Combining available standards and tools to build a compliance oriented website management system

Neeraj Arora
School of Computer Science and Information Technology
RMIT University
ASSETT Research Group
Melbourne, Victoria, Australia
narora@narora.net,
nearora@cs.rmit.edu.au

Linda Pannan
Teaching Fellow (Online Learning)
Teaching and Learning Portfolio
RMIT University
ASSETT Research Group
Melbourne, Victoria, Australia
linda.pannan@rmit.edu.au

ABSTRACT

Analyses of representative organisational structures and their implications for management of organisations' websites reveals that the various roles of developers and users of the system and content are of prime importance. Typically, policy and mandated procedures are used within organisations to enforce standards that deliver compliance, usability, and accessibility. A poorly presented organisational website resulting from any shortfalls in implementing the required processes and possible misinterpretations of the procedures not only reflects badly on the organisation and its ability to manage other business activity well, but also creates a lot of additional work for the system administrators.

This paper describes the design of a system to manage organisational websites created by using a combination of tools available in the public domain. The system facilitates standards compliance, and by providing for the enforcement of good design it also addresses usability and accessibility. An example implementation is discussed as it assists in understanding the design. We conclude by identifying the work needed to further improve the proposed system design and increase its functionality.

1. INTRODUCTION

Business corporations have embraced the Web as an effective tool for communication and marketing for some years now. As public funding for post secondary education declines further and globalisation threatens their market share [12], universities, also, have more recently invested much effort in the development, standardisation, and use of their websites for similar purposes as they adopt economic rationalist management approaches in seeking to reduce costs, and boost revenues.

Standards compliance refers to websites being conformant to recommendations provided from time to time by the World Wide Web Consortium (W3C) [36]. The W3C provides recommendations for markup formats used to present information and styling, such as XML Schema [30], (Extensible) Hyper Text Markup Language ((x)HTML) [35], Cascading Style Sheets (CSS) [34], Extensible Markup Language (XML) [25], Extensible Stylesheet Language (XSL) [26], and XSL Transformations (XSLT) [5].

Hull [14] defines accessibility in terms of a website that is capable of being reached, and that the information within it is easily obtainable, extending well beyond the definition that simply provides for the needs of people with disabilities. He contends that to achieve accessibility it is important that, first, websites be accessible to the technologies, for example through consistent markup and styling so that components of the content may be found by electronic means, and second, to people. To achieve this a website needs to be standards compliant and needs to follow commonly accepted guidelines, facilitating its accessibility. Nielsen [20] defines usability as a quality attribute. He considers it comprises the learnability of basic tasks when a website design is first encountered, the efficiency of performing these tasks, the memorability of performing these tasks using a website design, the errors made while performing them, and satisfaction gained by users of a website. The design of a website directly influences these features, and we can conclude that the design of a website directly influences its usability. Further, the quality of a website depends on a combination of its standards compliance, accessibility, and usability.

Nielsen and Coyne [21], Nielsen et al. [22], and Nielsen and Norman [23] have shown that investment in accessibility and usability of organisational websites yields returns in the form of profit. Although they target only commercial organisations in their articles these findings are equally applicable to universities and their website developments as they aim to increase their revenue through increased visibility and effective market exposure for their programs and services, in attempts to attract more students.

The realisation of the importance of standards compliance, usability, accessibility, and the management practices to support these, in producing a quality website, is evidenced in that many organisations have migrated their websites to new structured systems [2, 3, 6, 27]. Gupta et al. [11] promoted business use of Content Management Systems (CMSs) in 2001 and, since then, many businesses have installed content management strategies, systems, or frameworks. A CMS is based on a set of organisational policies and guidelines, and is enforced by software, or workflow models, and mandated or standard operating procedures. Although it may be supported by software an off-the-shelf, standard CMS re-

Appears in *Proceedings of The Eleventh Australian World Wide Web Conference, Gold Coast, Queensland, Australia,* ISBN 0-9751644-3-0, pages 20 - 37, July 2005.

lies on strict business practices to enforce adherence to policies and guidelines to control the content, presentation, and design of a website. Further, its effectiveness in providing standards compliance relies on the awareness of accessible and usable web design practices amongst staff providing and authoring web content, adequate technical expertise in staff dealing with the websites, as well as the reliability of tools used for checking standards compliance, accessibility, and usability.

