Web-based conference organization - a summary of computer experiences from PSE'97/ESCAPE-7

S.Hauan*, T.Krokstad, S.Skogestad and K.M.Lien
Department of Chemical Engineering
Norwegian Institute of Science and Technology (NTNU)

BACKGROUND

This article is written to summarize our experiences with electronic document collection/(re)distribution in relation to PSE'97/ESCAPE 7 [1] held in Trondheim, Norway may 25. to 29. 1997. More than 600 abstracts were received and, through review, reduced to 201 accepted papers published in a special issue of *Computers and Chemical Engineering*. Approximately 400 participants from 32 different countries were present at the symposium, with industry counting for approximately one third.

More information and sample script files are available at

http://www.kkt.chembio.ntnu.no/research/PSE_ESCAPE/

MOTIVATION

By using Internet as information carrier, there's a large upside potential for fast and structured distribution. Most universities and major companies are now online, and more are expected to join shortly.

Electronic collection and distribution of abstracts/articles offer several advantages to conventional, paper based systems through regular mail. We identified 6 major issues, roughly ordered by impact and importance:

1. **improved speed of distribution**
   Electronic transmissions are independent of slow and - from some countries - unstable postal communication. The ability to send a file, not paper, saves people outside western Europe up to 4 weeks compared to regular mail. Although courier is an alternative, it's usually quite expensive.
2. **easy updates**
   The authors are themselves responsible for paper layout, and, due to the possibility of speedy retransmissions, one may be (a lot) less restrictive to request changes. Usually, such requests are impractical due to time constrains

*Corresponding author. E-mail: steinhau@pvv.org
[1]joint event of by the Executive Committee of the Process Systems Engineering Symposium Series and the EFCE Working Party on Computer Aided Process Engineering

and in practice hardly ever done "just for conference proceedings" unless the layout is completely wrong.

3. **reduced workload**
Automatic collection and storage of cover-letters, abstracts, papers and eventual updates helps keep down the need for manual intervention. Some verifications/corrections should be expected, but they're significantly less than taking care of everything in person. Proper logging routines assures a complete history of communication with the author(s) with a minimum work of copying/archiving/indexing information that would almost certainly not be accessed except in the event of errors.

4. **higher level of consistency**
By refusing to accept articles with incomplete cover-letter information (minimum: corresponding author with address, fax and phone number), we avoid "searching for people in charge" of a specific paper.

5. **systematic feedback; automatic and manual**
Any submission may be immediately verified by mail or fax[2]. This **both** checks our primary means of communication with the corresponding author and provides a receipt of submission.

6. **more time for reviewer**
The ability to redistribute documents to reviewers, in particular those living very far away, saves a lot of time compared to regular mail. Hence one may either extend the deadline(s) for when abstracts/papers are due, give reviewers more time/flexibility for their job or combinations thereof.

Specifically, we aimed at solving all or parts of the following tasks through a web interface:

- **collection** of abstracts/papers/updates
- structured **feedback** to authors
- **redistribution** of material to reviewers
- passive **information** to authors and participants

The information part was trivial from an implementation point of view. Menus on the conference homepage were linked directly to documents generated in HTML and/or PostScript. Standard reminders or generic information was sent to authors using either mail (preferred) or fax - in addition to the usual paper leaflets.

The remainder of this article will describe structural and implementational issues of the first three aspects. We will also present statistical results and briefly discuss possible improvements to a similar arrangement in the future.

REVIEW & PUBLICATION PROCESS

From the users' point of view, a submission consists of 3 basic steps:

1. prepare a postscript version of his/her abstract or full paper

[2]through our mail gateway

2. transfer the postscript file to our anonymous ftp-server
3. complete the electronic cover letter in his/her web-browser

The final step iterates with the user to ensure a file has been sent and the cover-letter contains all required information. Step 3 is the "heart" of the process, iterating with the user to correctly glue together the file submitted in step 2 with information about the author(s). Details about the underlying programs are found in appendix A and B.

After successfully completing the electronic form in step 3, an *automatic receipt* is sent to the mail/fax address given for corresponding author. The ftp file is also *locked* to avoid multiple cover-letters pointing to the same file. For security reasons, outside users are not allowed to browse any files outside the ftp area, neither are any unprotected web-script without allowed to move files from the ftp directory. Instead, a completely independent process - known as a *collecting daemon* - is run every 15th minutes with the single task of "safely" moving any successfully[3] submitted document from the ftp area to our processing queue (spooling directory).

Once in the spooling directory, all documents are processed by the conference secretary through a password-protected webpage. The main tasks are to verify printability of the submitted file, check for possible "spoofing"[4] and get a copy for our paper based backup system.

