

Meeting Diverse User Needs: Implementation of a Departmental Information Strategy

Richard Jones, David Beckett and Sally Fincher
Computing Laboratory, University of Kent at Canterbury
{[R.E.Jones](mailto:R.E.Jones@ukc.ac.uk), [D.J.Beckett](mailto:D.J.Beckett@ukc.ac.uk), [S.A.Fincher](mailto:S.A.Fincher@ukc.ac.uk)}@ukc.ac.uk

26 February, 1999

Introduction

Any higher education department must provide a variety of information to several disparate audiences, recognising their diverse needs. Student expectations of learning support structures, institutional needs for quality assurance and enhancement and the demands of accountability from funding agencies continue to grow. The widespread use of IT, both at home and in schools, has led students to expect a rich networked computing environment; the widened pattern of access to higher education has simultaneously increased the level of support needed and limited access to the campus for many of them. Managing these challenges and changes demands flexibility and auditability: we must be able both to respond to new demands and opportunities and also to account for our actions. At the same time, government-imposed efficiency gains [sic] reduce the resources available to implement change. Technology can provide the mechanism to meet these information needs, educational challenges and to manage changing social and political demands.

In this paper we offer pointers abstracted and generalised from our experience for organisational change in similar environments. We explain the development of a large web-based information system ([CSWeb](#)) at the Department of Computer Science at the University of Kent at Canterbury designed to meet these challenges. Our goal has been not only to improve our site but also to embed it at the heart of the department's culture. We describe the technical innovations required to manage this large-scale site and that permit the flexibility needed to respond to change. Finally we suggest measures by which the success of enterprises of this kind may be measured. We analyse quantitative data that not only includes page hit counts but also reports users' patterns of activity through the site. Staff acceptance of the system is also measured by the degree to which they have provided further content to the site. Evidence of changes in student and staff expectations, behaviour and study patterns is presented.

Motivation

In 1996 the Department's web presence had no coherent structure or content. In common with many other academic sites at that time, the majority of the content consisted of crudely converted paper documents. The site was sparse, inconsistent and dull: in particular, it projected a poor image both externally to prospective students and to our peers in other universities and internally to our students and other departments. There

was neither encouragement for improvement, nor mechanisms to do so. The Department had no guidance to authors, no tools and no policy in this area. As a consequence very few staff had provided any content although many more used the internet as a resource.

We believed that a high quality web site would be a crucial component of the Department's future. University departments with a strong research focus depend on attracting well qualified postgraduate students: there was evidence that the web was becoming a conduit for applications and we were convinced that this would increase in the future. Furthermore as the Internet penetrated both schools and homes, we believed that undergraduate recruitment too would depend heavily on our web presence.

We also believed that the web could be used to enhance the department's esteem amongst its peers. Universities in Britain are subject to increased scrutiny of both their teaching and their research. Although statistical data such as publication counts and research income inform judgements of research, we believed that our reputation would also be enhanced if our web site were to be perceived as a useful resource to the community.

Strategy

A departmental web site requires widespread support within the department if it is to continue to be vibrant and attractive; its content must be relevant to authors and users and kept up to date. Cultural change is only adopted if it is perceived to offer substantial benefits over previous practice, to meet real need and to be easy to adopt; additionally, technological solutions must look and feel good if they are to be taken up.

Thus at an early stage we concluded that such an enterprise would only be successful if it were embedded at the heart of the organisation's culture. It is necessary to win the support not only of 'early adopters' but also of the majority of the department; enthusiasts should be regarded as a valuable resource in gaining the attention of the many [1]. The immediate goals should be to attract the initial interest of staff and to maintain it. To cater adequately for all users (staff, current and prospective students and our peers) an information system solution should be based on separate and distinct units of real need.

Projects of this scale do not succeed unless stakeholders feel some ownership; in particular, maintenance should not be seen as yet another chore imposed from above/outside. A site should be packaged as something that belongs in part to each academic and that delivers something of value to each personally. It is important that there is some component that is identifiably 'theirs' (for example, their home page, the pages belonging to the courses that they teach, their research group's pages). Such pages should both offer real benefit (for example, to showcase their research, or to provide a convenient way of enhancing their teaching with additional support material) and should be sufficiently attractive that staff would be pleased to be associated with them.