Unfortunately, there are reported shortages of usability experts [18] and web authoring tools are often not compliant to the latest standards [38]. Hence, a method of enforcing compliance in website production, and the ability to monitor it, that does not rely on the presence of highly skilled website management personnel would minimise the need for usability expertise and potentially solve the problem of inadequate website compliance, accessibility and usability resulting from the current gap in availability of such expertise.

In this paper we present a design for a system that is based on the premise that standards compliance facilitates website usability and accessibility, as reported by Hull [14], and Patterson and Ellis [24]. An initial consideration of the background on current CMSs and the needs of organisations leads into a description of our website management system design for which we then provide an example implementation. The merits and shortfalls of the proposed system are discussed, culminating in the presentation of concluding remarks.

2. BACKGROUND

In many ways university and corporation operations are converging as universities take on greater responsibility for creating their own revenue. For the purpose of this study we will consider universities as a representative class of organisation that uses the Web extensively for communication and marketing purposes. Our choice is guided by the fact that surveys and reports are readily available for university websites but not for corporate websites in Australia.

Any system for delivering standards compliant, accessible, and usable websites must address the needs of contemporary organisations and their processes. Background on organisational structure and the roles fulfilled by its personnel, on the use of CMSs in organisations and their effectiveness in meeting organisational needs, and the consequent status of organisational websites today is presented here to provide a context for our proposed website management system design.

2.1 Current status of organisational website compliance

Patterson and Ellis [24] conducted an analysis of the results of 134 user tests that assessed 44 Australian university websites. They identified a relationship between website usability and accessibility, and identified several website characteristics and website management practices of individual universities that influenced their websites' usability. When Alexander conducted a study of Australian university websites in 2003 [1] it was found that 98% of them were inaccessible, and that although policies were in place for governing web accessibility there were no procedures implemented to ensure that compliance with accessibility standards was achieved. Further, there was insufficient knowledge of accessible web design practices amongst staff dealing with the websites. An earlier similar study of eleven UK higher education websites by Sloan et al. [28] in 1999 determined that the content authors of these systems needed to be technically aware and educated if they were to understand the details of style guides, templates, and tools provided for checking standards compliance, accessibility, and usability. In general, this was not the case and Sloan et al. stress that content authors do need an awareness of the standards so that they can, at a minimum, understand feedback from testing tools used to detect accessibility and usability problems that may appear in the resources they produce. Nichols and Twidale [18], in their study on usability of open-source software, also report that there are not enough usability experts to participate in projects to control their usability and accessibility.

Although web authoring tools might be seen as a potential part-solution to this reported lack of expertise in ensuring standards compliance, when Vaughan-Nichols [38] investigated the extent of standards compliance of major web authoring tools used to author websites they found that none of them were fully compliant to the latest standards. The literature cites leading developers of such tools stating that compliance of authoring tools is often delayed and will remain so into the future. This increases the demand for technically educated content authors as the web authoring tools cannot be relied on to produce standards compliant web resources.

2.2 Use of Web Content Management Systems in organisations

We define a Web Content Management System (WebCMS) as the combination of business practices, that are used to enforce policies and encourage adherence to guidelines in order to control the content, presentation, and design of a website, and the software systems and workflow models that support these practices. This is depicted in Figure 1, where a WebCMS is represented as a system that combines the operation of a CMS and website management processes. Each of these components are influenced by both business policies and guidelines and, in co-operation, they produce an organisational website, presenting a web view of the organisation to the End user. Effectiveness of the WebCMS in producing standards compliant web-pages and an accessible, usable website relies on the simplicity of use of the system, its compliance to standards and its integration into the workflows of the organisation.

