

The NPRIE Website – A Glance behind the Scenes

Steve Burgess (Library Information Manager, West Yorkshire Probation Board, Website Manager, NPRIE)

An account of a purpose-built content management system for the National Probation Research and Information Exchange (NPRIE).

Introduction

NPRIE is the National Probation Research and Information Exchange. It provides a supportive network to library, information and research staff in the 42 probation areas in England and Wales. It also advises the National Probation Directorate and NOMS on information and research issues. Although the network is focused around information and research, all probation staff may join. Members of other criminal justice agencies, and those contracted to work with them, may join the network as associate members.

The main work of NPRIE is organised into these areas.

- Information and Library Management
- Research
- Data Standards
- Communications and Public Relations

Members of the committee also sit on a number of national committees, groups and boards. These including various user groups for information management systems (e.g. CRAMS, c-NOMIS, eOASys, IAPS etc – see the Glossary of Acronyms and Abbreviations on the NPRIE website to find out what these mean!), the Information Advisory Board, and the national Data Standards and Reporting Group.

Although NPRIE is “staffed” entirely by volunteers, and neither has nor attempts to raise any funds, it is a very active group that produces a wide range of outputs. Most of the information products NPRIE produces can be accessed now on the website. The exchange also organises conferences. The next of these will be in York in October (more info on the website).

Outputs include:

- The Probation Thesaurus – produced by the Information and Library Management Group (ILMG)
- The Probation Gateway (ILMG)
- The Glossary of Acronyms and Abbreviations (ILMG)
- The National Research Collection Catalogue (ILMG)
- The Stakeholder Survey Guide – produced by the Research Officers’ Group (ROG)

As you can see, in spite of its size, the Information and Library Management Group is very active.

As with many organisations, especially ones with a relatively sparse population widely distributed around the country, communication is one of the biggest challenges. NPRIE used to produce a regular newsletter but changes in staff, changes in the organisational structure of probation and its priorities meant that this appeared less and less frequently. What was needed was a mechanism to allow:

1. members of NPRIE to keep in contact with one another easily
2. the special interest groups to coordinate their work
3. the committee to keep members informed about developments
4. for general information and NPRIE information products to be stored and disseminated

In 2002 I was co-opted onto the committee to take on the redevelopment of the website – which previously had functioned as a signpost to NPRIE services rather than a means of delivering them.

My main objective in the redevelopment was to create something that would allow the editors to update the content of the website without using me as an intermediary. Like most other people in NPRIE, I have a full time job (Library Information Manager at West Yorkshire Probation Board) and although my employer allowed me to do the website development work in work time, I did see it as being a time bound project which would have an end point. Effectively, then, what was required was a content management system. As we had no funds – I decided to write my own.

My first task was to take the content from the existing static pages and put them into a database and then create a template which would pull the relevant data from the database to create the full HTML pages. Of course, most web pages are made up of more than just text – and even though the editable part of the page would be limited, I did not want the editors to be restricted to only being able to represent information in paragraph form. They would need to be able to use bullet lists, tables, images etc. How

would I allow editors to create this sort of formatting – without giving them a free hand in the use of fonts, colour and design.

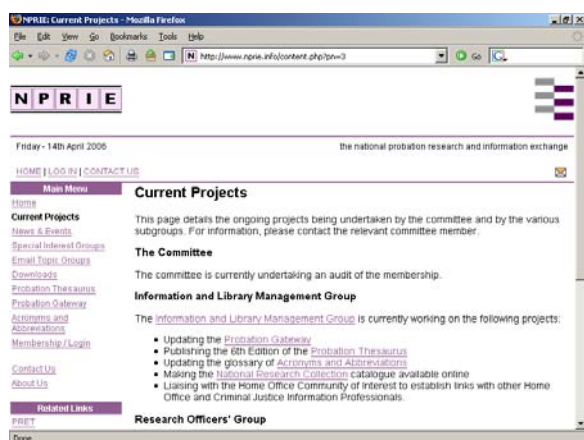


Figure 1

My solution was to regard each different type of content as a separate element on a page. So, a paragraph would have one entry in a database, a bullet list another, a table another etc. A relational link (one-to-many) between the PAGES table and the PAGE_ELEMENTS table would make this all work. Figure 1 is a screen shot of the website.

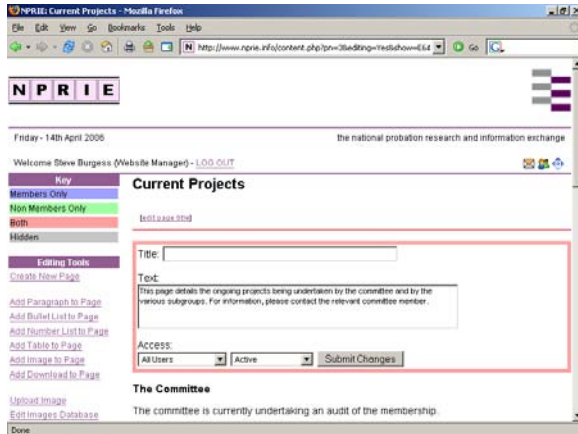


Figure 2

Figure 2 is a screen shot of the same page in editing mode.

