WEB-BASED PLATFORMS FOR COLLABORATIVE SUBMISSION AND REVIEW OF SHARED INFORMATION

BOB MARTENS
Vienna University of Technology
b.martens@tuwien.ac.at

WASSIM JABI
New Jersey Institute of Technology
jabi@njit.edu

TOMO CEROVSEK
University of Ljubljana
tcerovse@fgg.uni-lj.si

HANNU PENTTILÄ
Helsinki University of Technology
hannu.penttila@mittaviiva.fi

Abstract. The focus of this paper is the implementation of web-based data platforms for CAAD research and education. This review is based on three case studies that use different software environments for the creation of the specified data platforms: eCAADe/FileMaker-databases, ACADIA/CAMEO-WebObjects environment and SciX/WODA-language.

1. Introduction

In the pre-internet age the organization of an annual conference was mainly based on conventional paper mail. Submitters would send their abstracts to a conference secretary, who would duplicate and forward them to reviewers. Once the reviewers returned their reports, again by conventional mail, final notifications would then be sent in the same way. The whole procedure took a relatively long time and data had to be retyped again and again.

The use of fax-machines increased the speed of communication. Nevertheless, this caused additional work in the duplication and distribution of information. In this respect a self submission of data by authors was limited to the submission of a piece of paper or a fax. Sending a diskette containing a data document - still by conventional mail - became popular as soon as personal computers became widely available. When the internet became widely available to the academic community, alternative ways of transmitting and handling data were provided. Although the use of attachments to an email-message is still used to transfer bits and bytes of data (with a certain level of manual handling), several web-based interfaces for different kinds of collaboration have been developed.

In due course the use of internet applications became increasingly sophisticated, avoiding the disadvantages of slow conventional mail and manual handling of information from isolated data documents. Especially for the organization of annual (CAAD-) conferences submission and review sites were set up and further developed from year to year. In this paper, the development of different approaches of related database applications including experiences gained will be elaborated.

2. Frame Conditions on the Development of Platforms

The aim of this contribution is first to provide an overview of the necessary conditions for setting up a web-based database system for entry and submission of conference abstracts. However, the scope will be expanded from the narrower utilization area of handling abstracts towards remote collaboration in a much wider sense of academic activities. Three interactive web-operable database environments will be described here and interesting differences in approach can be observed. The WebObjects-based ACADIA/CAMEO-interface - developed by Wassim Jabi - has been applied since 2001. Hannu Penttilä designed the eCAADe-solution, which is based on FileMaker and has been in use since 2001. The third interface, which will be examined is a solution using the Woda Web Oriented Database System. It was established by Ziga Turk (University of Ljubljana) originally for the CIB W78 Workshop in 1996 then adapted for th ECPPM-conference series in 2002 and 2004 and adapted by Tomo Cerovsek for IAPS in 2004. The latter solution is part of the SciX Open Publishing Services (SOPS). Bob Martens supported Penttilä in terms of setting up wish lists for further development of the eCAADe solution as well as discussing/testing new features before final implementation. Furthermore he served as the main user for the IAPS-2004-site.

2.1. ACADIA/CAMEO-INTERFACE

CAMEO, which stands for Computer-Aided Membership and Event Organization, is an object-oriented system that enables an academic/scientific organization to administer its members and events. CAMEO was developed as an implementation of a set of conceptual guidelines developed through a historical survey of computer-supported cooperative work (Jabi 2003). CAMEO is composed of three main applications: 1) a membership management system, 2) a conference registration system, and 3) a double-blind review system. All three systems extensively share reusable components that were developed using a web-based three-tier, object-oriented software development and application server system that enables the creation of interactive web-based software. For this paper, we will concentrate on the collaborative features of the double-blind review.

WebObjects

Apple's WebObjects software is a high performance application server, an object-oriented software development environment and a database integration tool (fig. 1). It provides a framework for developing three-tier client-server applications as well as dynamic publishing. The three tiers are (1) Storage, (2) Application-Layer, (3) Web Interface.