Lombardi's 2003 survey [17] of consultants advising on or working with WebCMSs does not predict a positive outlook for these systems. Cost, customisation, migration, flexibility, and the ability to evaluate were highlighted as being problematic before and during implementation of a WebCMS by about 50% of respondents; a further 30% added additional problems as being those associated with the ability to integrate with existing software, implementation time, complexity, and inappropriate business workflow. Simplicity and standards compliance were among the many features that implementors wished for in a WebCMS [32].

Joyce [15] summarised a report released by Jupiter Research in 2003 that investigated reasons for the failure of WebCMSs. Complexity, exorbitant technical maintenance costs, and over-engineered site infrastructures were highlighted as some of the problems. Sixty-one of the 100 companies surveyed still relied on manual processes to update their websites, and 27 companies planned to build their systems up from scratch.

It is evident that investment in a commercial WebCMS has never been an easy decision for an organisation. An alternative might be to use open-source WebCMSs. However, Nichols and Twidale [18] highlight problems with usability and accessibility of open-source software in general. The Commonwealth Scientific and Industrial Research Organisation (CSIRO) [7] evaluated a few open-source WebCMSs [41] and found that only Zope [37] was partially suitable for their needs.

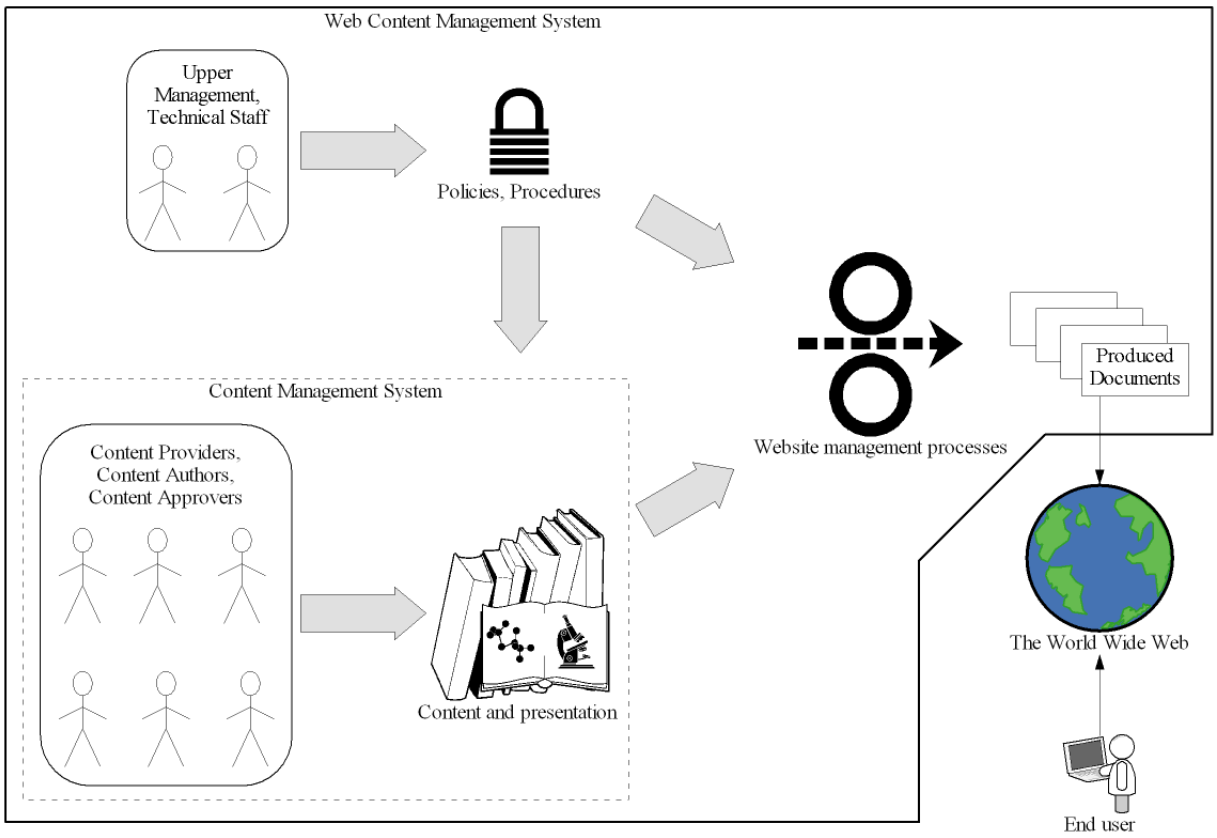

Figure 1: Production of an organisational website

2.3 Roles relating to websites in organisations

Organisations comprise a diverse set of people who work as a team to achieve a goal by dividing tasks between individuals, smaller teams, or groupings. We call a smaller team or grouping a 'unit' and their specific task-set in achieving a goal their 'role'. The range of tasks performed in managing an organisational website is well reported in the literature [6, 11, 40]. Figure 2 represents the generalised roles that exist within an organisation to create and maintain an organisational website. They are:

- *Organisation*
  The entity which is represented by an organisational website.
  - *Strategic Decision Level*
    This comprises roles that have responsibility for, and ultimate authority to make decisions impacting on the organisation's website, affecting the entire organisation. Roles in this category may consult other roles, who participate to create and maintain the website, before making decisions.
    - *Upper Management*
      Oversees the purpose, goals, and processes that govern the website.
    - *Committees*
      Generally comprise several experts in various aspects of the operation of the organisation.
  - *Business Unit*
    This comprises roles which are affiliated to each other with respect to the business of an organisation. They are logically bound in a team within the organisation, such as a department, a regional division, or a faculty.
    - *Management*
      Oversees the operation of the website components that represent the Business Unit.
    - *Content Providers*
      Deliver content to be placed within the Business Unit components of the website.
    - *Content Authors*
      Manipulate content into a format acceptable to the system used to publish content into the Business Unit components of the website.
    - *Content Approvers*
      Review the content published within the Business Unit components of the website.
  - *Technical Staff*
    Roles in this category provide technical services to maintain software and hardware required to implement an organisational website, and may provide support to other roles to achieve this.
    - *System Administrators*
      Commission and maintain the software or hardware infrastructure that forms the base for implementation of the website .
    - *Technical Support*
      Provide support services for software and hardware used to implement the website.