As you can see, the user edits each element on the page separately – so one page is built up from many different elements of different types. Each element may have a title and each element can be switched on and switched off at will. Equally, it is possible, on the same page, to have items that are visible to general users only, to logged in members only or to all users. This approach ensures that:

- editors cannot apply their own individual style or “creative panache” to pages
- bullets, tables, paragraph titles etc look the same throughout the site
- the same page can be used to hide some information from non members

To edit the more complex page elements (such as bullets or tables) the editor simply needs to define how many sub elements there are (i.e. how many bullets; how many columns and rows) and define any special features (i.e. is the bullet list an ordered or unordered list; are the first rows and columns in the table headings?).

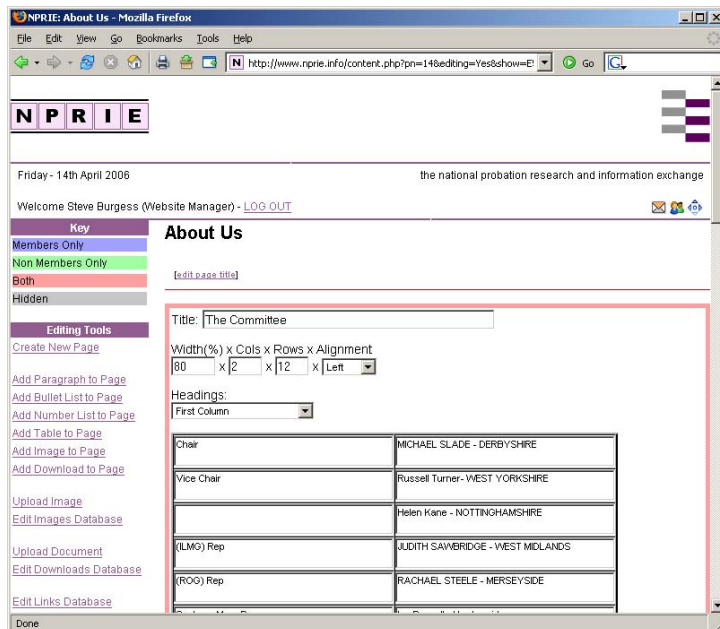


Figure 3

Figure 3 demonstrates this when editing a table.

This table is 80% of the available width, has 2 columns, 12 rows and is aligned to the left. The first column contains headings. Each of the cells in the table is now part of a form that the user can edit.

All of this information is stored in a MySQL database table on the server. The following “element types” are permitted:

- Paragraph
- Bullet List
- Numbered List
- Table
- Newline (not used)
- Downloads (there is a database of downloadable files – more on this later)
- Image (there is a database of images)
- Form¹
- Server Side Includes* (some pages use server side includes to allow more complex interaction with the database – an example of this is the Probation Gateway page)

The editor can change the order of elements on the page and hide them (but not delete them). A different script allows editors to actually add new pages and to tinker around with the menu system (the coding of which became so complex I’m not altogether sure I can remember how it works!).

Links

Of course, web pages are NOT solitary pieces of information and the great thing about websites is ones ability to link to other web pages either within the site or on other websites. Although editors could edit and add new content to the existing pages – they couldn’t, at this stage, make links to other pages or to other websites.

What I didn’t want to do was to have to teach them how to create <A HREF... tags within the paragraphs/tables/bullet lists they were creating. This wasn’t out of a desire to prevent them from acquiring the knowledge – but rather was more about ensuring a consistent approach to the creation of the tags (as the site is HTML 4 compliant). Also, each of the pages within the site are actually the same page (content.php) and the content you actually see is defined by a variable passed in the URL called pn (unsurprisingly this means “page number”). I wanted to eliminate the possibility that editors might get the URL wrong when creating links. The solution was to create a table in this ever-growing database containing information about ALL links that could be used on the site. Editors are not able to create a link unless they first add the information to this table.

The table has these fields:

LinkName
LinkAddress
Internal
IntPageNum (for internal links)

¹ indicates elements that, at present, only I can use as they both require interaction with PHP scripts and I felt disinclined to let editors loose on the server in this way.

All the links for all the pages on the site were added to this table – and, indeed, when the user adds a new page to the site the link information is automatically added.

So now when a user wants to create a link to a page on the site, they simply have to write the name of the link (LinkName) in their paragraph or bullet list or table in uppercase. A text processing script scans the text before displaying it on the page and replaces the capitalised text with a link and then naturalises the text. So, for example, on the home page the introductory text would have been entered as:

“NPRIE advises and supports members, agencies and other organisations in the development, use and provision of research and information in the pursuit of NATIONAL PROBATION SERVICE objectives and towards...”

The text processing script would have output:

“NPRIE advises and supports members, agencies and other organisations in the development, use and provision of research and information in the pursuit of National Probation Service objectives and towards...”

so when that paragraph is displayed on the home page, the capitalisation has gone and the link to the National Probation Service website is in place.

