# Architecture, Implementation, and Evaluation of ConfMan:
## Integrated WWW and DBS Support for Conference Organization

**Pål Halvorsen, Ketil Lund, Vera Goebel, and Thomas Plagemann**
University of Oslo, Center for Technology at Kjeller (UniK)
{paalh, ketillu, goebel, plageman}@unik.no

**Thomas Preuß[‡], Hartmut Koenig**
Brandenburg University of Technology at Cottbus
Department of Computer Science
{tpreuss, koenig}@informatik.tu-cottbus.de

[‡]Now at PricewaterhouseCoopers (PwC)
Thomas.Preuss@de.pwcglobal.com

**ABSTRACT:** In the context of the organization of DAIS'97 and IDMS'98, we have developed an integrated support system for conference organization based on World-Wide Web (WWW) and database system (DBS) technology. This paper describes design, implementation, and evaluation of the organization system *ConfMan*. We discuss two important contributions: (1) the systems-oriented aspects, i.e., design, implementation, and performance evaluation of an advanced distributed application that combines DBS and internet technology; (2) human-factor-related aspects, i.e., general usability of the system. The discussions in this paper reflect our experiences of organizing DAIS'97 and IDMS'98, and should help other conference organizers to reduce their workload.

**Keywords:** Web/database system interactions, performance evaluation, support for collaborative work, conference organization support, on-line PC meeting

## 1 Introduction and Motivation

National and international workshops, conferences and symposia[1] are an important means to exchange ideas, discuss approaches, meet colleagues, etc. from research and development communities. For the organizers of such an event, it is an honor to be responsible, but it comprises also a lot of work. Various applications can be used to handle the large amount of data and to support communication between organizers, authors, participants, etc. For example, internet-based applications, like e-mail, File Transfer Protocol (FTP), or World-Wide Web (WWW), are used to distribute the Call-for-Paper (CfP), collect contributions, distribute papers for reviewing to Program

1. We use in the remainder of this paper the term conference as a synonym for workshops, symposia, and conferences.

Committee (PC) members, collect the review results, send out notifications to the authors, announce the program (Call-for-Participation), and registration of participants. The advantages compared to traditional solutions like snail-mail or courier services are obvious: internet applications deliver information much faster, all information is available in a certain digital format, and at least so far, internet is much cheaper. Furthermore, the digital information can be used for further processing in other applications. For example, information about authors, reviewers[2], and participants are often handled in the conference secretariat with an office application like Excel or Filemaker. Unfortunately, there are a lot of incompatibility problems related to different representations in different applications. Therefore, a lot of work still remains to be done by "hand" or by some ad hoc written scripts, e.g., Perl-Scripts or Shell-Scripts. However, this type of processing is known to be error-prone.

One insufficient aspect of today's solutions for conference organization is the missing integration and integrity control. The second main disadvantage is that PC members have to travel if they want to take part actively in the final selection process. Traditional system support like telephone conferences or video conferences is only helpful for a few discussion partners. However, PCs comprise often more than 15 members. Therefore, in many cases only PC members who live not far from the conference location participate in the final meeting and determine the conference program. This initiated the idea to organize the PC meeting for DAIS'97 via WWW and resulted in the implementation of a major core element of *ConfMan* (Conference Manager), called *OBSession*. It is based on the public-domain relational database system (DBS) *Mini SQL* (mSQL) [5]. Review results, comments, and discussion arguments are stored in the DBS and can be retrieved during the discussion via WWW. The system was successfully applied for one of the first worldwide PC meetings via the WWW. 15 PC members from ten countries on four continents participated in that meeting. Another element of ConfMan is a WWW-based paper submission system which notifies automatically every paper submission by e-mail. ConfMan, first used for IDMS´98, integrates both elements to avoid "manual" transfer of data from one system to another and additionally supports further organization tasks, like handling registration of participants. ConfMan uses a WWW-based user interface and for certain tasks e-mail. For data management, mSQL is used to handle all organizational aspects starting from paper submission process, via review process and PC meeting, to author notification and registration of participants.

This paper describes design, implementation, and evaluation of ConfMan. We discuss two important contributions: first, the systems-oriented aspects, i.e., design, implementation, and performance evaluation of an advanced distributed application that combines DBS and internet technology. Second, we discuss human-factor-related aspects, i.e., general usability of the system. The reported experiences with using ConfMan for various conferences should be helpful for future conference organizations.

The remainder of this paper is structured as follows: we analyze the typical roles and tasks of entities during conference organization in Section 2. In Section 3, we describe the design and implementation of ConfMan. The functionality of ConfMan is described in Section 4. In Section 5, we evaluate ConfMan with respect to performance, reliability, security, and usability. Finally, we give an overview of related work in Section 6, and summarize the main contributions and give an outlook on future work in Section 7.

2. We will use the terms *PC member* and *reviewer* interchangeable.

## 2 Analysis of Roles and Tasks During Different Organization Phases

In order to find an appropriate design for a conference organization tool, it is important to have a good understanding about the following three aspects of conference organization: (1) the dynamics, i.e., what happens when; (2) which person/entity is involved, i.e., identify the different roles; (3) the task of these roles, i.e., who is doing what. It is often difficult to clearly separate the groups of involved persons and organization phases. Organization phases are often overlapping and certain persons might act in multiple roles. We distinguish between the following four roles that are involved in conference organization:

- **Organizers**: include general chairs, program chairs, publicity chairs, treasurer, local arrangement group, etc.
- **PC members**: persons who serve as reviewers and help to set up the technical program of the conference, also often involved as session chairs, etc. during the conference.
- **Authors**: all persons that submit a paper to the conference.
- **Participants**: all persons registered for the conference.

The relationships between these roles are illustrated in Figure 1, and their different tasks are discussed in the subsequent description of the different phases of conference organization.

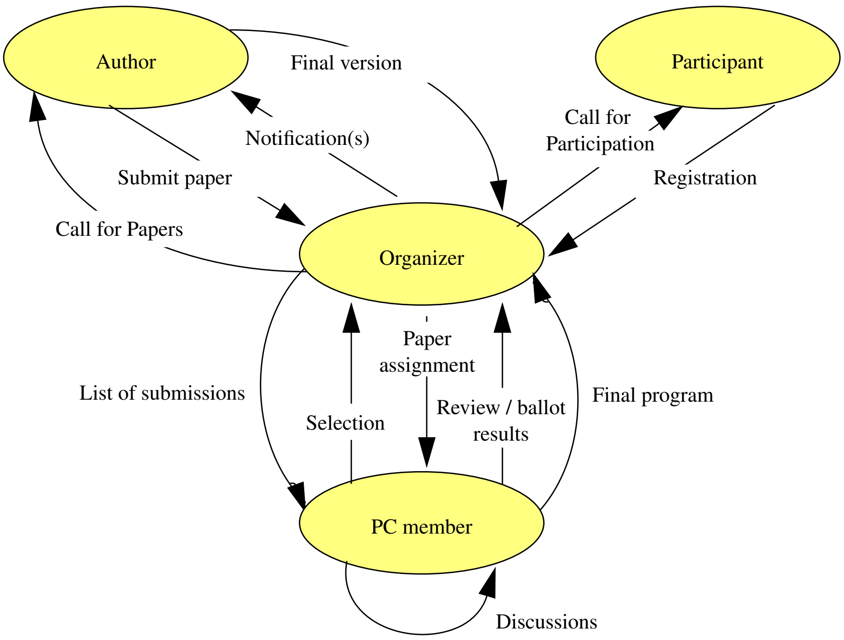


**Figure 1:** Roles and tasks in conference organization.

Generally, we can identify the following phases during conference organization. Figure 2 illustrates these phases using IDMS'98 as an example.