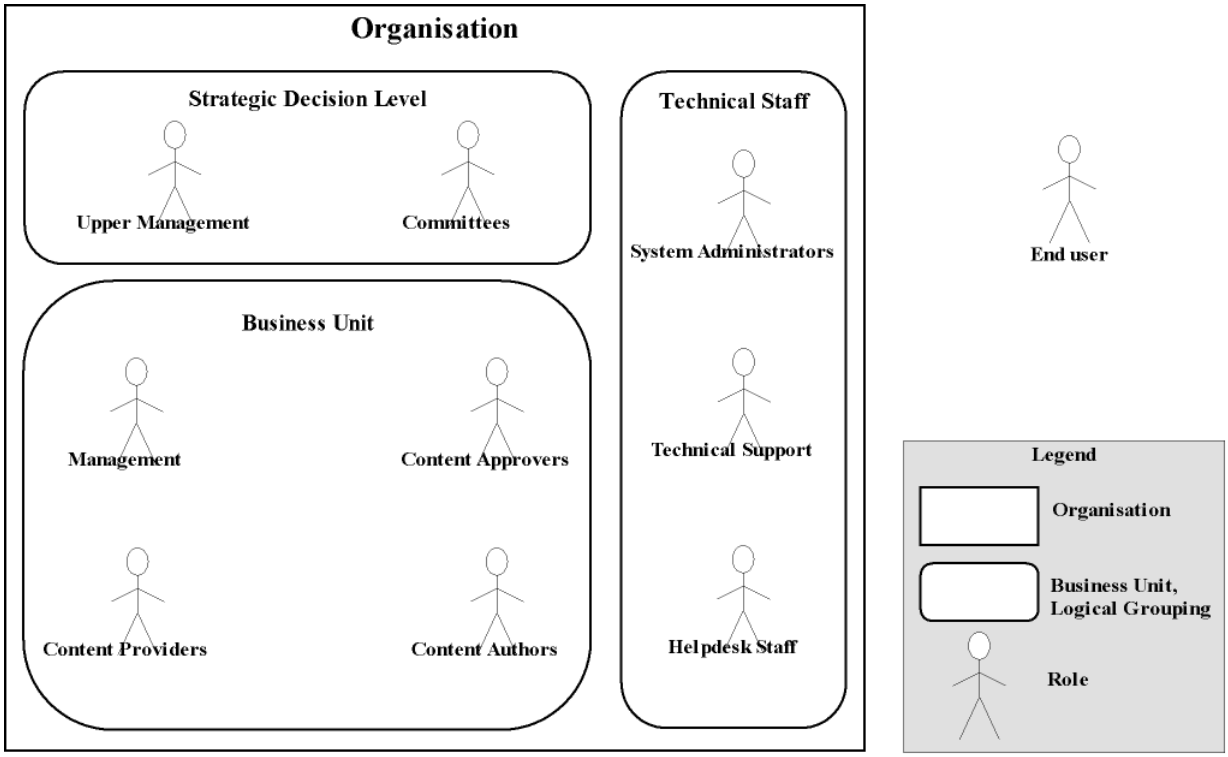

Figure 2: Roles in an organisation that relate to creation and maintenance of an organisational website

  - ∗ *Helpdesk Staff*
    Provide first level support to other roles participating in creation and maintenance of the website.

- *End users*
  A role that is related to the organisation as a current or potential client, customer, or employee of the organisation. They approach the organisational website to obtain information or to use services provided by the organisation.

Business Units and Logical Groupings are formed on the basis of the affiliation of roles which comprise them. We do not assume any restriction on the number of people or business entities required to fulfill a role, nor do we assume any restriction on the number of roles a person or business entity can fulfill. Interactions between roles can vary from one organisation to another and these have, therefore, not been generalised.

As an example, Gupta et al. [11] analysed three content management approaches, and referred to the roles used in their operation: the Centralised Approach, where all content is handled by a single group; the Distributed Approach, where a number of smaller groups perform roles as mentioned in Figure 2; and the Hybrid Approach that exhibits the Centralised Approach's central group controlling policies and procedures, and the Distributed Approach's smaller groups contributing content. It was found that discrepancies in interpretation of business rules by the groups, and limitations of procedures between groups could cause a drawback of the distributed aspect of the Distributed and Hybrid approaches.

3. A WEBSITE MANAGEMENT SYSTEM DESIGN

While the best way to achieve compliance with various standards is to have technically competent content authors, they are often not available in reality and a systems solution to managing content, and website presentation of it, becomes a necessity for organisations. Although many organisations have turned to WebCMSs to find a solution, research and surveys have shown that none of the WebCMSs studied are readily configurable, scalable, and maintainable. Commercial WebCMSs are costly, while their open-source relatives require a lot of customisation. Further, most WebCMSs are not completely up-to-date with the more recently introduced standards recommended by the W3C, such as the markup, styling, and templating standards.

Considerations of cost and maintainability prompted us to investigate building a system to manage an organisational website from existing standards compliant software that is available in the public domain. Our aim was to develop a system that produces standards compliant output and provides central control of website design in order to facilitate the implementation of usable and accessible design. We have attempted to design a system that allows compliance to be achieved with a minimal feature set in the first instance, and that enables the addition of new features as necessary. Hence, our main system design aims are to keep the separation of presentation and content, use as many existing tools as possible, convert workflow processes into software processes, and remove the requirements of interactions between organisational roles from different logical groupings.

