Paperwork on the Fly

5 AUTHORS, INCLUDING:

António Pedro Aragão
University of Minho
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Carlos Baquero
University of Minho
73 PUBLICATIONS **391** CITATIONS
SEE PROFILE

Luis Barbosa
University of Minho
117 PUBLICATIONS **568** CITATIONS
SEE PROFILE

Pedro Rangel Henriques
University of Minho
209 PUBLICATIONS **480** CITATIONS
SEE PROFILE

Paperwork on the Fly

José Luís Faria, António Pedro Aragão, Carlos Baquero, Luís Soares Barbosa, Pedro Rangel Henriques

Universidade do Minho - Departamento de Informática-Portugal
{jose,aaragao,cbm,lsb,prh}@di.uminho.pt

Abstract

A particularly boring duty of every university lecturer is the filling up several academic management forms like: course: program, objectives, bibliographic lists,…;lecture notes, exercises,…; class: timetables, summaries,…; etc – some of them on a daily basis. He/she must refill these forms again and again, every semester and every year, sometimes with replication in different forms and comply with various requirements. This type of work require time borrowed from research and teaching.

With this in mind, we developed an web application to carry on such bureaucratic jobs in a quick and more effective way.

This paper introduces ADIS-Academic Department Information System, a web-based application currently under testing and implementation on the Departamento de Informática of Universidade do Minho.

It also describes in more technical detail, the *Fly* component and the lessons learned after its use along one academic semester. This on-line web application allows the lecturer to register summaries, programs, list its own timetable and print them. Summaries can also be filled using an email message.

The server which promotes this application is a usual PC with Linux, MySQL, PHP and a web server Apache.

Keywords: Web Application, e-University, Academic Information Managment.

1 Introduction

Every university lecturer is committed along every academic semester and every year to the filling up of several paperwork with some data about: course: short and detailed syllabus, objectives, bibliographies, …;lecture notes, exercises, …; class: timetables, summaries, …; students marks, etc. Some of this forms, in paper, must be filled on a daily basis. Many times these information, in different forms, must be provided to the administration. He/she must refill again and again some groups of these forms and comply with various requirements.

This paperwork is actually done by hand, by lecturer, and requires a lot of time. A precious time which is needed to research, teach and for preparing the class.

2 The Main Goals

We are technological ready to promote a better way to do this paperwork, not by hand but electronically.

The first goal we want to achieve is almost replace the paper by email and a web browser.

The second goal is, to build a system, which is able to pre-fill some forms, and leave to the lecturer only the field he/she must write. For example, based on the timetable of a given lecturer the system may suggest the summary layout and leave only the abstract of the summary to be written. Although he/she can change everything, if needed.

Another goal is replication control. Every time she/she needs another copy or the same data he/she can produce another document accessing to a web browser. Or better still, the administrative services may access to the database to list all information they need.

The last main goal, is prepare the system to give the lecturer an easy way to make available the summaries and other course-related data on the web to the students.

3 Our Proposal

The ADIS is a medium system under development in our department. At this moment one component is completed and on-line, which manage the data about summaries, programs and timetables. This component of ADIS is called *Fly*.

Another concern in the department refers to publications. We teach but we also make research. Every lecturer have articles, PhD Thesis, MSc thesis, etc. It is necessary to implement a new management structure of organize all the publications and make available on our web site.

The basic schema for ADIS is in the figure 1. There is three different kinds of access: administrative access to provide: calendar for the semester, the timetables, …; obtain the programs for courses; obtain the summaries; other information; lecturer access to insert summaries, syllabus, list other information; and finally the public access to students or other people to search for publications, abstracts of publications, summaries, programs, etc.