Academics face increasing pressures on their time from all directions. If staff are to adopt a new site, then it must be simple to create new content. Ease of navigation and presentation of a competent image require that pages look good and exhibit some coherency and consistency: a *house style* should be maintained throughout the site. But pages should also meet authors' needs as they perceive them. These requirements may conflict. A solution should provide a simple to use mechanism that does not require authors explicitly to read and observe complicated guidelines (our solution does not even demand any knowledge of html), and should provide sufficient flexibility for authors to be

able to customise their pages within the framework of a house style.

Take up is better promoted by encouragement (implicit pressure) than by dictat. A sound guiding principle is that it should be easier to conform to a house style than to diverge from it. However, any successful system should be sufficiently flexible to allow customisation within the style, so that innovators are not stifled. Exemplars can be provided to stimulate the desire in individuals to improve their provision. The currency of information should be maintained with minimal human intervention. On the other hand use of the site can be promoted by making it the sole source of key information. Pressure from peers and students is a further effective motivator. The easiest way to share information is often to place it on the web in an accessible form and we find that staff are commonly asked by colleagues, research collaborators and students to provide information in this way.

Explicit pressure may be applied effectively in appropriate circumstances. This may be in the form of specific requirements to act (for example, to maintain core pages such as course descriptions) or as inducements with a reward. For example, one metric for research activity of our staff is derived solely from bibliographic data held in a publications database and presented on the web: only publication entries on the web are counted towards an academic's research activity.

As well as attracting and keeping the support of authors, a site should be relevant to users. Indeed, it should to come to be regarded as a core resource by all stakeholders. As well as being the sole source of key information, the site should be easy to use and reliable.

Consistent format and structure eases navigation. The system should save users' time in discovering and retrieving information (for example, reading, printing, copying to their computer). To prevent users drowning in a sea of information, not all of which is relevant to them, different views of a document (or a cluster of documents) should be provided for different categories of user. For us, the important distinction is between internal and external users -- a more sophisticated model was beyond the technology of the time. These views should be based on the principle that only the content relevant to, and the links available to, a particular class of user are offered thereby allowing pages to be focused on the needs of those users and removing any need for those all too common 'access denied' messages that both waste users' time and frustrate.

Accuracy and reliability are paramount if the site is to become embedded in the department's culture. Sites that are perceived to contain largely inaccurate and out-of-date information are soon abandoned. Not only should information be accurate, up-to-date and complete but, given our multiple views model, different views of a set of pages must be consistent. Information is most efficiently and most accurately checked by stakeholders themselves (for example, authors and students) rather than by those whose responsibility is limited to the maintenance of the site. Completeness of information is also improved by replacing clumsy and bulky paper documents by the web, thereby allowing easier distribution and updating of documents. Moreover these electronic documents should tailored to individual need as far as possible, thus improving the chance that they might be read rather than abandoned to the filing cabinet or the waste bin.

A further advantage of a web site is that it need not be restricted to static text. Other value-added services include improved communication such as automatic building of

mailing lists for course groups, and interfaces to other services. One such service was an interface to a system of active badges worn by staff, allowing such questions to be asked as is Dr X in their room? Are they alone or in a meeting?).

Specific provision

In this section we describe how we have incorporated our strategic aims into the development of our web-fronted information system. CSWeb is structured to support the core activities of the department: teaching, learning and research, and to reflect the lifecycles of different classes of document. Although it was clear that the site would play an important role in our quality assurance mechanisms and would improve the efficiency of our enterprise, the principle that administration should support rather than drive our activities was important.

Web business cards are provided for all stakeholders -- academic and support staff, postgraduate and undergraduate students. A business card is a personalised page containing uniform key information for that individual: the aim is to provide a single point of access to the site ('one stop shopping'). In the case of an academic or a research postgraduate it includes office and phone numbers, links to their publications and lists of the courses they teach and their tutees, all taken from departmental databases. Undergraduate business cards similarly contain lists of registered courses and class timetables. Staff and research postgraduates can further customise their cards to describe their research interests, to highlight particular publications or to include other information.