- **Preparation**: In this phase, many basic decisions for the conference have to be made. Typical activities during this phase are the selection of: (1) PC, (2) conference date, (3) conference venue, (4) publisher for the proceedings, (5) supporting institutions, e.g., IEEE or ACM, and (6) sponsors. All these aspects result in the first "product" of the organization team: the CfP, which is distributed to the corresponding community.

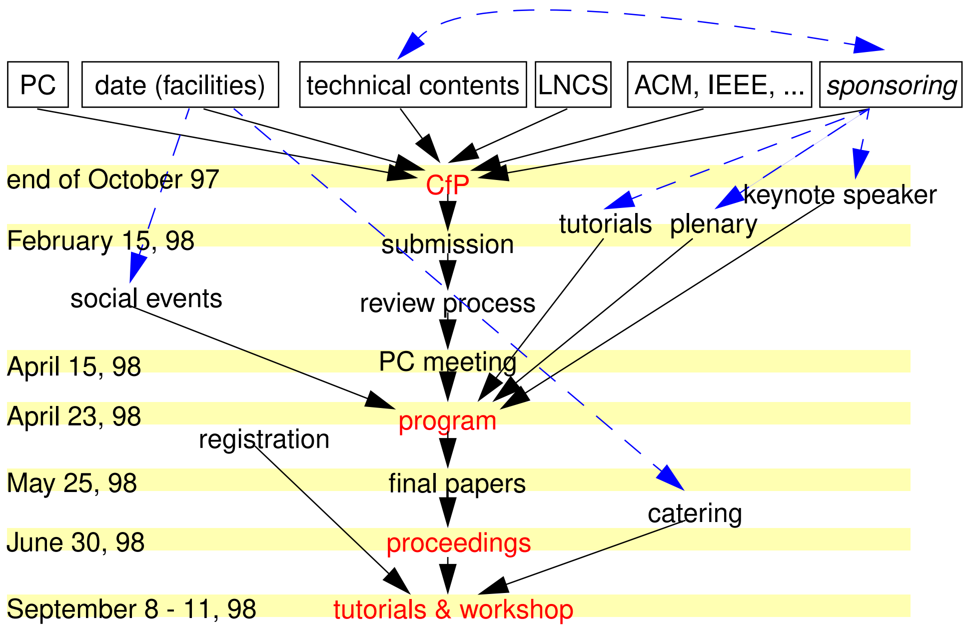


**Figure 2:** Phases of preparing IDMS'98.

- **Paper submission**: the CfP invites authors to submit papers. The organizers have to acknowledge the receipt of each paper and to put them into an archive. The results of this phase are a list of submitted papers and the papers themselves (today mostly in electronic form).

- **Paper reviewing**: there are three steps during this phase: (1) Papers have to be assigned to the PC members for reviewing. Often, PC members receive the list of all submitted papers and may choose their favorite selection. (2) The organizers send the assigned papers to the PC members together with a review schema for each paper. (3) The organizers collect all reviews, put them into an archive and keep track of the relationship paper-review-PC member. However, this relationship must not be shown to other persons than those belonging to the organization committee (confidential information). Finally, all reviews are "merged", and all submissions are sorted according to certain criteria, e.g., average overall rating.

- **PC meeting**: The efficiency of the PC meeting is heavily depending on the result of the review phase. During the meeting, PC members discuss which papers should be accepted or rejected. Generally, there are some papers for which all reviewers agree that they should be accepted or rejected. Papers with contradicting reviews and those which lie in between clear accept and clear reject have to be discussed most. Additional reviews might be necessary for these unclear papers. Thus, the result of the review process should facilitate the classification of the submitted papers. Furthermore, all review comments etc. should be easily accessible during the PC meeting. The result of the PC meeting is a list of accepted and rejected papers. The authors are notified and receive all ratings and review comments - except confidential comments for the PC.

- **Proceedings preparation**: Authors of accepted papers prepare the final version of their papers according to the review comments and format guidelines. The organizers collect all the papers and prepare the proceeding, generally in cooperation with the publisher. In parallel, the organizers elaborate all details of the conference program and announce it (Call-for-Participation).

- **Registration**: Participants register for the conference. The organizers have to acknowledge the registration and received payment for every participant. Often organizers assist participants to find accommodation for the time of the conference, e.g., with block reservations and special prices.
- **During conference**: Arriving participants register at the registration desk and receive the proceeding, some buy extra proceedings or lunch tickets, and receipts are printed, etc. The organizers must be able to control the details of each registration. Furthermore, a list of participants is created and distributed.
- **After conference**: A summary, book-keeping, and reports for supporting institutions like IEEE or ACM have to be written (based on statistics of the conference).

ConfMan fully supports all phases, except “preparation” and “proceeding preparation”, which is not supported, and “after conference”, which is only partially supported.

## 3 Design and Implementation of ConfMan

The main challenges and goals for design and implementation of ConfMan are:

- All major organization tasks should be supported by one integrated system.
- The usability of the system, including its performance, should be good enough to attract the different user groups to use the system.
- The system should take care of data integrity.
- The system should use a flexible protection mechanism that supports security and openness at the same time. For example, CfP, conference program, and on-line registration have to be accessible for everybody. Contrary, submitted papers, review comments, etc., must only be accessible for authorized users, like PC members, which are defined by the organizers.

Two important building blocks of the ConfMan architecture are indirectly defined by these goals: (1) all data has to be stored, retrieved, updated, and deleted via a DBS, thus, data integrity will be automatically assured from the system, and (2) all interactions will be performed via WWW (and by e-mail for a few particular tasks), thus, world-wide access to conference information is easily achieved by setting up a WWW server.

### 3.1 Database System

All information provided by the different user groups, like contributors, participants, and PC members, must be persistently stored. This can be done be using a file system, a DBS, or an office application like Filemaker. Filemaker provides much more functionality than simple file systems, but does not support DBS concepts, like abstract data model with query language, and transaction management. To allow concurrent access, ensure consistency, and easy retrieval, we have chosen a relational DBS to manage the conference information. Since ConfMan relies on a WWW based user interface, it is important to have an easy integration of DBS and WWW. Another requirement for ConfMan is to use public domain software wherever possible. Therefore, we use the lightweight relational DBS Mini SQL (mSQL) [5] in ConfMan. The mSQL system consists of a database server, and various tools for communication with the server. It is made for high data access performance on “small” hardware platforms, such as PC’s. The system is distributed as source code, and it can be compiled on a number of platforms [5]. For creation and manipulation of databases, the mSQL Query Language is used, which is a subset of the ANSI SQL standard. The queries are embedded in cgi-scripts (written in Lite - see Section 3.3).

### 3.2 Schema Design

To design the schema for storing all involved data in mSQL, we have first modeled a conceptual schema as Entity-Relationship (ER) diagram. Based on the ER design we have performed a mapping from the ER diagram to the relational schema according to the standard rules for relational schema design [3].

The data managed by ConfMan can be divided into two major parts: (1) information about papers and their reviews, and (2) information about the conference and the participants. The conceptual design of data about papers, reviewers, reviews, comments about papers, and votes is depicted as an ER diagram in Figure 3. A reviewer may review several papers, and a paper is reviewed by various reviewers. In addition, a reviewer may vote for all the papers (in both first and second ballot), and a paper may receive several votes. During the ballot, an on-line discussion is performed where reviewers may submit comments about a paper. When reviewers access the ballot page, papers with new comments are marked “hot” (see Figure 8).