Figure 1. Apple WebOjects' tool set

By using components in templates that are linked to actions and data in the WebObjects application, the server can act on user requests dynamically, configure a response HTML page and return that to the user for further action (fig. 2). WebObjects provides mechanisms for maintaining information, called states, during a session or even after a session has terminated. Because of that functionality, multiple participants (clients) can view and edit the same information while being logged on to the same WebObjects application. In addition, WebObjects interoperates with various database systems through a tool called Enterprise Object Modeler (EOModeler) that maps the relationship between a database and a set of objects. Basically, an enterprise object corresponds to one or more records in a database. The WebObjects application manages these objects such that the

developer can concentrate on the business-logic of his/her application. WebObjects' ProjectBuilder allows the developer to create the necessary software to handle the business logic of the application in question. Lastly, a WebObjects or Interface Builder constructs the necessary web templates and connects its elements to software variables and methods.

CAMEO-Review

The CAMEO-Review system allows authors to register for paper submission. The author(s) fill an online form that gets saved in a database and the software generates a unique ID and e-mails it to the author(s). The software instructs the authors to use that code in naming their submission. This ensures the confidentiality of their submission. While the system could in theory allow file uploads, it currently does not make use of that capability. Instead, it instructs the authors to send a PDF file of their submission via e-mail. The PDF file is then saved on a web server.

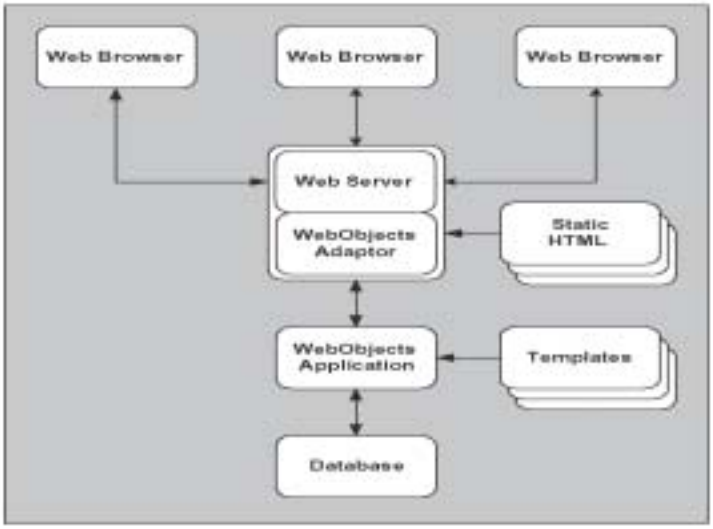

Figure 2. The architecture of WebObjects and CAMEO.

Concurrently, the CAMEO-Review system administrator collects the list of reviewers, enters them into the database and specifies a login and random password for each. Additionally, once the papers have been submitted, they are assigned to the different reviewers through the establishment of a relation in a relational database model. When reviewers log in, they see a list of papers that have been assigned to them and online evaluation form for each paper that includes a numerical scoring system, private comments to the Technical Chair and public comments that are sent to the author(s). The

CAMEO-Review software collects all information and averages the score of different reviewers. It then sorts the papers and presents all information, through a private administrative interface, to the Technical Chair for evaluation. Final notification is conducted via e-mail.

The object architecture of CAMEO-Review includes representations for a Paper, an Evaluation, and a Reviewer. Author information is removed before the system is opened to reviewers. Papers are assigned to multiple reviewers and reviewers can review multiple papers. This is accomplished through a many-to-many relationship (fig. 3). The CAMEO-Review system, while simple in its architecture and implementation, provides some significant insights into the design of collaborative systems. Below is a summary of the most pertinent issues.