Tools simplify the customisation of pages within the department's house style and allow the easy maintenance of clusters of pages for our research groups (description, current projects, staff and postgraduates, events) and for the modules we teach (descriptions, classes, assessments). Tool support is described below.

Two views of the CSWeb site are provided and maintained in lock-step automatically: one for internal and one for external users. External browsers are directed by our server onto the external document graph; there are no links from this graph to the internal one, so no time is wasted attempting to follow links to inaccessible pages. Key data is culled from departmental and university databases and used to update CSWeb pages automatically (typically either nightly or weekly depending on the page).

We aim to move more and more material on to CSWeb. As well as publications, research group and course pages, we have placed student handbooks, past examination papers, FAQs, an anonymous question asking facility [4], minutes of Board of Studies and Staff-Student Liaison Committee meetings, teaching allocations and administrative responsibilities, job descriptions and departmental internal administrative notes (financial procedures, health and safety, and so on). Some of these may need further access control.

Engineering design

These requirements imposed substantial technical demands upon our information system. It was clear that no authoring or site maintenance tools available at that time met our needs. Indeed we are unaware of any current tools that would suffice. Any support system had to be flexible. Data held in the system would be put to multiple uses and it would be

both expensive and unreliable to re-key it. Furthermore, although we could identify our current requirements and even a number of ways in which these might develop in the future, we were well aware that we would require further uses of that data in ways that we could not foresee, whether due to changes in our *modus operandi* or due to external reporting requirements.

We also required the customisation of personal and other pages within the common format of a house style that would guarantee the consistency of the content and the quality of its representation. It was important to be able to change the format of all the pages on the site with ease. Furthermore it must be possible to render this data in different reporting formats for different purposes and offer different views to different audiences. The formats required might be html, simple ascii, or be designed as input for other tools such as spreadsheets or word processors. Again, it was important to be able to construct new formats to meet new requirements reasonably easily.

The data had to be timely: it would not be accurate if it were out of date. Our system had to ensure the rapid revision of pages. Furthermore it had to incorporate different clients' definitions of 'timely': some pages may be a day or even a week out of date (for example lists of research publications), other groups of pages must be updated immediately.

We required a degree of 'device independence'. We ourselves operated in a heterogeneous environment of different operating systems and different browsers. We had no control over the browsers that might be used by visitors to the site but we were aware that at least one target audience -- potential undergraduates -- might visit the site with comparatively old versions of browsers. Neither did we wish to restrict the reporting mechanisms to any particular format.

Finally, the site would be too large -- it currently comprises 13000 pages, including 3700 that are publicly accessible, using 1Gb of file store. Manual techniques, with or without the aid of available template-based authoring tools, would not scale to meet this challenge. In summary (as software engineers understand) we had to design for change. It is not surprising that no existing tools or systems met these requirements. Our solution included several innovations.

There were several reasons for making the site database driven. A large amount of data pertaining to staff and student details, and quantitative course data (what, when, registrations, assessment deadlines, etc) was already held in our departmental database [7]. Although much of this data is shared between different web pages, it has a single source--the database--and thus re-keying with its consequent expense and risk of errors is avoided. In addition, we can leverage existing mechanisms of access control and security of the data. To this database we added further tables holding information such as membership of research groups and publications data. As far as possible we avoid direct entry of data in favour of collecting that data from existing systems. Unfortunately it is not always possible to construct interfaces to the University administration's central databases (such as lecture timetabling).

Not all data is held in the database. The house style (template pages, images, buttons) and other bulk text items are held in the standard file store. Author-generated customising text is also held in their file store along with any pages that do not make use of data held in the database.

The scale and complexity of the system demanded that it be tool driven. Our database engine does not have the processing power to serve pages on-the-fly. Core pages are regularly and automatically updated using standard Unix utilities such as `make` and scheduling tools. This mechanism allows us to update different pages at appropriate frequencies, including where necessary on demand.