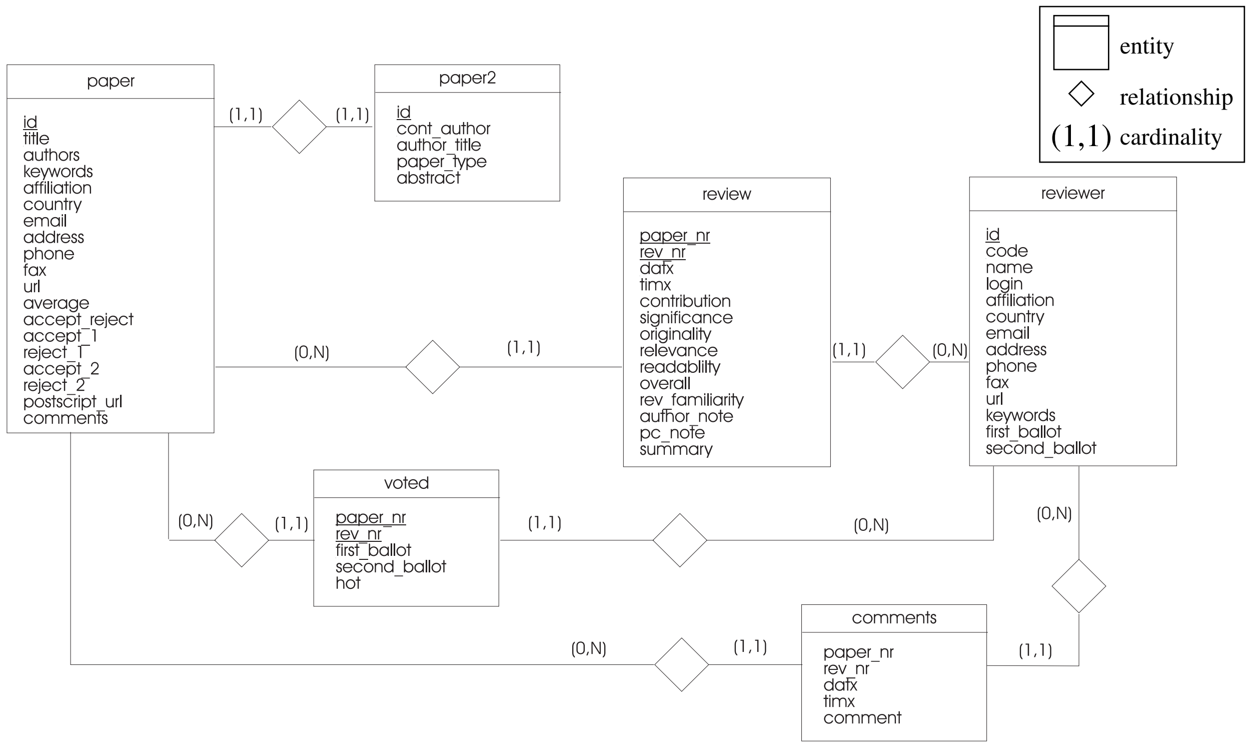


**Figure 3:** Conceptual schema shown as ER diagram (part I).

The conceptual design for conference data and participants is shown as an ER diagram in Figure 4. Information on conference events, e.g., tutorials and conference sessions, is stored in the event1 table. Information on “extras” such as extra proceedings and lunch tickets, i.e., quantifiable items, is stored in the event2 table. Each participant may register for several events.

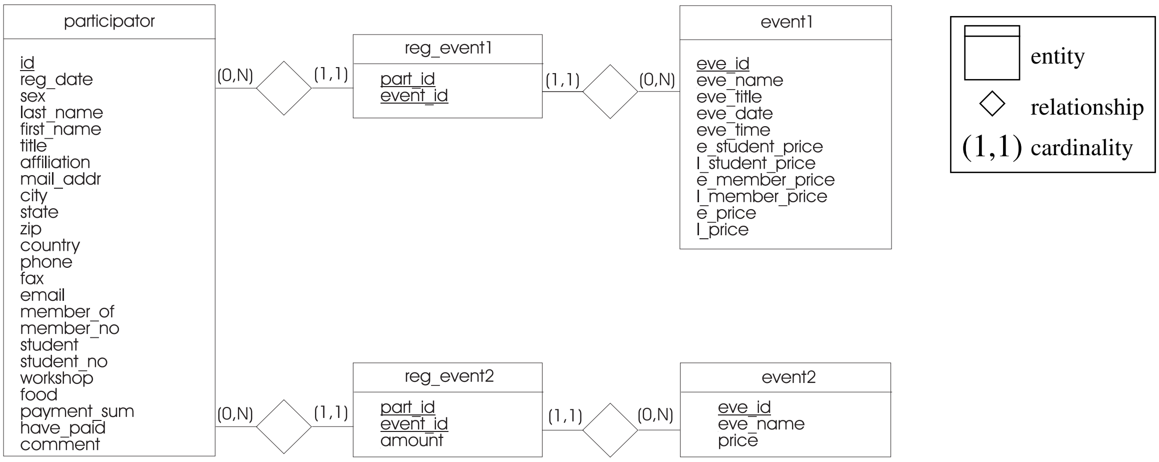


**Figure 4:** Conceptual schema shown as ER diagram (part II).

The most important tables of the database schema of ConfMan are shown in Figure 5. All papers, reviewers, and participants are assigned a unique ID, and in most of the tables, this IDs is also used as a primary key (shown in Figure 3 and Figure 4 as underlined attribute names).

| Table | Description |
|---|---|
| `paper` | Holds information about papers, its authors (with address, affiliation etc.), keywords, average result of the reviews, and number of votes. |
| `paper2` | A “patch” to ‘paper’ table. Holds additional information about papers, including abstracts. |
| `comments` | Contains comments of PC members for all submitted papers. Basis of on-line discussion. |
| `reviewer` | Information about PC members. |
| `review` | Stores ratings and comments from each review. |
| `voted` | Stores votes (accept/reject) of each reviewer on each paper. |
| `participator` | Information about participants name, address, etc. |
| `event1` | Information about conference or tutorial(s). |
| `event2` | Information about orders like extra proceedings, lunch tickets, etc. |
| `reg_event1` | Registration of which participants who have registered for the conference and/or a tutorial(s). |
| `reg_event2` | Registration of other orders like extra proceedings, etc. |

**Figure 5:** Main tables in the ConfMan database.

Finally, there is also a number of small tables used to keep track of information like current phase in the review/ballot, allowed paper types, paper keywords, mail commands and messages, and authorized administration personnel.

### 3.3 Protection Mechanisms

To protect the confidential data stored in the database, we use the HTTP login and password authentication mechanism of the WWW server. All scripts and pages of the paper reviewing and PC meeting phases together with the administration pages are protected. The ConfMan system has been used with two different WWW servers: CERN [10] and APACHE [1]. In both cases, the protection is provided through access control lists (ACLs).

In addition to the WWW server protection mechanism, the administration module is protected via the DBS. Scripts for administrative personnel have a hard-coded user authorization. For all accesses, a query is sent to the DBS checking the user login from the HTTP authentication against a list of trusted people. The user can proceed if the authentication query finds a match, otherwise an error message is displayed, and access is denied.

### 3.4 Lite Programs and cgi-bin Scripts

All accesses to the DBS are performed via cgi-bin scripts. These scripts are programmed in Lite, which is a complete, interpretative programming language with syntax and semantics very similar to C. To ease the coupling between the WWW and the DBS, Lite is integrated in a WWW-to-mSQL interface package (W3-mSQL) [5]. Lite scripts are used to access the mSQL DBS and retrieve data from the database. They are also used to generate HTML documents and present the retrieved data on a WWW page. The basic procedure to generate HTML documents is as follows: (1) when a client activates a Lite script, the script connects to the mSQL server via either UNIX interprocess communication (sockets) or a TCP connection, (2) the database is opened, and the access operations (queries) are executed, (3) an HTML document presenting the results is generated by the Lite-script and sent back to the client through the WWW server, and (4) the connection to the DBS is closed.