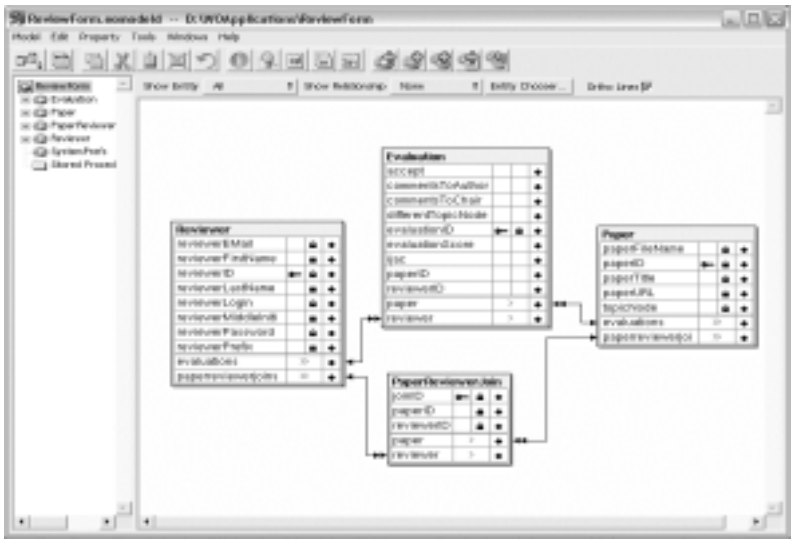

Figure 3. CAMEO database schema

Competition as Collaboration

What is unique about systems such as CAMEO-Review is that they reverse our traditional notion of collaborative systems. One can say that the common aim of potential authors, reviewers, and conference administrators is to create and share a body of knowledge. This common goal is achieved through a process of competition, selection, and finally confluence. Furthermore, a collaborative process – such as having the reviewers compare notes, discuss the papers and reach an agreement – is actually detrimental to the quality of the final product.

Hierarchy

The object model of CAMEO-Review clearly delineates a hierarchical structure which is often missing from collaborative systems. Authors have limited access to the system: they are allowed to register their entry, view the public comments of reviewers, and learn the final outcome of the review. Reviewers are allowed to read and evaluate the papers assigned to them and communicate privately with the Technical Chair. Finally, the Technical Chair has full access to the system, assigns papers to reviewers, and can read all comments and review all numerical scores.

Synchronicity vs. Asynchronicity

CAMEO-Review does not exercise the issue of synchronous vs. asynchronous collaboration – It is strictly asynchronous. However, CAMEO-Review does have to deal with time as a component of the system with submission deadlines and review periods in which the system is open to reviewers for logging in and retrieving papers for review.

Implementation

CAMEO-Review has been implemented for the ACADIA 2001 and 2003 conferences. In both instances, the review system aided the submission and selection of papers. Anecdotal evidence suggests that the system was viewed very positively by those who used it. Reviewers, in particular, enjoyed being able to download the PDF papers at their leisure and fill out the form for each paper when convenient. The system allowed them to log out and return to the system to complete their review.

Figure 4. Reviewers' view

2.2. ECAADE/FILEMAKER INTERFACE

This interface has three functions: 1) freely accessible abstract submission, 2) password-secured blind abstract jurying and 3) maintenance and monitoring facilities for the managers. FileMaker Pro is a database application that includes a so-called ”web-companion” plugin. The eCAADe’s web-interface with the FileMaker’s database files is rather simple, consisting usually of four html-pages per each database file: 1) listing of all database records, 2) creation of a new database record, 3) modification of an existing database record and 4) deletion of a database record.

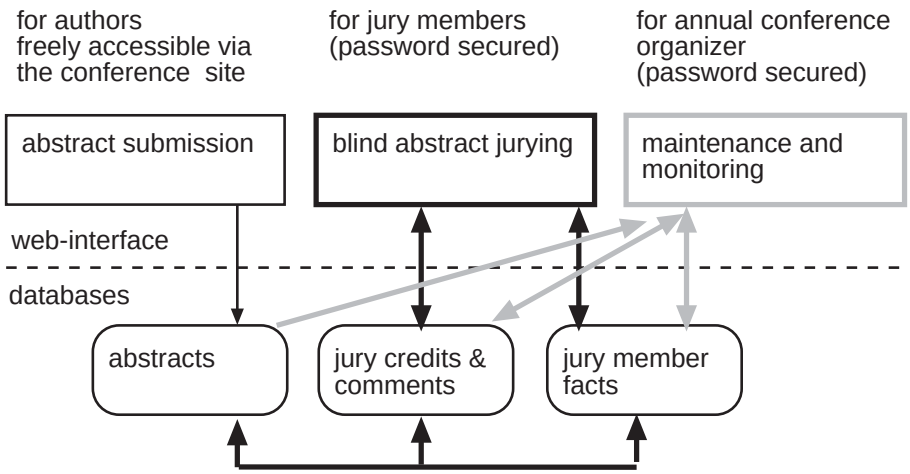

Figure 5. Architecture of eCAADe's Interface

eCAADe’s system was first introduced for the 2001 conference in Helsinki and it has since then been enhanced and fine-tuned on an annual basis for the Warsaw 2002, Graz 2003 and Copenhagen 2004 conferences (http://www.mittaviiva.fi/databases.html). The gradual enhancements made to the original system during these years have been:

- blind-jurying (added 2002)
- security (enhanced 2002)
- automatic creation of email notifications (added 2003)
- keywords and personal interest areas (added 2004)

System structure and functions

The core of the database portal (fig. 6) is a database with the submitted abstracts, which stores all entered data including author facts and date & time stamps (150...180 abstracts annually). In case of a successful submission the author immediately receives a confirmation receipt. In case

of missing fields, the author is requested to go back, fill the missing info, and submit the paper again.

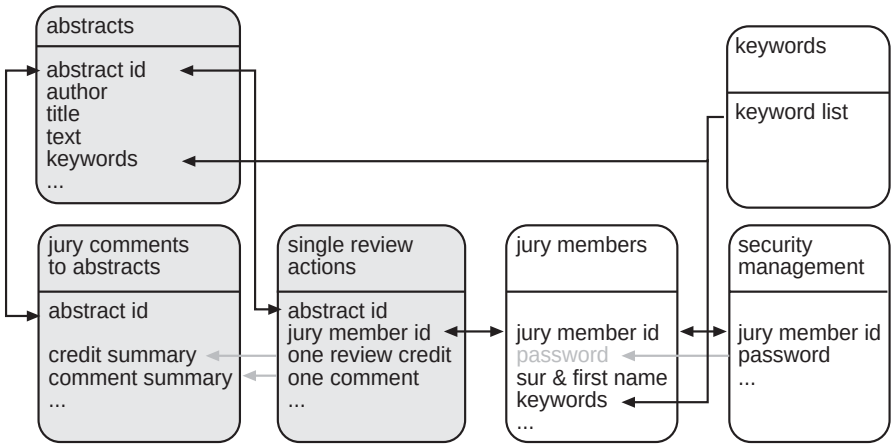

Figure 6. Structural components

The eCAADe-system, as it currently exists, is designed for the submission of single abstracts and, as the stored data can only be checked by means of a login, it is inevitable that a small number of submissions are duplicated. This is particularly true as some of those submitting have a firm belief in the idea that ”multiple pushing the final submission button proves meaningful.” Sorting and filtering the submitted papers is easily done by the conference organizer, and in most cases the act of duplication is obvious.

Recognition facilities can, and should be built in the future, so that in case of a subsequent submission, an alert will be sent out ("This entry is already recorded"). In the eCAADe solution a personal login is allowed only for the reviewers; submitters can not return to their submissions afterwards.

Jury view

All the jury comments on the submitted abstracts are collected during the month-long reviewing process. For security reasons the comments are stored separately from the original abstract data. The abstracts are assigned to jury members (35...50 persons world-wide) by the conference organizer, who manually checks that the jury member does not have a close connection to the authors. A jury member is naturally allowed to read only his/her abstracts and without author facts to ensure the blind jurying. In the eCAADe review process each abstract has 3 reviewers resulting in approximately 10 papers assigned to each jury member. Jury members are also allowed to update their personal data, address, email etc., as well as their personal expertise areas with keywords. Those same keywords are used when authors submit their

original abstracts. By using keywords eCAADe is trying to develop a better match between the submitted abstracts and the expertise and interests of the jury members.

Figure 7. Jury view

A jury member can read and rate single abstracts with a simple web-form. The field "comments to the authors" has been made obligatory in 2004 and this helps the conference organizer to deliver proper feedback to the authors. This way we enable young researchers to learn from the more experienced jury members.