Figure 3 depicts graphically the proposed system design. The design integrates with the roles model of an organisation (see Figure 2) and fulfils all software, business practices and workflow processes undertaken in the *website management processes* of current

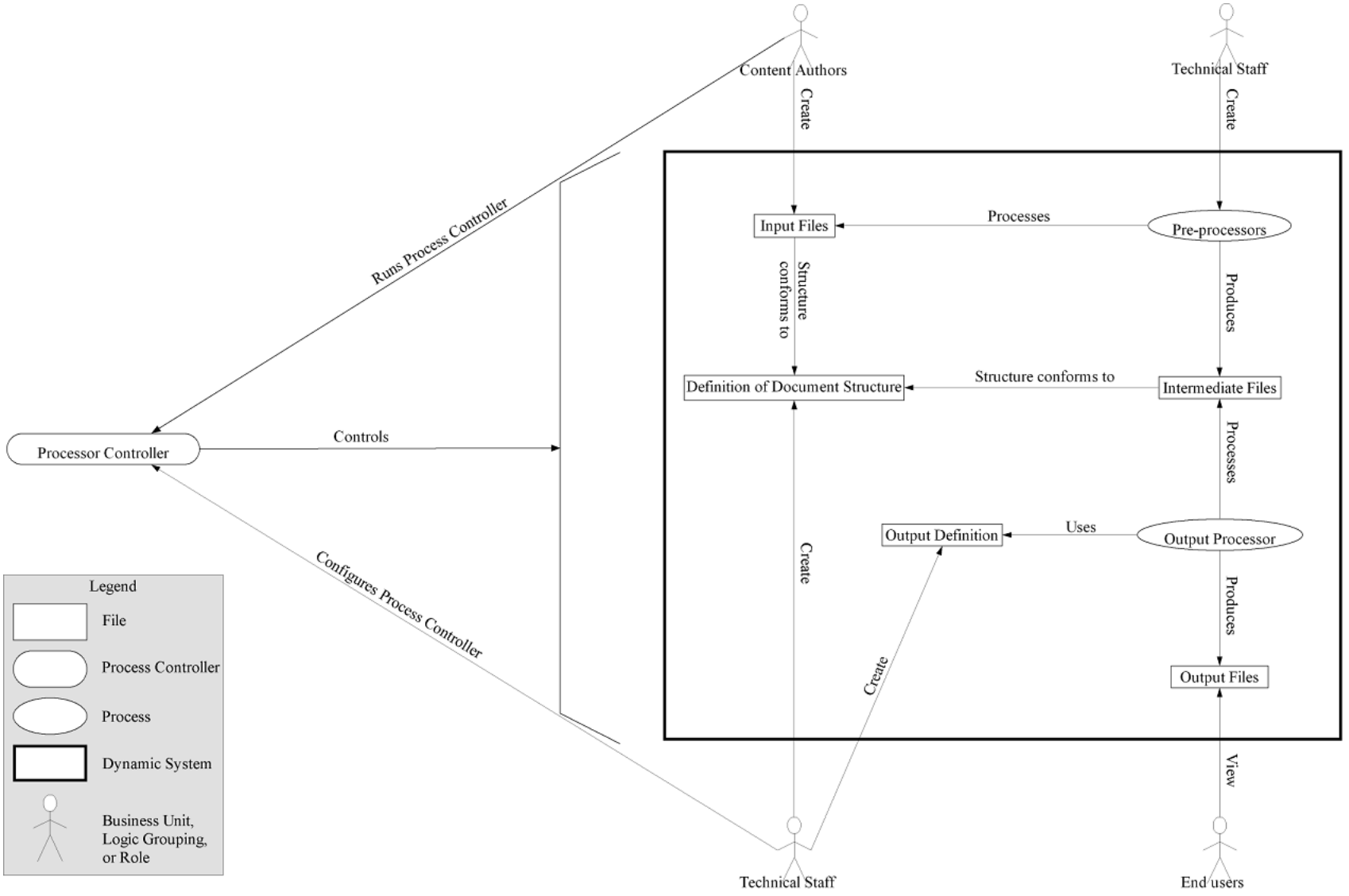

Figure 3: A modular website management system design mapped to roles

WebCMSs, as depicted in Figure 1. Hence, our proposed website management system design provides a software solution to the website management processes. The system is divided into following basic components. Note that some of the provided examples may relate only to Unix. They are provided to enhance clarity, but the design may be implemented within any operating system context.