There are three possible actions when fake submissions have been discarded:

- **ACCEPT**: sent back to the author
  Everything is ok. We mail/fax a *manual submission verification* to the corresponding author specifying the proper reference number and store the file in our database. Dependent of submission type (abstract, full paper or final paper) we add various messages like deadlines, registration information and reminders for copyright transfer forms.
- **TECHNICAL**: internal message to the administrator
  something is wrong, our secretary is unable to print whole or parts of the document. The case - file and a short description of the error encountered - is forwarded "technical administrator" who looks at the file and attempts to figure out what is wrong and if anything might be done to make the file printable <u>without</u> involving the author. Usual problems and a few solutions are discussed later. After analysis, the file is moved back into the spooling queue for reprocessing; either with a fix enabling a successful printout or with a message intended for the corresponding author if no solution was found.
- **REJECT**: sent back to the author
  the document is incomplete or unprintable and we warn the author(s) that we *for technical reasons will be unable to use the file in it's current form.* Attached is also the above mentioned message from the administrator with more details about the cause of the error and (hopefully) the steps required to create a file in the proper format. The original submission is removed from the spooling queue and temporarily stored

The PostScript files were distributed to reviewers by e-mail as zip-archives, encoded by MIME <u>and</u> by regular mail. A few connections where paper based information was known to be slow were tested in advance and known to work well.

[3]meaning we had file in the ftp area attached to a complete electronic cover-letter

[4]an irrelevant submission of garbage by someone who just accidentally found our submission interface and attempted to make our life tougher.

RESULTS & STATISTICS

Reception: articles and registration

Overall, 733 of 968 (74%) successful submissions of abstracts and articles were made through our web-interface. In particular, almost every updated article was sent electronically. Even though it is reasonable to assume almost everyone had been sent a copy of our paper leaflet, quite a large fraction found it more convenient to print and fax the registration form found on the web. Details are found in tables 1 and 2 respectively.

type	total	electronic copy		paper only	
abstract	575	395	(69%)	180	(31%)
full paper	241	186	(77%)	56	(23%)
paper update	154	152	(99%)	2	(1%)

Table 1: Reception by type

total received	from paper leaflet	from webpage	other
387	254	117	16

Table 2: Registration by type

Error statistics: what was difficult?

Overall, there were 97 rejected submissions out of 733 attempts (13%), slightly higher than the estimated 10%. Most errors were quickly corrected in one iteration with the corresponding author, and in only 8 cases (1%) we were unable to solve the computer issues involved and had to demand a paper copy.

The errors may be broken down by type as follows:

	abstract	full paper	paper update
electronic copies	395	186	152
#completed forms	474	214	195
#submissions in order	421	174	153
#technical rejects	37	24	36
other response	16	16	6
#different people	26	20	27
wrong format	22	15	26
incomplete file	12	3	1
incorrect layout	n/a	2	9
other errors	3	4	0
never succeeded	5	2	1

Table 3: Electronic submission statistics

Remarks:

- The errors are listed in priority order, meaning a submission containing an error with respect **both** to file format and to layout, only will be listed as a format error. In practice, the notification saying *technical reject* had a comment on both issues.
- Typical format errors, listed with decreasing frequency, were caused by modern, level 2, PostScript, raw word processor files, PCL files and non-conforming Windows or Macintosh printer drivers.
- The row for "other response" in table consists of submissions after the deadline and those who submitted the same file more than once or sent an updated copy before the previous version had been processed. It also includes a few cases of manual intervention in cases where authors had 2 or more earlier submissions rejected.

Roughly 10 authors had severe trouble producing conforming PostScript and their initial submissions are found in the error columns for both abstracts and full paper. Their persistent attempts to use our submission interface were greatly appreciated, as it aided us in identifying flaws in our design- and help systems.

Redistribution

Redistribution by email got a mixed reception. Very few reviewers specifically asked for the paper based copies, but only about 25 percent actually confirmed their use of the electronic versions sent to them. The latter group had the opportunity to distribute their workload more evenly in time. Late papers could also be accepted for more than a week and then sent directly by e-mail in a second batch to those who confirmed this as a working alternative.

DISCUSSION & EXTENSIONS

In general, most people had no trouble using the electronic form from their web-browser. The number of errors in transferring files by ftp was also surprisingly low. On the other hand, the total number of problems encountered were slightly above expected (13 versus 10 percent). Although most computer systems have the ability to generate proper postscript files, this step is frequently completely transparent to the users, hence preventing actual insight in the underlying mechanisms. A step-by-step instruction for the most commonly used word processors in a web page could be helpful.

