

The building of a successful zero administration homepage system for students and its relation to the wider campus infrastructure.

Alan Berg*, Léon Raijmann[†]

*Informatiseringscentrum, University of Amsterdam, The Netherlands.
a.m.berg@uva.nl
l.e.l.raijmann@uva.nl

Abstract

The homepage system for students, at the University of Amsterdam, has run stably and successfully for the last three years. At present over 3200 student homepages exist and the number is growing. The administration of the student sites is delegated to the students themselves and is integrated with the campus directory service. This implies virtually zero administration costs. The system is light weight as it is only comprised of only a few Perl scripts and some clever system configuration and a few extra services. Judging by the popularity of the system and its history of stability, the technologies involved may have a wider relevance. This paper describes the cocktail of simple components that allow this service to function and says a little over the students that use the system.

Keywords: Website, Integration, Management

1. Introduction

The following screen grab shows the entrance to the student homepage system [1]. The system is simple in intent and scope, costs as little administration as possible, is cheap to run, scalable and uses well known technologies [2-7].

The end user experience is seamless. To create and manage a homepage requires little input. The interaction for creating a homepage for a student is as follows; a student logs into a web interface with their *uvanetid* [2], a personal and unique number provided by the University, they then input their password, accepting a user policy contract and within a few seconds they have an automatically built website. From that moment on the student is free to edit and/or upload files to their directory on the website and start building a magnificent statement. Over 3200 students have already chosen to do so at the University of Amsterdam and the number grows steadily. What is surprising from the administrator's point of view is that there is no administration. The student simply opts in or out. The contract signed during the homepage creation allows for any

necessary administrative action in case of abuse, but this happens rarely.

Behind the scenes the system dovetails nicely into the Campus Directory system that uses the LDAP open standard. Each student and member of staff has a record on the LDAP server(s). Within each record a few attributes are specifically created for the homepage system. As soon as the homepage is created the attributes are updated and viewable.



Figure 1: The homepage systems main page.

The main advantage of this dovetailing is that of password synchronization the user has one password the *uvanetid* password for many systems. When the students log into their

mail system via webmail or at the command line or connects to the University Internet Service or asks for exam results through the University study web, the same password is used and all systems ask the LDAP[8] service for that information. This is more than convenience, it has direct impact on the way the student views the overall campus system. At present there is also password synchronization between the LDAP servers(s) and the ICs Microsoft windows network, leaving plenty of room for extra service later. At present the student population is around 22,000, so we expect with a little advertising that the user base can grow rapidly.

2. The underlying system

The underlying system is powered by clever configuration, a web interface and a few Perl scripts. Figure 2 pictorially describes the detail of this infrastructure

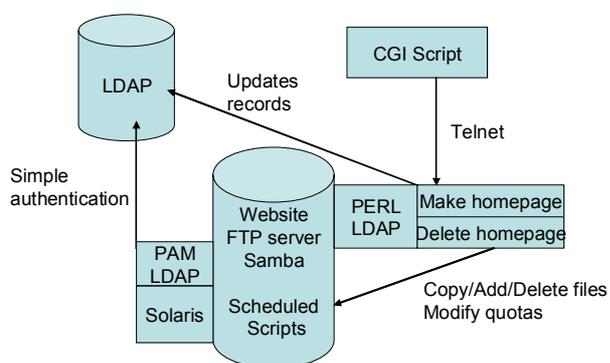


Figure 2: An infrastructure view of the underlying homepage infrastructure.

The specifics of the homepage system are as follows. The services supplied are:

1. Creation and deletion of homepages using templates.

Homepage creation involves running a small piece of Perl code that recursively trawls a set of directories to copy the directory structure and files into the students homepage. The code also changes the value of attributes within the campus directory service, allowing other code later to be aware of the status of the student with reference to this system. The Perl code leans heavily on the use of PerlLdap[7] module.

2. Delegated administration so that the student can opt in to having a homepage.

The native web interface for the delegated administration was that of a Netscape LDAP server which was modified to include an extra section for this service.

3. Authorization through the central directory service.

The homepage system is based on Solaris 7.0 with **Pluggable Authentication Module** set of services and surrounding infrastructure[4,6]. This has at its kernel a set of authentication modules that can connect to various systems to generate acceptance tokens. The pam_ldap module was most viable for our situation. The advantage of this methodology is that because the pluggins our layered we could choose other systems to authenticate to in a flexible and invisible way. Leaving future expansions or changes to the campus systems viable.

4. The use of the central directory service for maintaining status information on a per user basis.

As is becoming more and more standard the LDAP server(s) are being used as a basis for a Directory Enable Network (DEN). Configuration and status information is stored in a central repository that is available throughout the enterprise. For Windows this is ActiveDirectory and for UvA as a whole the focus is on LDAP.

5. Uploading of files via FTP[3]. The students take advantage of a known interface and can up and download files through FTP. The surprising issue is the lack of issue. FTPing is a skill that the majority of the student population simply has.

6. Support for Frontpage express by extending web server functionality. Frontpage express was chosen as the web editing tool of preference for the service. To dovetail to this tool modification of how the Server interacts with the editor was required. The way this was achieved was rather hardcore, a server pluggin[9] that was written in c.

7. Regeneration of website once an hour to describe who has a homepage. A Perl script was written that performed an LDAP search once an hour and then generated a sub site describing indexed end users. This has proven intrinsically reliable as, if there is a network connection problem, then the last site generated is still seen.

8. Log file analysis once a day with a top ten/100 list created automatically once a day. Another custom script that by brut force trawls through the last days statistics. This code will be discontinued in the following release of the homepage system.