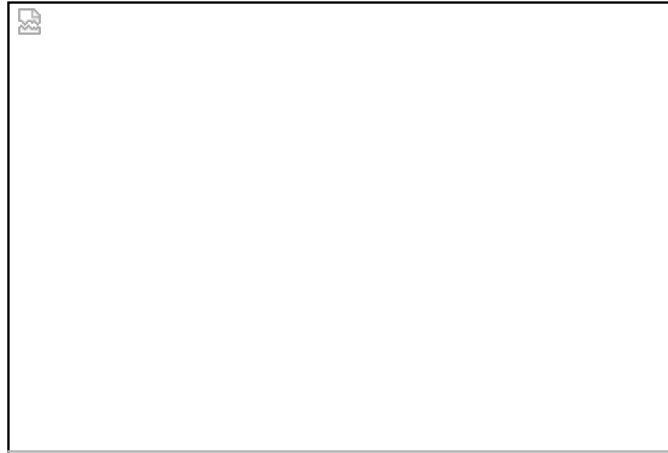


Figure 1: Building pages with active templates

Pages are generated using a set of tools and libraries written in perl [5]. Customisation of pages is achieved by processing template pages written in standard html in which a macro language [2] is embedded (see figure 1). These *active templates* offer a number of advantages.

- Each class of pages is generated from a single template. The template provides a consistent framework (enforcing the house style, easing navigation) and simplifies the construction of otherwise complex html (we make heavy use of tables to format pages). Furthermore, every page in this class can be restructured by changing just the template and then re-building that part of the site (with a single command, e.g. `make staffpages`).
- Construction of pages is aided by use of *include* files. As a tool processes the template, it replaces any include directive with the contents of that file in its output. This approach has two benefits. First, and most importantly, it provides simple customisation of pages without the need to make any change to a complicated template file (or indeed without any knowledge of html). Each author can customise their page by giving the relative pathname of a file that contains some content, html or plain text, that they want to include. Suppose the template for staff home pages contains a line

```
#include 'interests'
```

The tool will build an HTML page from the template, replacing this line by the contents of a file called `interests` in the current directory, wherever this file exists. The author simply needs to know the file names expected by the template and where in the output each include file will be inserted. On the other hand, this mechanism also simplifies the construction and maintenance of templates: common features can be included by an absolute pathname.

- Template files also provide an interface to the database. We provide different tools for

building staff and student business cards, module pages and research group pages. Each tool interrogates the database and makes an environment of (name, value) pairs available for use by the template (for example `$login` might hold the login name of a lecture or student). A value may also be a composite data structure such as a list of tuples and an `#foreachrow` construct is provided to iterate over the list. A typical example is to provide details of a list of courses taught by a member of staff. For example,

```
<H2>Courses I lecture on</H2>
<TABLE>
  #foreachrow @lecturecourses
    <TR><TD><A
      HREF="$module_url">$course_code</A> </TD>
      <TD>$course_title</TD></TR>
  #endforeachrow
</TABLE>
```

might produce

Courses I lecture on	
CO311	Functional Programming and Logic
CO500	Machine Languages and Compiling Techniques

A simple conditional clause is provided to test whether variables are set. In the example above, it would be sensible to test whether `lecturecourses` was empty before generating the 'Courses I lecture on' section of the page.

- The macro language also provides an interface to the tool's context either through predefined variables (such as the URL of the page being created), command-line arguments given to a tool or by allowing arbitrary programs to be executed and their output captured. The most common use of the latter facility is to date stamp pages with their modification date.
- Templates allow us to provide different views for different audiences: currently we instantiate two - local and external. Sections of the template may be marked as for internal/external sight only. From a single template file and its associated include files we generate separate local and external pages. No content from an internal section will appear in the external page, nor will content marked external appear in a local page. The tools support the shadow tree structure by generating links to either `.local` or `.html` versions of pages. This is achieved by using the `local_path` variable as the extension in a URL: thus a hyperlink in a template to `index.@local_path` would generate links to `index.local` and `index.html` respectively in the internal and external versions of the page.