### 3.5 Architecture

The architecture of ConfMan is shown in Figure 6. The server comprises four layers:

- **mSQL** DBS represents the lowest layer.
- **W3-mSQL** layer interfaces mSQL to WWW and contains an interpreter for Lite scripts. The W3-mSQL communicates with the mSQL server via TCP or UNIX sockets.
- **Lite scripts and HTML pages** represent the core of ConfMan.
- **WWW server** at the highest layer enables (restricted) world-wide access to the Lite-scripts and HTML pages of ConfMan via the HTTP protocol.

The set of Lite scripts and HTML pages is divided into three parts. Two parts are protected: one for PC members and one for administrative personnel. The open unprotected part mainly contains functionality for paper submission and participant registration. The scripts and HTML pages of these three parts implement four functional modules of ConfMan which are described in the following section.

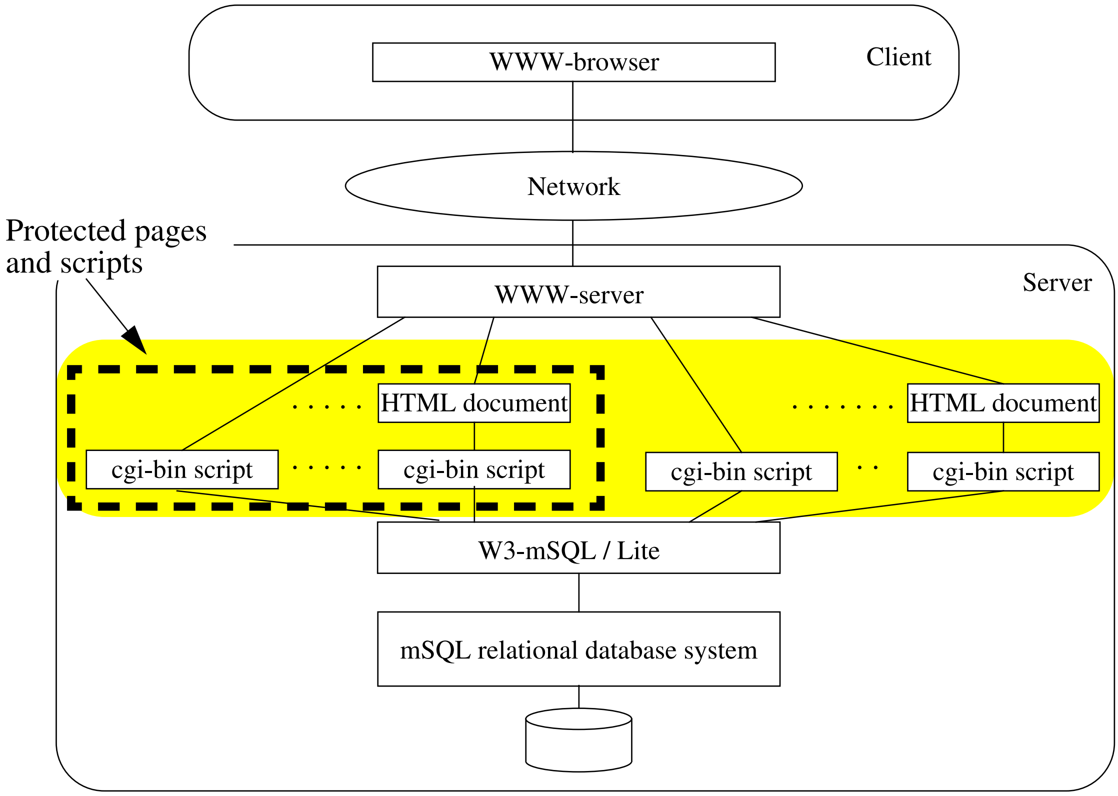


**Figure 6:** System architecture of ConfMan

## 4 Functionality of ConfMan

The functionality of ConfMan can be divided into four main modules. Each module corresponds to one of the four roles discussed in Section 2 (see Figure 1), paper registration, PC on-line meeting, registration of participants, and administration.

### 4.1 Paper Registration

After the CfP is sent out, papers will be submitted to the conference. This is handled by the paper registration module which allows an author of a paper to register the paper via WWW. The author has to fill out a form including contact author information (name, address, phone, e-mail, etc.) and paper information (authors, title, paper type, keywords and abstract) on a WWW page. This information is sent to the conference management system and persistently stored in the mSQL DBS. When receiving of this information, the system automatically generates an e-mail notifying the contact author what information has been stored and how to hand in the paper via FTP or e-mail. Even if submission of the paper itself fails, all data about author and contribution is stored and can be used later to identify a missing paper.

### 4.2 PC On-line Meeting

When the deadline for paper submission is passed, the system is used to generate a list of all papers, containing authors[3], title, and abstract. This list is sent to all PC members to select which papers they want to review. Next, the assignment of papers to PC members is entered into the database via a corresponding WWW page. Afterwards, the system sends a list of assigned papers to each of the PC members.

After receiving all reviews, the PC on-line meeting can be set up to determine which papers are accepted and which not. The meeting is managed by the "PC on-line meeting" module of Confman. Instead of sending a lot of e-mails, travelling to PC meetings etc., all the work can be done via WWW. A very important element is the policy applied by the conference organizers. The experience shows that an exactly planned procedure is very important for the success of the on-line meeting. For DAIS'97 and IDMS'98, a two-phase procedure with a duration of one week was applied (see Figure 7). The meeting started with an initial ranking of all papers according to the average overall recommendation. Then the papers could be discussed. To simplify the procedure, the program organizers proposed to accept all papers with an average ranking higher 7.0 and to reject all papers with a ranking below 4.0. This was usually accepted after a short discussion. After two days the first ballot took place and the new ranking was published. The number of accepted and rejected papers could be increased, so that the discussion could focus on the papers around the acceptance line. The final ballot was made two days later. One or two days later the final program could be published. In some (very few) cases were an agreement could not be found during the second discussion phase the final decision was mad by the conference organizers.

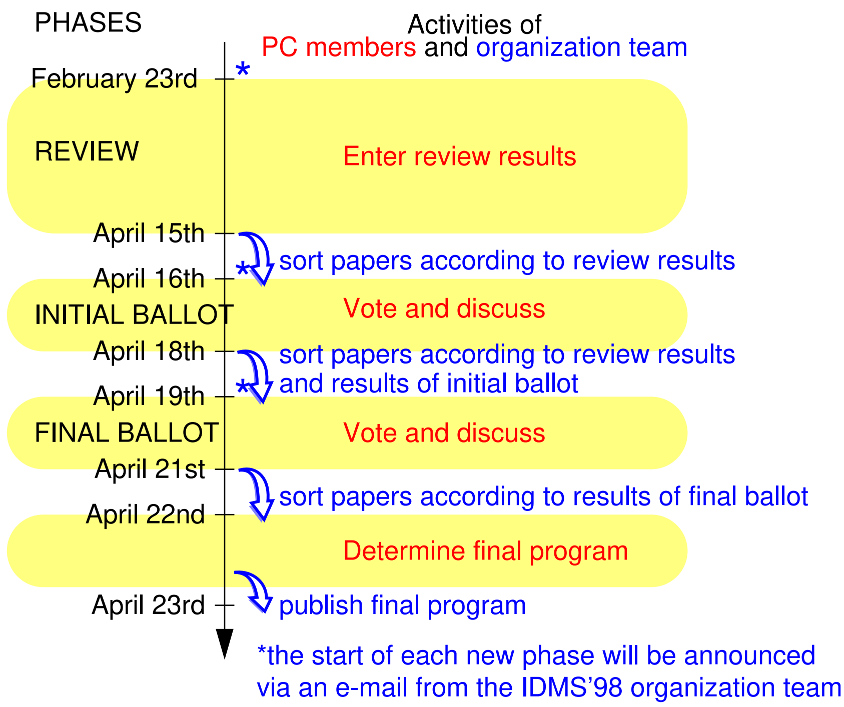


**Figure 7:** Phases of PC work taken from IDMS'98.

3. The authors name, affiliation, and country may also be left out of all e-mails sent to and HTML pages shown to PC members, e.g., in case of double blind reviews like in the ACM SIGCOMM conference.