Conference organizers view

The third essential file in the database portal, is the so-called "review actions database", where single jury members' reviews of single abstracts are stored. (fig. 6). All review actions are finally collected together for calculation aims. The annual conference organizer has a wide access to all these files, to monitor for instance:

- the number of submitted abstracts.
- the progress of the jury members' evaluation task.
- the final rating of each abstract

Figure 8. Monitoring view

Though just a simple list, with sorting features; the organizer gets an immediate image of the jurying process and abstract reviewing status.

Some of the conference organizer activities are easier done ”off-line” rather than via the web. Assigning all the submitted papers to the jury members is seemingly easier and quicker to do with Excel rather than with the web-interface. The final selection of the accepted eCAADe-papers is preferable with printed paper copies. The ”traditional method” also allows the organizer to divide the papers into conference sessions at the same time. The email notifications are sent to authors after the review procedure is finished and this can be done quickly with FileMaker’s ”send email” utility. The conference organizer could also easily ”order” some eCAADe-base data in raw text format remotely via the web, but in practice it has proven easier and quicker to do so manually – the on-site databases have to have a local manager anyway to ensure that the system & machinery is running correctly.

Predecessors and spin-offs in architectural education

The eCAADe-interface is based on several smaller web-applications which were developed for the schools of architecture at Tampere and Helsinki University of Technology during the period 1996–2003. The main application areas have been serving as course communication (http://www.arkit.net): for example signing up for exercises and courses, course schedules, lecture materials, topic-oriented link-collections, weblogs and different search and help-engines.

2.3. SCIX/WODA-INTERFACE

The system was developed in the framework of the SciX project (www.scix.net) to support organizers of scientific conferences with following activities: 1) registration of participants with extended profiling, 2) multi stage on-line submission (handling of abstracts and/or full texts), 3) automation of review assignment process, 4) review of submissions according to a predefined set of criteria with an adjustable voting mechanism and 5) notification of involved actors about assigned tasks, forthcoming deadlines and events. The solution is based on a modular architecture and uses the WODA-engine (http://www.ddatabase.com/). Depending on the specific characteristics of a certain series of conferences, a tailored customization is feasible. The interface environment also includes electronic publication tools to handle the submission of full papers and a following review. For example, the "book of abstracts" can be published directly from the database. A simple version of solution was used already in 2001 (http://www.ikpir.com/gi2001/) for a national conference and went through a process of continuing improvements for the ECPPM 2002 conference (2002.ecppm.org) and IAPS 2004 conference (iaps2004.scix.net).

Technical details

The solution is based on WODA - Web-Oriented DAtabase (Turk, 1999) – that was conceived as a library of CGI functions written in Perl. Since the library is free, one only needs a web server that can serve Perl scripts. Based on the data definition set up by the owner of the database, and parameters passed as a part of the URL, WODA automatically generates appropriate Web pages that are used for adding, updating, searching, reviewing records as well as for administration. Typically one would create one definition (Perl script that calls WODA library) per typical object (i.e. user or submission). These definitions represent tables, which can be relationally linked. The engine itself offers a broad range of facilities, such as full -featured natural language search, category based browsing, and simple email based agent for automatic notification that operates using data in any of the available tables. The quality of application depends on appropriate utilization and customization of features, which may – with additional knowledge of Perl – exceed expectations. Since WODA embodies agent based technologies two communication channels are used between authors and organizers: (1) web based application, and (2) email.

Submission and review processes

Submission and review processes are supported by appropriate functionalities through the web-based application, which is explained and illustrated in more detail. A Web-based solution acts as a communication

channel between authors and organizers/editors. Any potential user must register (create account) to gain access to the system, which allows him/her to add/modify submissions and get feedback in timely fashion. Administrator or main editor may promote any user to reviewer, which requires special measures such as anonymity. The availability of information stored in the database, depends on permissions set for particular type of users that can be: author, reviewer, conference editor/administrator, or conference secretaries. Permissions maybe set at several levels: 1) tables, 2) records, and 3) fields, and 4) web-page views level (allowed parameters and paths). These permissions can be changed during publishing life cycle and in general they can be changed – according to the type of review process that was selected by particular organizing commitee. For example, in the framework of the iaps2004 conference the blind review of abstracts and full texts was selected.