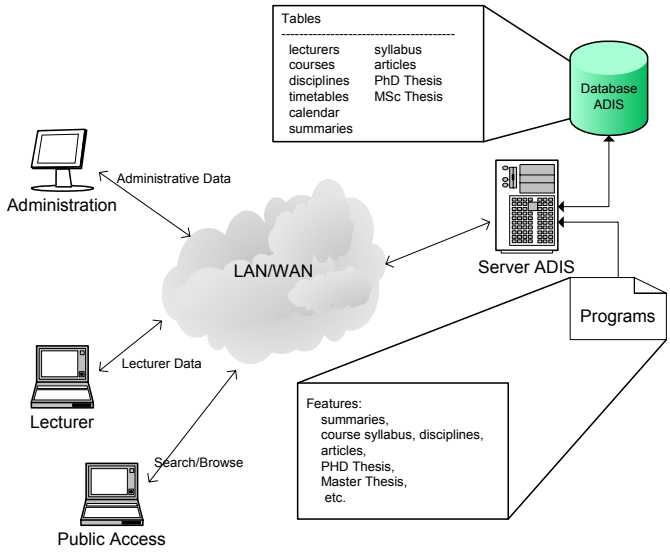

Figure 1 – Global schema of ADIS

2.1 The Fly

This component has been on-line for about one semester. Its structure is detailed on figure 2. It is in the database server that the system saves all data. Each lecturer has a profile in it which may be edited at any time. The profile has some options like: type of results: PDF or PS; obtain results by email[1] or by web, summaries sent by email or not; lecturer email, URL, etc.

At the beginning of each semester the tables "calendar" and "timetables" must be filled with the appropriate information. After this operation the system is ready for use.

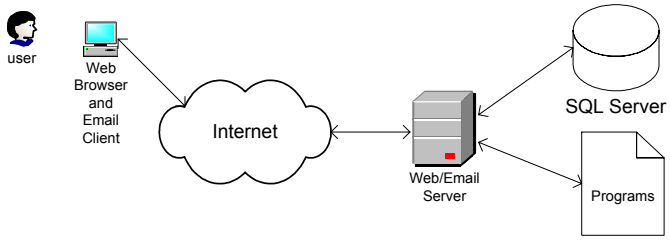

Figure 2 – Global schema of the *Fly*

This component offers a menu with options about summaries (insert, edit, remove, browse, print and reprint), syllabus (insert, edit, remove, browse and print a short, detailed syllabus), browse and print the timetable; browse the courses, disciplines, the lecturers and books owned by the department.

Printing in this application does not entail the usual meaning, it means generate the documents and send them by email or give a URL to download them.

A special effect is implemented on the summaries. The lecturer can insert a summary in a form on the web browser, this is a normal way. Or else he/she can choose, in *Preferences*, to fill summaries by email. It means, he/she receives an email message every 5 minutes after the end of each class. This message contains the last summary of the class and new one pre-filled. The user just has to reply this message by writing the abstract, and the system inserts on the summaries table. The lecturer receives a notification message if it the summary was correct or if troubles were detected in the message: for example missing data, wrong data, etc.

4 The Web Application

This system is composed by one usual PC as a server with some tools freely available as described in table 1.

Software	***Description***
Linux Red Hat	Operating System
Apache	Web server
MySQL	SQL server
PHP	Language Script

Table 1: Software used on the System.

The choice of interface based on a web browser is very important, because the lecturer can use any operating system with any web browser to access to the system. All the pages of this application use the basic HTML and Forms[3]. At this moment every web browsers, in almost all versions, recognize and implement forms. Even it the lecturer uses its own old PC at home, he/she can access the application through the Internet without problems, in despite of an eventually slow connection.

Other important choice is the use of PHP[2]. It is powerful, simple and runs on the server. By the way, using PHP enables the dynamic generation of HTML pages in an easy way. Moreover it has a good set of features dedicated to access to a SQL server.

The system use the MySQL and Apache because they work very well on a PC without any sophisticated hardware. The server is a Pentium MMX-166MHz, 64MB of RAM and hard drive with 1.3GB.

To improve security, the system has a table on the SQL server with the login number of each user and each of them has a password assigned. This password is encrypted and private to the lecturer. This a first security level. The web server uses the module of SSL to implement a secure tunnel between the server and the web browser client. With this approach it is possible to use the Internet at any place with some security.

All this system works very well and potentially supports the around sixty lecturers of the department.

5 Conclusion

Some experience has already been gathered with the *Fly* component. At that moment it manages data about:

- 57 lecturers
- 57 timetables
- 72 courses
- 127 disciplines
- 9 syllabus
- 779 summaries

Around 21 lecturers are using the *Fly*, some of them inserting summaries in a web browser and others by email.

Although by law a hard copy of the summaries is required, this system contributes at the following levels:

- regular notification of the need to write the summary
- uniform presentation of summaries and syllabus
- easy replication of any programme
- access centralized, in a electronic way
- form filling speed up
- hardcopy with uniform look
- reduce mistakes with the date and hours
- possibility of resorting to SQL to generate new reports with other combinations.
- simple and effective editing facilities

The opinion of almost users is very positive. The system has benefited from some improvements given by the users.