For security and confidentiality reasons this module is login and password protected, so each reviewer is assigned a login and password. When a reviewer accesses the WWW pages in the review phase, a personalized WWW page with the list of all papers that are assigned to this reviewer is displayed. The reviewers may download the papers, or proceed to register the review results, i.e., ratings and comments, via a WWW reviewing form. The review information is stored in the mSQL DBS.

In both ballot phases, the PC members may see the review information of the papers as shown in Figure 8. ConfMan presents all review results, including detailed and average ratings, comments to the authors, and the confidential comments for the other PC members. All information is retrieved on-line from the database via a script, and combined in a WWW page that is generated on the fly. Furthermore, PC members can participate in on-line discussions about papers and vote accept or reject for a given paper. All comments and votes are stored directly in the database. Once a PC member has voted for a paper, the voting feature is disabled for this PC member, but the he or she may still participate in the discussions. When all the papers are either accepted or rejected through voting and discussions, the final program is determined. Furthermore, ConfMan indicates new or unread discussion arguments for each single paper by changing the color of the corresponding "comments" icon. This gives PC members a quick overview of currently discussed papers and relieves them from checking every paper in the list.

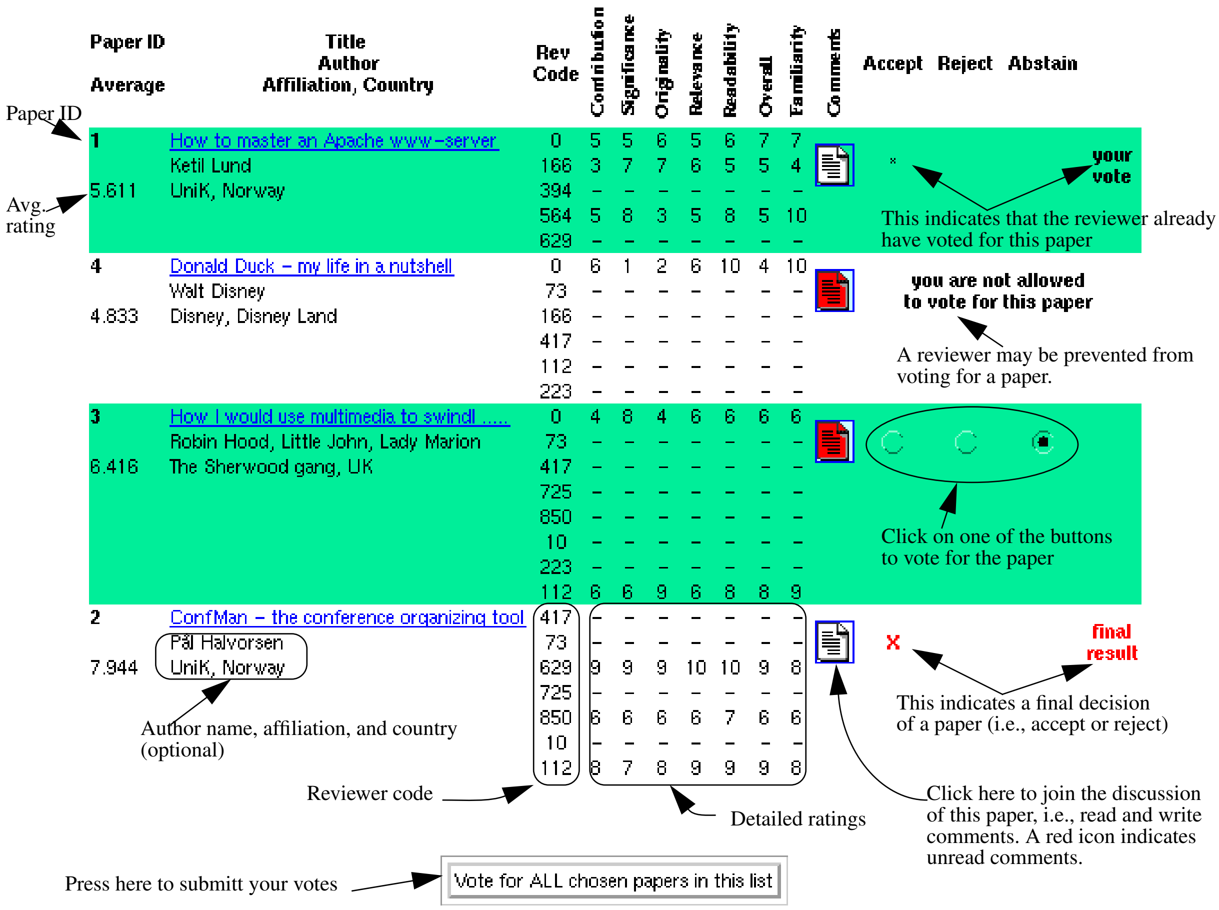


**Figure 8:** Example of an on-line ballot page.

In order to prevent PC members from looking at each others reviews during the review phase, the phases are separated from each other. This is necessary, since the reviews for each paper are displayed on the comments page. Consequently, it is not possible to enter the ballot/comments pages during the review phase. Phases are defined by a single protected variable, which can only be changed by the organizers. For each access to the ballot pages, the value of this variable is checked.

### 4.3 Registration of Participants

When the final program is ready, the Call-for-Participation is sent out and the "registration of participant"-module registers and stores all information about the conference attendees: Every person that wants to participate at the conference must fill out a registration form via WWW including personal information (name, address, phone, e-mail, etc.), conference information (special food requirements, extra banquet tickets, etc.), and tutorial information (which tutorial(s) to attend and possibly lunch). This information is sent to the mSQL DBS and managed by it. The registration information, including chosen events and costs, is displayed to the user on a WWW page and additionally sent by e-mail as a confirmation. The conference participant must follow a link to the payment form that includes all payment related registration information. This form should be filled out with details for the payment itself, e.g., credit card details, signed and then sent to the organizers (by fax or mail). If the participant needs to return to the payment page later, the URL is provided in the confirmation mail.

### 4.4 Administration

The administration module exclusively supports the organization committee. This module provides functionality to interact with other modules like sending e-mails (notifications, paper assignments, etc.), inserting and updating data in the mSQL DBS, and displaying status of the reviewing process. This module can only be accessed by members of the organization committee and is therefore protected via login, password, and an additional lookup in the DBS to verify whether the user is authorized to access these pages and scripts. We have divided the administration module into the following five parts:

- **Paper administration**: information about each paper may be displayed, e.g., authors, reviewing status and if the paper is accepted or rejected. The information may also be updated or deleted, and the generation and sending of e-mails for notification of received papers and of final acceptance is managed.
- **Review administration**: by clicking a particular link, the organizers can send (via e-mail) the list of all submitted papers to the PC members. Papers can be assigned to PC members for reviewing, and via mouse click, the PC members can be informed about this assignment, i.e., each PC member receives an e-mail containing the list of papers to be reviewed. Additionally, reviews may be inserted manually by the organizers, e.g., if a review arrives by e-mail (or mail) because a PC member does not have access to the WWW.
- **Reviewer administration**: includes a list of all PC members, indicating the papers each PC member has to review. Different colors show which papers have already been reviewed and which not. This feature is particularly helpful for program chairs to monitor the progress of all

PC members during the reviewing phase. It is also possible to prevent a PC member from voting on and participating in the discussion of a paper, e.g., because he/she is (co-)author of the paper. Furthermore, general information about a reviewer, e.g., address, affiliation, etc., may be changed.