Figure 9. System user image

Authors / reviewers

Registration procedure allows users to enter personal details and select his/her username and password, as well as additional information maybe required. For example, in the case of iasp2004.scix.net, three to five (out of a list of 20) predefined keyword areas describing area of expertise had to be selected. These keywords were later used in order to facilitate the allocation of submissions to suitable reviewers. Through a personal login user may use the system as a participant only, an author or maybe promoted to a reviewer. Options, available to an user, are given through simple list of links (fig. 9).

The user can submit an abstract, may re-examine submitted data, as well as observe the status of an individual submission. If the abstract is accepted, a user can upload the following full-text version to the server for further review. Since the author may be at the same time a reviewer he has access to the interface for reviewers and do the work on his/her assigned reviews. These options are available if the organizers promote author to a reviewer and if one has assigned reviews. In order to reduce unproductive communication between authors and organizers, the interface is quipped with facilities like “forgot your passwords”, or “formatting errors”.

Figure 10. Selection of reviewers based on area of expertise and workload

Organizers / editors

After the submission deadline has passed, reviewers must be assigned to the individual entries. However, this may cause difficulties, as specific areas of expertise have to be matched both on the side of the submitters as well as the reviewers. The solution is based on an efficient key wording system. Main requirements for automatic assignment procedure were: 1) equally distribute the work load among reviewers, 2) “automatically find” the most appropriate reviewers for particular papers; 3) quickly depict status of different review assignments; and 4) automatically manage notifications. Figure 10 illustrates an automatic review assignment form with a suggested – filtered list of reviewers for a particular paper.

Subsequently, the use of the web-based interface supports finalization of decisions and notification within condensed timelines. In cases where this type of work is not database-driven, a high potential for mistakes exists. Recognition of double entries etc. must be regarded as important in this context. The use of semi-automated classification techniques has also been implemented to allow clustering of entries with similar topics into coherent sessions. Important advantage of WODA is that it keeps internal Audit Trail that allows editors to recall status of particular submission/user at certain point of time. A full list of the editorial options is shown in figure 11.

Figure 11. Organizers options

3. Conclusions

The World-Wide Web has become the de-facto platform for collaboration. In contrast to earlier collaborative systems that created their own proprietary collaboration protocols, current systems rely on the familiarity and ubiquity of web browsers and the standardization of HTTP and IP protocols to ensure the acceptance of their software. In general, even simple and low-cost Web-based solutions have proven successful and easy to use in terms of flexible handling of the information and by this providing content for special interest groups.

The elaborated platforms reverse a common feature of collaborative systems. As we commonly see, online collaborative systems rely on social awareness to enable interaction. Chat rooms (both textual and multi-dimensional) are prime examples where knowing who else is in the room is crucial to the collaborative experience. Scientific standards, however, require that the system is blind: 1) The authors do not know who else is submitting papers, 2) the authors do not know who will review their papers, 3) the reviewers do not know the identity of the authors 4) the reviewers do not know who else is reviewing the same papers they are reviewing and 5) the authors are not aware of candid comments regarding their paper that the reviewers send to the technical chair. While traditional collaborative systems rely on social awareness to enable interaction, submission and review platforms as presented in this paper depend on social *unawareness* to accomplish their goal. Any breakdown in this unawareness not only hinders this goal, but renders it invalid.

Possible re-use and adaptation of the presented interfaces for web-based data collection and handling interfaces in terms of a "modular architecture" is obviously of interest. In principal the use of the presented interfaces is not limited to the organization of conferences. One could think about rentable database-infrastructures for data environments - which are customized for the execution of specific working tasks. Programming knowledge is not a prerequisite for the final end-user.

References

Jabi, W.: 2003, Reflections on Computer-Supported Cooperative Design Systems, *CAAD Futures 2003 Proceedings*, Kluwer Academic Publishers, Netherlands, pp. 169-180

Johnson, B. and Kolarevic, B.: 1999, EVAL: A Web-based Design Review System, Media and Design Process, *ACADIA '99 Proceedings* [Salt Lake City], pp. 30-39

Turk, Z.: 1999, WODA: A Slim Web Oriented Database, Heterogeneous and Internet Databases, City University of Hong Kong Press, Hong Kong, pp. 95-109