We believe that this mechanism -- generating consistent pages -- is a sounder solution than attempting to 'fix' errors later [3]. Our solution places responsibility on the author rather than the web master; it is more flexible as it does not require graphs of pages to be isomorphic; and at the page level, it supports consistency of *content* rather than simply checking for identical sequences of tags.

- Buttons and banners are time consuming to create but useful to provide. Our house

style defines the format to be used for buttons and side-bar banners but does not resolve the problem of how authors can create new ones. We provide mechanisms to solve both these problems. Images for use as side banners can be created with a `gimp` script [8]: authors have only to provide the text that the banner is to contain. Buttons are created with a similar tool: again the author need only provide the text to label the button--in general `mkbutton` can render this label in any format on a button of any shape and we currently provide two co-ordinated shapes. To further assist author, the template language's `#button` command places a row of buttons on a page including the url, image and alternate text of each button.

One of the earliest goals of this project was to place bibliographic records and, wherever copyright permitted, the full text of staff publications on the web. However we recognised that this data was required for multiple purposes: for the web, for the University's annual Research Report and for our return to the funding council's 4-5 yearly Research Assessment Exercise. Each of these needs demanded a different reporting format. Our solution was to offer a web form for staff to submit details of their publications to our publications database. These forms and the database tables were modelled on `bibtex` [6], a widely used bibliography format. One strength of `bibtex` is its flexibility. It is possible to define new document types in addition to the standard article, book, etc types; within each document type new attribute fields can be created in addition to the standard author, title, year... fields; and finally the choice of these document types and fields to be extracted and the design of their format can be very largely controlled through the use of style files. By defining new attributes, we can link publications to relevant staff, postgraduate and research group pages. By providing different style files, we can extract bibliographic reports from the database in almost any format required.

A motivation for this project was to support our recruitment strategy. As well as allowing unguided access to our site, we also wished to provide a more structured tour of web pages relevant to particular target audiences (such as potential undergraduates, potential postgraduates, etc). A tool for the simple creation of *tours* has been provided. A tour is a visit to a pre-set collection of pages within the site. At each stop the user is presented with the page, a commentary on that page, and a navigation bar not only allowing movement to future or previous stops within the tour but also indicating their progress. One use of tours might be to provide a 'virtual Open Day' for prospective students. The tours, and hence the commentaries, are independent of the pages visited thereby allowing different tours to share the same pages but to provide different commentaries.

Evaluation

In this section we suggest measures by which the success of enterprises of this kind may be measured. Our goals were to provide a flexible, easily maintained information system that enhanced the department's research profile, supported teaching and learning and facilitated administration. Our strategy dictated that CSWeb be embedded at the heart of the department's culture. In this section we explain how we analysed the success of the project and measured its take-up, and we summarise our findings.

We distinguish the provision of content of the site from its use. Staff acceptance of the system is measured in part through the degree to which they have added material or customised their parts of the site rather than relying solely on automatic provision of pages. Conversely, the degree to which support material is not provided in other forms

(notes, other file store) also provides a measure of the dominance of CSWeb. Staff, as well as students and external visitors, are also users of the site. We wished to measure how different categories of user interacted with the site. Such an evaluation cannot be done properly by a simplistic analysis of page `hits' that does not track the users' activity through the site. One activity that we wished to measure was the route individuals took through the site.

Pages provided automatically or by secretarial staff (for example, programme handbooks and module descriptions) provide adequate, if minimal, content. To discover the extent to which CSWeb had been adopted by staff we measured the degree to which staff had customised these pages and how much additional content had been supplied.

- Of 61 modules offered, 46 added further material other than the course description (which is mandatory and in any case done by departmental support staff). The most popular use of CSWeb was for support materials (e.g. lecture notes, other resources) and assessment descriptions. Of the remaining modules, 7 were modules offered to non-specialists: we conjecture that these students had lesser expectations of finding material on CSWeb.
- More than half the research active academic staff had customised their business cards. As expected, innovation follows well-known paths [9]: there is a correlation between customisation of an academic's business cards and of his/her course pages.
- All research groups have used of the web to present their work though some customisation hooks have never been used.