- **Participant administration**: a summary of all participant registrations is displayed, i.e., how many participants have registered for the conference and for which tutorials, how much income will this mean for the conference, and how much of these fees have already been paid. Participant information can be changed, e.g., to indicate that the payment of the conference fee from a participant has been received. Receipts for the participants can be generated, and different reports can be run.
- **Miscellaneous part**: This section is divided into two parts: general functions and configuration functions. The general functions enable the organizers to change the phase in the PC online module, e.g., from "reviewing" to "first ballot". Moreover, personalized e-mails may be sent to all contact authors and/or PC members (the e-mail can also be sent to arbitrary persons by entering the e-mail addresses). Finally, a static HTML page can be generated, containing a list of all submitted papers, abstracts, and links to postscript or pdf versions of the papers. From this list, the reviewers can view the abstracts and download the papers. The configuration functions are used to tailor a ConfMan installation for a particular conference by defining conference and system specific information like conference title, e-mail addresses, e-mail command, texts for e-mail messages like acceptance notification, mail footers, paper types, keywords, conference/tutorial information, prices, early registration deadline, membership options, payment options, etc. All this information is included in a table in the database, instead of being "hard-coded" in cgi-bin scripts. Based on the information in this table, ConfMan generates conference specific paper and participant registration WWW pages.

## 5 Evaluation

In this section, we evaluate ConfMan with respect to response time, reliability, confidentiality, security, and usability.

### 5.1 Response Time

PC members of DAIS'97 and IDMS'98 mostly gave positive feedback concerning the response times of ConfMan. However, the response time of fetching the list of all papers including the review results (see Figure 8) during the ballot phase is often too long and represents the most crucial operation in ConfMan with respect to response time. Therefore, we concentrate the following analysis on this operation that joins three tables, updates one table, generates a WWW page that is often larger than 100 Kbytes, and transfers this WWW page to the browser.
The response time, denoted $t_{res}$, experienced by each user depends on several factors (see Figure 9): delay introduced by the browser ($t_{br}$), network delay ($t_{net}$), delay introduced by the WWW server ($t_{ws}$), and the response time of the mSQL and Lite package ($t_{DBS}$). In order to analyze $t_{res}$ and to break it down into its components, we have conducted experiments with a WWW browser and the

entire ConfMan system on a single 167MHz Sun UltraSPARC1, running Solaris 2.6. Our test data consisted of about 80 papers, 290 reviews and 12 reviewers. We measured the response time of getting the list of all papers and review results with the help of the software monitor *Quantify* [8] and by inserting probes (system call *gethrtime*) into the source code of the system.

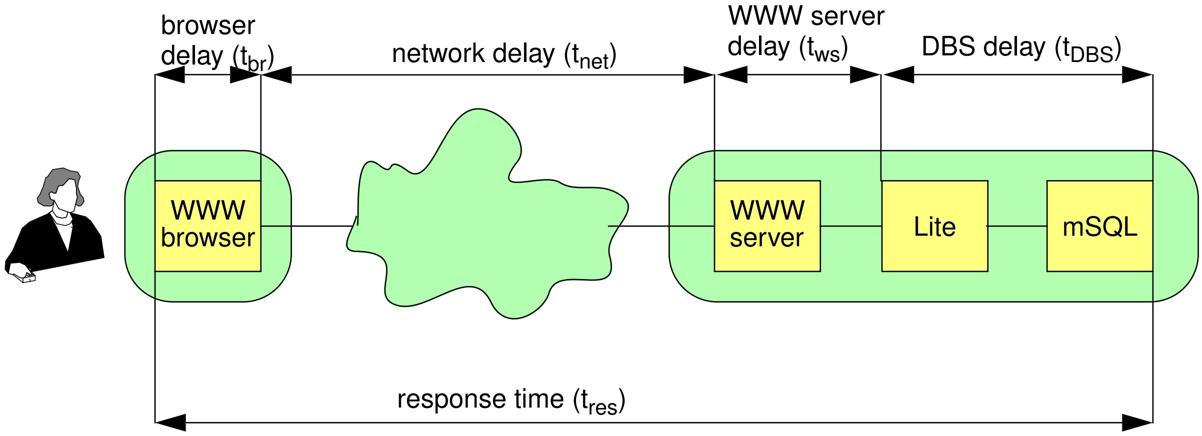


**Figure 9:** Components in response time

The average value of $t_{res}$ in this experiment is 30 seconds, a value which is far from ideal. However, neither the WWW browser nor the WWW server cause considerable delays; $t_{br}$ and $t_{ws}$ are each less than half a second. Furthermore, the network delay can be ignored in our experiment, because $t_{net}$ ranges in the order of milliseconds (ms). Consequently, the main bottleneck is the DBS delay. Our measurements show that more than 95% of $t_{res}$ is due to $t_{DBS}$. Further analysis show that the problem is located in the mSQL server. The execution of the Lite script ($t_{Lite}$) needs about 250 ms. The Lite script sends SQL queries via a TCP/IP connection to the mSQL server and receives the query results via the same connection. The data transfer between mSQL server and Lite-script ($t_{TCP}$) needs in average 60 ms. However, the aggregated response time of the mSQL server ($t_{mSQL}$) is more than 28 seconds. A more detailed analysis of the mSQL server showed that the bottleneck is located in the execution of the SELECT statements, i.e., the query processing itself. Reading data from disk is negligible compared to query processing.

In order to verify these results, respectively to improve ConfMan, we implemented a second version of the ballot script.[4] The new implementation takes advantage of the fact that much of the information shown in the ballot page is static. Only the comment icon and the voting part (the rightmost elements in Figure 8) may change during the ballot phases. In the new version, we considerably reduce the number of necessary SELECT statements by generating all static data only once when the phase is changed from review to first ballot. The static data is then stored - together with HTML-tags - in a separate table in the database, one record for each paper. This new implementation reduces the number of queries, but on the other hand, the amount of data read from the disk increases. The measurement results of this implementation show a much better performance, $t_{mSQL}$ is 10 seconds and all other factors remain the same. Thus, the entire response time is reduced

4. Another possibility to implement the ballot script is by using “ordinary” SQL-joins (called, server joins), instead of joins performed within Lite. However, experiments show that the performance of this alternative is even lower.

by approximately 20 seconds. Table 1 summarizes these measurement results.

| | $t_{br}$ | $t_{net}$ | $t_{WS}$ | $t_{Lite}$ | $t_{TCP}$ | $t_{mSQL}$ | $t_{res}$ |
|---|---|---|---|---|---|---|---|
| **Version 1: Running joins in LITE** | **< 1 s** | **<< 1 s** | **< 1 s** | **< 1 s** | **<< 1s** | **> 28 s** | **30 s** |
| **Version 2: Generating static data once** | **< 1 s** | **<< 1 s** | **< 1 s** | **< 1 s** | **<< 1s** | **> 8 s** | **10 s** |

**Table 1:** Comparison of the two ballot script versions

The impact of reducing the number of queries by generating once the static information is even more drastic with an increasing number of concurrent users. Figure 10 compares $t_{DBS}$ in both implementations with a number of concurrent users ranging from one to 32, using a Perl-script which forked off processes running the Lite-script. The figure clearly shows that the new implementation outperforms the original.