9. Quota service limiting the end user to 20Mb of hard disk space. At the time of the users homepage a quota is set. At present the quota is set at a relatively

low value to discourage the transfer of binaries (for example MP3 files) between groups of users.

10. CGI scripts for form mail, counter generation. The students seldom use this scripts rather opting for links to outside services, for example for statistics nedstat[11].
11. Automatic cleanup. After a student ceases to exist in the University's administrative data source, a count down begins for the student's LDAP record. After a period of around ninety days the recorded is removed. A script that scans the LDAP server once a term spots this and removes unwanted websites. The large delays in time give ex-students sufficient time to transfer content.
12. Hardware. The system is made from one standalone Solaris box. After initial burn in problems the system proved extremely reliable.

3. The student experience

The definition of the student groups is transient due to fashions and technological shift. Students tend to occupy a whole range of knowledge levels and attitudes. You cannot please them all. If you provide too many services it may confuse one group, too little leave the more hungry too hungry. So you are left with the hope to provide specific functionality with clear goals that encompass enough of the averaged feature set. Observing the spectrum of websites does help to give us insight. We roughly characterise our target audiences as:

Non user.

The non user create their own website just to see if they can and after doing so leaves the system alone. This leaves a standard 'under construction' homepage which is left inactive. When trawling the index page for the main website you see a minority of these sites. In fact, the danger of such pages is that it may make the browsing of the website less relevant. Who after all wants to be bored with these pages?

File transfer.

Having a minority of websites report 'under construction' does not imply a large inactive minority, because division of this minority can be made. Many users simply store files on the server so that they are transportable via the Internet. This can avoid the problem for the user of having to physically move floppies and other storage mediums between systems, the so called 'sneaker net' effect.

Experimenter.

The user wishes to learn how to build a website and place different sometimes in appropriate mages and pages together.

Single user.

The single user creates a website for their own personal interest and tends to be quite active in changing their website. As you browse the website or look at the top 100 most viewed WebPages you will be surprised by the depth and quality of many of this type of users work.

Community user.

Although officially only for one student, a number of sites are dedicated to specific groups' interest, for example: Union clubs, personal interest groups or classes. If the homepage system was open to members of staff you would expect a greater number of online communities to emerge. The system would need to be expanded to support such communities via additional workflow and groupware.

Weblogger.

Although this type of user has not emerged clearly yet, we expect an increasing trend towards using the website as a personal dairy. This is known as a weblog. The use of server side scripting is intensive in this type of infrastructure and would thus be expensive in maintenance and service desk support.

Redirects.

A number of sites are simply redirect links to other websites. This allows students to advertise themselves to other students with already completed content.

No doubt there are other user groups than the ones mentioned above, but what we try to portray is the variety of interests involved. It is clear that if we fulfil the expectations or hopes of all these groups, we would end up developing a rich range of technologies focused on a portal centric universe (University). Client side technologies extensively used within sites include flash and JavaScript, if there were server side scripting possibilities for the population, a clear minority would enjoy the opportunity.

4. The near future.

The Informatiseringscentrum plans to upgrade the system in the short term. By the time you have read this document the new system will be in place. The upgrade includes a change in look and feel, improved structure within the original scripts and also the ability for the six thousand staff members from the UvA to use the system. No extra functionality has been added, in fact a number of the indexing scripts have been depreciated. It is a testament to the original idea of an automated delegated system that the expectations of impact on system management time are near zero. The question that is left in this new experiment is: will the existence of students and staff in the same virtual sphere enhance or decrease the overall popularity of the system from the student's perspective? The only thing that we can be certain about is that on average the student population is more technically savvy than the staff. One can now expect more service desk calls.

Finally, the University is busy with the beginnings of a portal centric approach to personalization of content. Web storage is a clear bread and butter requirement. It will be quite a graphic design feat to place the rather non uniform homepages within the in-house style of the portal system. It risks being either too chaotic or too uniform an integration. At the least the resultant view will be fun to see.

4 Summary and Conclusions.

The student population has made good use of the homepage system from the Informatiseringscentrum at the University of Amsterdam. The two main features of the system that allowed a very cheap interaction were the delegation of administration to the end user themselves and the use of the LDAP repository for central authentication and configuration.

In version two of this product, we are personally looking forward to see how the staff and the students homepages intermingle and advertise the Universities unique atmosphere and culture. We also see the alumni population advertise their existence here.

Acknowledgements

I would like to acknowledge the other team players within the original homepage project. These are; Jaap van Ginkel, Jim Mintha and Marijke Vandecappelle.

References

- [1] The homepage system at the University of Amsterdam
<http://home.student.uva.nl>
- [2] Information about the uvanetid.
http://www.ic.uva.nl/uvanetid_studenten/
- [3] The home of the PROFTPD project.
<http://proftpd.linux.co.uk/>
- [4] Information over the pluggable authentication module.
<http://www.dementia.org/~shadow/pam.html>
- [5] The home location of the Samba project.
<http://www.samba.org/>
- [6] A rich source of information for the Solaris operating system
<http://www.sun.com/bigadmin/>
- [7] The Perl LDAP project FAQ.
<http://www.mozilla.org/directory/faq/perldap-faq.html>
- [8] IBM redbook over the details of an LDAP infrastructure.
<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg244986.html?Open>
- [9] The NSAPI programmers guide.
<http://developer.netscape.com/docs/manuals/enterprise/nsapi/>
- [10] Nedstat, a free statistics webservice.
<http://www.nedstat.nl>