The use of PostScript as an "exchange format" worked out well, mainly for 3 reasons:

- independence of machine- and word processor type and version
- authors themselves are responsible for layout
- no binary file transmissions are needed

In order to minimize the number of file format errors, one should consider to allow download of proper printer drivers[5] for the most commonly used PC and Mac

[5]beware of copyrighted drivers

platforms. From our experiences, no special actions are needed for UNIX users as they rarely had trouble with non-conforming PostScript.

There are several possible extensions to the scheme described in this paper. The most obvious one would be to require all authors to submit their work in electronic form. Considering the large fraction of authors who already did so, this is hardly expected to cause major concerns. With email as a backup solution, the need for shipment of paper based copies should be minimal. Similary, one may opt to perform distribution and collection of reviews in electronic form only.

It is also possible to avoid the use of ftp for file transfers. In recent versions of HTML, one may create electronic forms capable of reading files directly from the users' home machines. A further refinement is found in Java where an ftp transfer process could be run in the background with the PostScript file selected from a dynamic menu. This latter option was considered and actually tested in the early phases of implementing our transmission system, but discarded due to the limited use and distribution of Java at the time to avoid turning anyone away purely for technical reasons.

A more futuristic approach would be to both collect, review and **publish** articles without ever ship a single page of physical paper. Such an electronic journal opens for a large variety of interesting possibilities. There would be no page limits, no firm deadlines, complete cross references and easy mechanisms for searching the complete journal. Each article could also contain links to demonstration packages on the author's home machine to allow readers to repeat simulations and study the details behind what is found in the publication itself without spending large amounts of time on their own implementation. A willing author could also allow reader-defined perturbations in parameters, physical states or specifications to study specific behaviour of the cases in question. Although an interactive, electronic journal should not be expected to replace existing ones, it would most certainly be a valuable supplement.

It should finally be noted that the sole author of the underlying programs was physically in the U.S., several thousand miles away, while the whole thing was running. Any difficulty encountered processing incoming files was resolved through email and the occasional phone call.

CONCLUSIONS

We have successfully implemented and used a web-based system for collecting abstract and articles for a conference of reasonable size. A major fraction of the authors chose to submit their work in electronic form, saving both themselves and especially the conference secretariat time, money and effort.

Appendix A: Implementation details

The underlying programs were written in perl, a scripting language optimized for text-processing, and run through the CGI-mechanism[6] in the webserver.

- main submission page (HTML) with links to help pages on how to generate postscript files and how to use ftp
- main submission script, recursive on 3 levels:
    1) pick your filename from a dynamic pull-down menu, press continue on the bottom of the page when done. Except when submitting abstracts, the user is prompted for a reference number and a password[7]. Further details about specific checks and internal information flow is given in appendix B.
    2) Iterate with the user for list of authors, paper title and contact information for corresponding author. Some fields are mandatory, others are not.
    3) create a temporary file for cover-letter information, lock the selected file to disable resubmissions and send an automatic receipt to corresponding author. Print a message on the screen saying where the automatic receipt was sent and that a manual verification should be expected, usually within a day or two.
- internal verification script, used to check the incoming abstracts/articles before they were moved to our database.
- collecting script, independent and inaccessible from the outside, in charge of safely moving submissions from the world-writable ftp area into the protected spooling queue.
- various utility script, performing consistency checks, generating reception statistics, collecting reviews, writing accept/reject messages to authors and

[6]Common Gateway Interface

[7]as documents are further processed by our secretary and no direct access is permitted, this only serves the purpose of eliminating potential spoofing. We just asked for last name of corresponding author as password to avoid extra administration

Appendix B: Internal checks

Several consistency checks should be performed as the users complete the electronic cover-letter.

a) complain loudly and return if form is incomplete, the given "password" is incorrect or the reference number is invalid or previously rejected by review.

b) complain loudly, print and error message and return if the user selects an obviously incorrect "file", for instance a directory or the README file.

c) complain loudly and ask the user to contact the administrator by mail if the submission is identified as an "update", meaning it will replace any existing file in the database if processed as requested. Such files are NOT copied into the spooling queue to avoid possible mistakes by whoever is doing the manual inspection.

d) complain softly, print a warning message, but continue to step 2 if the selected file fails a few simple test. Typically, we test for pure text, binary files or anything with zero size indicating error(s) during ftp transfer. The file is however still accepted and later processed; This way, the casual user will manage to submit an incorrectly generated file and then, in turn, get specific feedback from us. On the other hand, some experienced users might catch and correct their error(s) directly without our intervention.