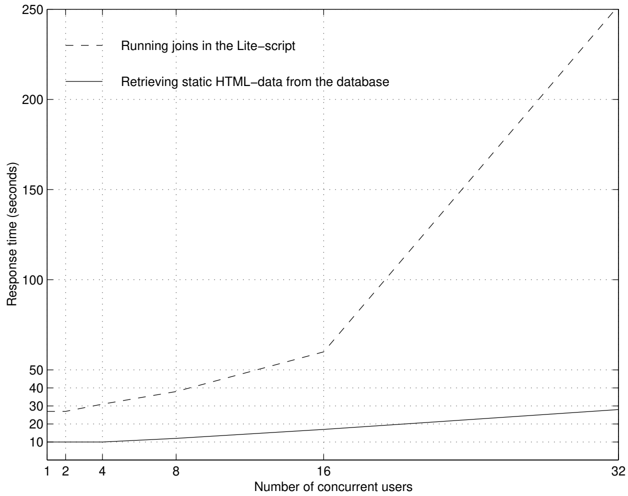


**Figure 10:** Comparison of the two ballot script versions with increasing number of users

Obviously, $t_{net}$ is not the bottleneck in our experiments, because WWW browser as well as ConfMan reside on the same workstation. In order to get an idea of the impact of $t_{net}$ in arbitrary scenarios, e.g., WWW browser in Asia and ConfMan in Europe, we determine the minimum amount of bandwidth that is needed to transfer the result of the ballot script to the WWW browser within the limits of $t_{DBS}$. In both versions of the ballot script, processing of queries and transfer of query results to the WWW browser is interleaved and concurrently performed. This means that the transfer of data is evenly distributed over $t_{res}$, and we can therefore estimate the minimum bandwidth requirements, i.e., the network is not the bottleneck and $t_{res}$ depends on $t_{DBS}$, as shown in (C1).

$$\text{Bandwidth requirements} \geq \frac{\text{Amount of data}}{t_{DBS}} \tag{C1}$$

Using (C1) we find that the original implementation leads to a transfer rate of approximately 5 KByte/s, while the new implementation gives a transfer rate of approximately 8.5 KByte/s on our server. Consequently, only in the cases where the client has a very low-bandwidth connection, e.g., using a modem, the network will potentially be a bottleneck.

## 5.2 Reliability

Since all information is stored in a DBS, data integrity is to a large extent guaranteed. However, to be able to recover from disk crashes and other catastrophic events, it is important to take backup regularly. For IDMS'98 we chose a rather pragmatic approach, having a crontab job performing a database dump every fourth hour, and storing the dumps on three different disks in a round-robin fashion. This approach was chosen since mSQL is able to do a database dump without having to take the system off-line.

In addition, all accesses to the IDMS'98 WWW pages were logged by the WWW server, and since the transfer of data from client to server is based on GET, the data could be reconstructed from the log.

The reliability of the system is dependent not only on safe data storage, but on the reliability of every component between the client and the server. Not all of these components can be controlled. There is, for instance, no way of guaranteeing a reliable internet service. We did experience a couple of incidents of connection problems. Further, we also experienced that the WWW server or the mSQL server stopped (due to unknown reasons), and had to be restarted. However, this was not a big problem. On the other hand, it shows that it is of vital importance to monitor the system.

## 5.3 Confidentiality

ConfMan assures a confidential reviewing. The reviewers remain anonymous during the whole reviewing process. They are only known to the conference organizers. In the ballot phases, during which all reviews can be read by the PC-members, each reviewer is identified by a code, and not by name. Additionally, it is possible to conceal the names of the authors from the PC-members.

During the review phase, it is common that PC members delegate the reviewing task to subordinates. This means giving away login and password to the persons in question. To ensure that only the PC member can participate in the ballot phases, the passwords of all PC members are changed after the review phase.

In addition to reviewer confidentiality, there is support for blind reviews, i.e., authors, authors' affiliation, and authors' country may be omitted from the e-mails sent to and the HTML pages shown to the PC members.

## 5.4 Usability and Acceptance of Confman

In reaction to DAIS'97, IDMS'98, and other conferences which are currently using ConfMan, we received several reports of conference organizers and PC members via formal and informal communications[5]about their experience with Confman. The general attitude was that Confman is a very useful and easy to use tool. As one of the most positive aspects it has been mentioned that Confman allows every PC member to participate in the PC meeting if time, distance or money do not allow to travel to the conference place. Further positive aspects in the feedback include 24 hour

5. The formal communication has been performed via questionnaires that have been answered from PC members and informal communications is based on personal discussions.

availability of the system and the fact that it brings a methodical procedure into the reviewing process.

Generally, all conference organizers and PC members that gave us feedback regard all features of ConfMan as very useful. The only exception is the on-line PC meeting component about which opinions are quite controversial. Therefore, we report only the evaluation results of the on-line PC meeting with respect to the following aspects:

- **Response time**: PC members valued the response time they experienced with ConfMan mainly as "acceptable" and sometimes as "poor". These complaints have been addressed by speeding up the system as mentioned in Section 5.1.
- **Availability**: The availability has only been a concern for PC members of EDOC'98, because of particular negative experiences. Frequent power interrupts due to constructions caused frequent reboots and large downtimes of the ConfMan server during the on-line PC meeting. These interruptions made it nearly impossible to perform the on-line PC meeting. A mirrored installation or a fallback internet access might be used to considerably reduce this risk.
- **Design and navigation**: Design and layout of ConfMan was evaluated by the PC members as easy to understand and easy to navigate within. The log files confirm that PC members have been able to use ConfMan in a straightforward way without danger of “getting lost in hyper space”.
- **Social aspects**: All returned questionnaires expressed that social aspects are a very important issue and are missed when performing only an on-line PC meeting. Surprisingly, the PC members expressed controversial opinions about the effects of this fact. The majority simply regard the missing personal and informal communication as negative. However, some PC members say that it enables a more comprehensive and objective paper selection process. Decisions within ConfMan are transparent for all PC members and personal deals between PC members are not supported by ConfMan.
- **Economic issues**: ConfMan offers an alternative to expensive, time-consuming and travel-intensive PC meeting with a small number of participants. The time effort needed for the paper selection process was mostly evaluated as higher for the PC on-line meeting than for traditional PC meetings. But considering the overall time effort including preparation and travel time, ConfMan saves time in an order of magnitude.

## 6 Related work

To evaluate the functionality of ConfMan, we have studied several conference organization systems. Information about these systems are retrieved through e-mail conversations with the system implementors and by accessing their WWW pages. We have compared ConfMan with five other systems. These are:

- WIMPE (Web Interface for Managing Programs Electronically) is a set of Perl and Tcl/Tk scripts supporting electronic paper submission and reviewing for a conference [11]. It contains functionality for sending e-mails to authors and reviewers, assigning papers to reviewers, and generating different reports, e.g., from the review phase.
- EDAS (Editor's Assistant) is a Tcl programmed, WWW based software package for paper submission and reviewing [7]. EDAS sends confirmations, notifications etc. automatically by e-mail, and functionality for generating reports like state-of-the-art of the review phase are sup-

ported.

- SIGACT has an electronic submission server that handles paper submission and an electronic program committee software package used occasionally for an “electronic PC meeting” to skip the physical one. Information is sent to the server in predefined forms by e-mail [9].
- The PUMA (the Cornell Publication Manager Project) WWW based conference management tool that supports paper submission and reviewing [7].
- SIGDA was initially a system for paper review in ACM TODAES Journal (Transactions on Design Automation of Electronic Systems), but has later been modified to support submitting of papers for review, as well.

