2008). *Evaluation Approaches in Software Testing*: Faculty of Computer Science, University of Magdeburg.

Girardi, C., Ricca, F., & Tonella, P. (2006). Web crawlers compared. *International Journal of Web Information Systems, 2*(2), 85-94.

Olston, C., & Pandey, S. (2008). *Recrawl scheduling based on information longevity*.

Tintarev, N., & Masthoff, J. (2009). Evaluating Recommender Explanations: Problems Experienced and Lessons Learned for the. *UMAP 2009*, 54.

Van Der Geest T, Van Velsen, L., Klaassen, R., & Steehounder, M. (2008). User-centered evaluation of adaptive and adaptable systems: a literature review. *The Knowledge Engineering Review, 23*(03), 261-281.

Weibelzahl, S. (2003). *Evaluation of Adaptive Systems.* University of Trier, Trier.

Weibelzahl, S. (2005). Problems and pitfalls in evaluating adaptive systems. *In: Fourth Workshop on the Evaluation of Adaptive Systems in conjuction with UM'05*, 57-66.

Weibelzahl, S., & Weber, G. (2001). *A database of empirical evaluations of adaptive systems*.