5.1 Future work

We plan to develop the remaining components and integrate all of them. The rest is now under development.

In the beginning of the next semester, September of 2003, we plan to make available a new release of *Fly* with some new features. Coming from suggestions of the final users. We hope make available the part of ADIS related to publications with its first release.

In the rest of ADIS we will try solve problems arising with the publications managing systems. Using databases as a repository for information (such as members, publications and other documents), we can have a centralized management system. The idea is to submit information about a publication and let someone approve or disapprove it. Then, if approved, the publication will become available on-line. It will be possible to have all the information about all the available publications and other documents. In this way we can provide fast access to information (example: someone wants to know written papers about a certain issues) and speed up workflow of documents.

Let us describe all stages and capabilities of this component in ADIS system. In a first stage we introduce all the information about the publication (title, date, ISBN, abstract, etc) and then link it with authors, whose information is already in the database. Then, in a second stage, commonly known as the commit stage, someone with enough privileges, such as the information management administrator, will accept or reject publication. This stage also can be seen as a validation step, since this kind of information is sensitive and maybe made available for browsing with a web browser. Finally, the last stage, provides this information on-line for people, with the usual facilities: searching, browsing and downloading. Of course the features of this component don't end here; this information can be integrated with other information and then contribute to other documents such as department annual report. This component, like others in the ADIS system, is achieved using a set of forms and HTML dynamic pages for the input process (stage one) and accept process (stage two). In the output process the ADIS system provides HTML pages, other text sources format media such as RTF and TeX and other final text format media like PDF e PS. This component solves the problem of having the information about publications scattered around our department and impossible to be found by an on-line searching tool. Now we can have the information in electronic formats that helps people by making easier things like fast searching and downloading. Again it is possible to obtain other text formats for build many department reports.

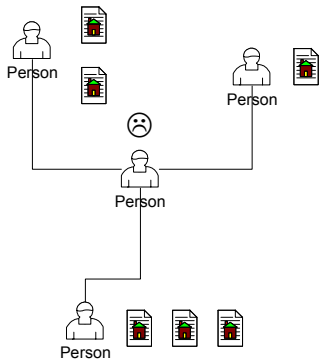

Figure 3 – System in use

There are available on the web several self archiving programs. One that is starting to be used by a lot people is **GNU EPrints**[4]. It is important in the future that the ADIS system interacts with this kind of programs. Maybe, in the near future, **GNU EPrints** or other programs becomes an

internet standard program for self archiving publications. Therefore, it will be important that integration with other information systems will be possible using the ADIS system.

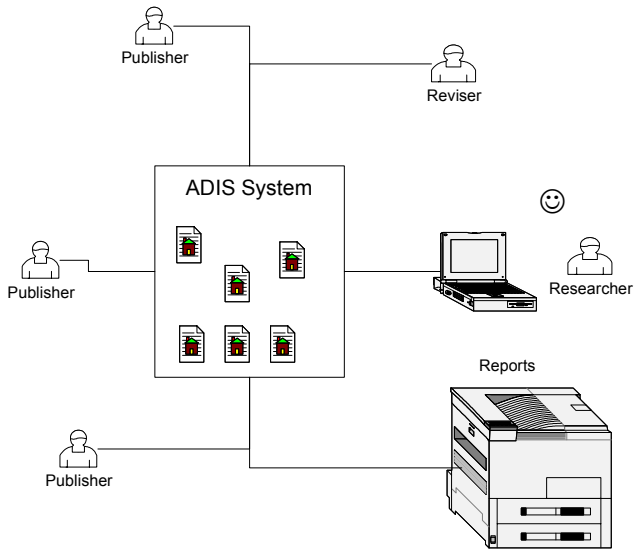

Figure 4 – Proposal of implementation

References

[1] Blum, Richard. "Running qmail", SAMS Publishing, 2000.

[2] Schwendiman, Blake. "PHP4 Developer's Guide", McGraw-Hill, 2001

[3] Stein, Lincoln D.. "How to Set Up and Maintain a Web Site", Addison-Wesley, second edition, 1997

[4] http://www.eprints.org