<

User activity was tracked by enhancing the standard web logging through *cookies* and recording the reference information given by the browser, allowing us to save the location of whence the user had come to a particular page. The enhanced logging data was collected for a term (October to December 1998). It was filtered off-line to ensure that it consisted solely of access to web pages (no image files, style sheets, nor Java classes). Requests caused by local robots gathering pages for our search engine were also removed.

Each log entry was then considered as a source and destination url pair and the use of each pair was integrated over the recording period to discover the most popular `clicks' between pages. These arcs were then used to construct graphs of the popular routes around the site and to evaluate the most popular source and destination urls (see Figure 2). The most popular source urls indicate those that are most likely to be bookmarked.



Figure 2. Hyperlinks by traffic volume

From a detailed analysis of the user activity we concluded:

- Staff and students use CSWeb extensively. The logs recorded 1.2M accesses (120K accesses per week). We split the requests into user categories by using host names and by knowing which hosts were used by staff and students. This gave us 490K external accesses, 60K by staff and 675K by students: average academic and student accesses per week were 160 and 140 respectively.
- Students bookmark their course programme, modules and their business cards.
- Students use module home pages extensively, browse to find staff, search and use the anonymous feedback forms.
- <
- The more popular modules tend to be core first-year teaching modules. We conjecture that there are two reasons for this. First, newer students are more web-savvy. Second, as CSWeb has matured, it has become the prime source of information so first-year students are less used to other resources whereas second and third years have developed other mechanisms for acquiring information. Third, the pattern of first-year teaching -- with many small assignments -- encourages greater use of the web whereas later courses do not require provision of such a quantity of material.
- Certain groups of students (for example conversion MSc computer scientists) use the site particularly extensively, judging by the heavy traffic from their course programme page.
- The most popular route through the site for students is from the home page to the general teaching page, to the Computer Science Programme and then to an anonymous questionnaire page [4] of a first-year module.
- For general external users, graphs of the site centred on various research activities were popular especially when they were support pages for books published by our staff. However most users started from the home page and then took one of three

different routes -- looking for people, our research or our teaching (including prospectuses for undergraduate and taught postgraduate programmes).

Conclusion

In conclusion, we have built a large web site that is relatively easy to maintain. The site is supported by a number of technical innovations. It is largely database driven (which was unusual at the time it was developed); a context-aware template language provides consistency, easy customisation and separate views of the site for different categories of user; and flexible reporting formats allow comparatively easy response to new challenges.

Our approach of encouragement through ownership, ease of maintenance and peer pressure rather than command from on high has proved successful. There has been widespread take-up by both staff and students, and innovative use of tools to track user activity has revealed interesting results. Planned developments include further automation, dynamic generation of pages, interfaces to other university databases (such as timetabling), better tool support to give users greater ownership of their own data, and simple data mining tools.

Finally we would like to thank Andy Peel for implementing much of CSWeb.

References

1. Everett M. Rogers, *Diffusions of Innovation*, Fourth Edition, Free Press, 1995
2. Andrew Peel, [*HTML Macros -- Easing the Construction and Maintenance of Web Texts*](#), Technical Report 4-96, Computing Laboratory, University of Kent, January 1996.
3. Shueh-Cheng Hu and Richard Furuta, *A Tool for Maintaining Multi-variant Hypertext Documents*, In Jacques André, editor, *Electronic Publishing 98*. Springer, May 1998.
4. David Barnes, [*Students Asking Questions: Facilitating Questioning Aids Understanding and Enhances Software Engineering Skills*](#), ACM SIGCSE Bulletin, 29(4), pp. 38-41, December 1997.
5. Larry Wall, *Programming Perl*, O'Reilly and Associates, Inc., 1996.
6. Leslie Lamport, *LATEX Users' Guide and Reference Manual*, Addison-Wesley, 1986.
7. [*The Ingres RDBMS*](#).
8. [*The GNU Image Manipulation Program*](#).