In addition, we know of three other systems: the AAAI conference review software, the SIGPLAN conference software, and the Witan Web system [12]. Unfortunately, we have not been able to obtain enough information about these systems, and they have therefore been omitted from the evaluation. The results summarized in Table 2 show the most important and time consuming parts of organizing of a conference.

| | ConfMan | PUMA | WIMPE | SIGACT Electronic Submission Server and Electronic Program Committee Software | EDAS | SIGDA conference software |
|---|---|---|---|---|---|---|
| Paper submission | y | y | y | y | y | y |
| Reviewing | y | y | y | y | y | y |
| On-line PC meeting | y | n | n | y | n | n |
| Registration of participants | y | n | n | n | n | n |
| Sending e-mails | y | y | y | y | y | n |
| Generating reports | y | y | y | y | y | n |
| Editing stored information | y | y | y | y | y | y |
| Persistent storage | mSQL DBS | predator DBS | plain files | plain files | any hashed DBS | DBS, plain files |
| Programming language | mSQL lite | c++, sql, perl | perl, tcl/tk | perl | tcl | |
| Freeware/commercial | freeware | freeware | freeware | freeware (preserving attribution to SIGACT) | freeware | in-house development |
| User interface | www, e-mail | www, e-mail | www, e-mail, tcl/tk | e-mail | www, e-mail | www |
| Security | HTTP authentication passwords and user authorization via DBS lookup | HTTPS protocol | Reviewers have a unique code which is embedded in the review forms, and “secret” URLs | Submitters get a key that is used to identify their files | HTTP authentication passwords, and papers, reviewers, and reviews are assigned cryptographically random keys. | Restricted user access via Netscape nsconfig protocol |
| Concurrency | via DBS | via DBS | reviews are submitted through an e-mail interface, i.e., serialization by processing one e-mail at a time | reviews are submitted through an e-mail interface, and each reviewers have a separate file | via DBS | via DBS |

**Table 2:** Comparison of different conference organizing tools.

Our first observation is that ConfMan has more functionality than the other systems. Only one other system supports an on-line PC-meeting, and only ConfMan supports registration of conference participants.

As described earlier, ConfMan uses the HTTP authentication mechanism which is similar to the other systems using WWW as interface. However, when data is retrieved from the database through protected scripts, and the HTML page is generated, there are no protection mechanisms for transmitting the data over a network via WWW (or e-mail). Only EDAS provides cryptography through random keys. The e-mail based systems uses a unique ID or personal key to submit reviews into files. Furthermore, ConfMan also protect pages and scripts containing information and functionality meant for the organization committee only, via a lookup of authorized persons from the DBS. The lookup is based on the login name from the HTTP authentication mechanism. In addition, the systems using WWW may monitor the activity in the log where all accesses to our pages and scripts are written.

In all the systems which uses a DBS for persistent storage, concurrency is supported through the DBS concurrency control. The two systems based on plain files avoid the concurrency problem by only allowing submission of reviews via e-mail. However, this makes it more difficult to ensure that the reviews are sent in a format that can be understood by the system. If something is wrong with the format, the review either has to be inserted manually into to system by someone in the organization committee, or an e-mail has to be sent back to the reviewer asking him or her to resubmit the review in a correct format. If on the other hand, the review form is presented on a WWW page, the reviewer is forced to use the required format, and f something still is wrong, feedback is provided immediately.

## 7 Conclusion

The main challenges and goals for the design and implementation of ConfMan have been to develop an integrated system that ensures data integrity, appropriate usability and performance, and which allows flexible and fine-grained protection of different parts of the data. The general conclusion from our evaluation is that ConfMan meets our design goals. Furthermore, our and others (not yet quantified) experience is that by using ConfMan, the organization committee can save a lot of time. The already broad usage of ConfMan is probably the best argument for its usefulness. ConfMan was/is/will be used at the following workshops and conferences: DAIS'97, IDMS'98, WOSP'98, MoMuC'98, Estelle'98, EDOC'98, KiVS'99, ACM SIGCOMM'99, IDMS'99, and DAIS'99. The use of ConfMan is planned for FORTE/PSTV´99 and IWTCS´99.

At the time we began the work on DAIS´97 and IDMS'98, we knew of no applications sufficient to support conference organization. This was the initial motivation to develop a system like ConfMan. Later on, we discovered several other systems, as shown in Section 6. The clear advantages of ConfMan compared to the other systems are: the on-line PC meeting, the integration of all organization tasks in one system, and its improvements based on experiences from the broad usage of ConfMan.

The most “controversial” part of ConfMan is the on-line PC meeting. There are two major reasons for this: (1) the first version of the ballot script had unsufficient response time, and (2) it represents a very different way of performing a PC meeting. The new implementation of the ballot script considerably reduces the number of queries in the mSQL server and by this the overall response time (in our experiments for example from 30 seconds to 10 seconds). The loss of the personal aspects of a face-to-face meeting has been mentioned as disadvantages of this new approach of PC meeting. However, the ballot and discussion of IDMS'98 have shown that the discussion during the online PC meeting took place in a very natural manner. PC members referred to arguments and opinions of each other in the same way they would have done in a face-to-face PC meeting. Furthermore, ConfMan enables a more comprehensive and objective paper selection process, decisions within ConfMan are transparent for all PC members, and it allows all PC members to participate in the PC meeting regardless of time, distance, and money constraints.

In our opinion, Confman should be applied in connection with a face-to-face meeting. This gives a opportunity for a personal discussion, but in contrast to previous PC meetings all PC members can participate in the final and not only those who live usually not too far from the meeting place. This might be a trade-off for a reasonable combination of both ways which may be applied as long as appropriate video conference services via the internet for PC meetings are not available.

The above mentioned reactions reflected only the opinions of PC members. The feedback of conference organizers themselves was entirely positive because the system facilitated their work considerably and allowed them to automatically trace all the processes of the conference organization.

We offer ConfMan with certain disclaimers as public domain software to other interested persons respectively institutions (see http://confman.unik.no/). The system will be developed further in cooperation between our both departments. A new version is planned for 1999. Furthermore, we intend to take care of the software and fix reported bugs - as far as our time allows.

**Acknowledgments:** We would like to thank Martin Karsten from Darmstadt University of Technology for providing us with the source code of the IDMS'97 paper registration system. We would also like to thank Daniel Dietterle and anonymous students at Cottbus University for implementing the original OBSession system.

## References

[1] APACHE, http://www.apache.org/

[2] EDAS, http://www.cs.columbia.edu/~hgs/edas

[3] Elmasri, R., Navathe, S.B.: "Fundamentals of Database Systems", Second Edition, Benjamin Cummings, 1994

[4] Hughes Technologies: "Mini SQL 2.0: User Guide", Release Version mSQL 2.0.1 (2.0.1 v1), Hughes Technologies Pty. Ltd., Australia, July 1997

[5] Mini SQL (mSQL), Hughes Technologies Pty. Ltd., http://www.hughes.com.au, 1998

[6] Nicol. D. M.:"WIMPE: Web Interface for Managing Programs Electronically", Users and Implementation Guide, http://www.cs.dartmouth.edu/~nicol/wimpe,
http://www.cs.dartmouth.edu/~nicol/wimpe/wimpe4.ps

[7] PUMA, http://www.cs.cornell.edu/Info/Projects/PREDATOR/puma

[8] Quantify, http://www.rational.com/

[9] SIGACT, http://sigact.acm.org/sigact/esubserve.html,
http://sigact.acm.org/~shor/e_committee.html/

[10] W3C: "CERN httpd", http://www.w3.org/Daemon, June 1996

[11] WIMPE, http://www.cs.dartmouth.edu/~nicol/wimpe/wimpe.html

[12] WitanWeb, http://witanweb.iit.nrc.ca/