I did consider not requiring the user to explicitly define the link by putting it in upper case – but I personally find websites where every mention of, for example, the National Probation Service would end up being a link a little bit annoying. It interrupts the reading flow of the page.

Uploads

One of the main functions of the website is a repository of NPRIE publications – editors needed to be able to upload files to the webserver. So the next development was to:

- allow editors to upload files and images to the server using a form (see Figure 4)
- allow editors to make links to these files in a consistent way

Figure 4

It was also necessary to ensure that those documents that were only available to members could not be accessed by entering the full URL of the document in the browser – but in a way that fitted in with the existing authentication scripts.

The decision was to put ALL documents into a in the root of the (virtual) webserver. This meant that documents could not be accessed directly by the browser. Documents could only be accessed by a script.

When an editor wishes to make a document available on the website they add a “download” page element to the page and edit it using the form in Figure 5.

The screenshot shows a web editor interface. At the top, there is a 'Title:' field. Below it, a message says 'Please select a download from the Downloads Database:' followed by a dropdown menu currently showing 'NPRIE Annual Report 2005'. Underneath, there are icons for file types and a text label 'NPRIE Annual Report 2005'. An 'Access:' section contains two dropdown menus, one set to 'All Users' and another to 'Active', along with a 'Submit Changes' button. At the bottom, there are more icons and a text label 'List of RDS Training Courses for R&I staff'. Small links for editing and moving elements are visible at the very bottom.

Figure 5

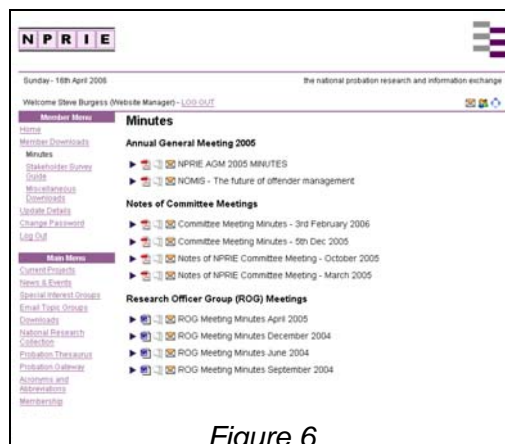


Figure 6

The editor can specify a section title for the download if required and then simply chooses the download using the drop down list. This list is fed from a database of all documents that have been uploaded to the site. The images are created automatically on the basis of the file type, whether a zip version of the file is available and whether the file is of a size that it would be appropriate to e-mail. The little triangle is a toggle button which reveals the bibliographic information about that particular file.

The page shown in Figure 6 shows how downloads can be grouped using Titles for the first download in the group and not for subsequent ones.

Uploading Images works in very much the same way – see Figure 7 – but this time the editor can specify whether an image is left, right or centre aligned; and whether it is an inline a newline image. For example, specifying an inline, right aligned image will mean any subsequent text will wrap to the left.

Editors cannot change the representational size of an image and must specify alternative text for standards compliance.

The screenshot shows a web editor interface for adding an image. It features a 'Title:' field with 'Committee 2005' entered. Below is a message: 'Please select an image from the Image Database (a popup will appear with a preview so you can see how the image will appear on the page)'. A dropdown menu shows 'The 2005 Committee Photograph'. There are 'Alignment Options' with buttons for 'Centre', 'Left', and 'Right'. A 'Link (optional):' field is present. At the bottom, there is an 'Access:' section with dropdowns for 'All Users' and 'Not Active', and a 'Submit Changes' button. A preview of a group photograph is shown in the center.

Figure 7

Server Side Includes

The result of all of the above work enabled the editors to keep pages up to date. The website now fulfils their needs to keep the membership informed about developments, new events and new products. They can publish sensitive information to the membership without making it available to general users. Prospective members can apply to join NPRIE and the entire membership process is automated through the website. That said, however, the website still isn't a dynamic site as such. The creation of pages, while easier than before, is still a relatively labour intensive process. I had a need to put dynamic content on the site (such as the National Research Collection and the Probation Gateway) in such a way that fitted in with the page/page element model. I realised that I

could create a page element called Server Side Includes which would just contain a reference to a scripting file on the server.

This allowed me to add the following dynamic features to the site

- The news feed (powered by My-RSS.co.uk) on the home page
- The National Research Collection searchable database (only available to members)
- The Probation Gateway
- The Glossary of Acronyms and Abbreviations

This means that a change to any of the databases above is reflected immediately on the website. The ability to add server side includes means that the website can be developed in relatively sophisticated ways – while retaining the editing system and page description model as originally implemented.

I have moved on from this project now and am working on other projects (i.e. the redesign of the West Yorkshire Probation website – to be launched in the summer – and some private projects of my own). I think the NPRIE web site could do with a design overhaul and the menu system could do with being improved – but one could tinker around forever and the site, as it stands, is functional and serviceable by non-technical users which was the objective I set myself at the very beginning.