- *Business Unit, Logical Grouping, or Role*
  The Business Unit, Logical Grouping, or Role relates to the roles mentioned in Figure 2. They are participants in an organisational website management system.
  - *Content Authors*
    A role that provides and maintains the Input Files for the Dynamic System. They also initiate the Process Controller to process the Input Files.
  - *Technical Staff*
    A role that provides the Dynamic System with the Definition of Document Structure, creates the Pre-processors, writes the Output Definition, configures the Process Controller, and maintains each of these.
  - *End user*
    A role that views the Output Files to obtain information.
- *Process Controller*
  Controls the system, for example by using GNU Make and Makefiles; is configurable by the Technical Staff who might use Makefiles, for example; and its operation is initiated by the Content Authors, for example by using GNU Make. It also defines the relationships between various software components in the Dynamic System.
- *Dynamic System*
  The system described within the boundaries of the Dynamic System is software based. It acts on input from the Business Units, Logical Groupings, or Roles. Figure 3 shows the relationships between various components of the Dynamic System.
  - *Definition of Document Structure*
    A file that describes the structure that a content document should have before it can be processed by a Preprocessor or an Output processor. This file may take the form of a DTD (Document Type Declaration), for example.
  - *Input Files*
    Files that adhere to the structure as defined by the Definition of Document Structure, and are created and maintained by the Content Authors, for example, in XML.
  - *Pre-processors*
    Programs or scripts, for example, PERL Scripts, that take an Input File or Intermediate File as input to produce another Intermediate Files. They may use the produced Intermediate Files or let the Output Processor process them.
  - *Intermediate Files*
    They are produced from Input Files or from other Intermediate Files and are not a stable part of the system. They contain data, for example in XML, to be used for processing by Pre-processors or the Output Processor.
  - *Output Definition*
    Defines, perhaps as an XSLT Template, for example, the structure and layout of Output Files produced by the Output Processor, transformed from the structure of an Input File or an Intermediate File as defined by the Definition of Document Structure.
  - *Output Processor*
    For example, the XSLT Processor xsltproc, takes Input Files or Intermediate Files as input to produce Output Files by adhering to the structure and layout defined using the Output Definition.
  - *Output Files*
    Are the final product of the Dynamic System, and may be in valid xHTML for example. Produced by the Output Processor using the Output Definition with Input Files or Intermediate Files as input.

Processors in the Dynamic System communicate with each other using files. Thus, it is possible to implement standards compliance in these files, have processors which act on standards compliant input, to produce standards compliant output.

The description above has introduced standards that we propose to use for various components in the example implementation of our design. That is, XML for Input Files and Intermediate Files, Document Type Definition (DTD) for Definition of Document Structure, XSLT for Output Definition, and xHTML for Output Files. We recommend use of software programs that work with standards compliant input and produce standards compliant output, for Preprocessors and the Output Processor.

4. EXAMPLE IMPLEMENTATION

Separation of content and presentation may be achieved through the use of XML, XSLT and CSS. Presentation can be separated further into document presentation and visual presentation, where document presentation refers to the ordering of content in presentation related mark-up tags, or structural presentation [33], while visual presentation refers to the visual layout and display of content on the screen. Further, as suggested by the examples provided in the W3C recommendations, valid xHTML and HTML documents may be produced by processing XML documents, conforming to a DTD through an XSLT processor (xsltproc [39]) using an XSLT template. Following the best practices as recommended by the W3C, styling or visual layout and presentation may be achieved by use of CSS.

The combination of such standard compliant tools provides us with the framework for our example implementation of the website management system which is described in Figure 4. This system uses GNU Make [9] as the Process Controller, a Makefile as input to the Process Controller, a Document Type Definition (DTD) [8] as the Definition of Document Structure, XML files as the Input Files, two PERL [13] scripts as Pre-processors, an XSLT (template) as the Output Definition, and an XSLT Processor (xsltproc) as the Output Processor.

The various organisational roles (from Figure 2) create and modify components within the Dynamic System as described in Figure 3. Roles that do not participate in the website management system participate in the CMS and in preparation of business policies and guidelines.

GNU Make integrates and runs many processes in order as defined in its configurable input called a Makefile. As every process that GNU Make runs has a definite standards compliant input and output, the input and output files can be considered as states.

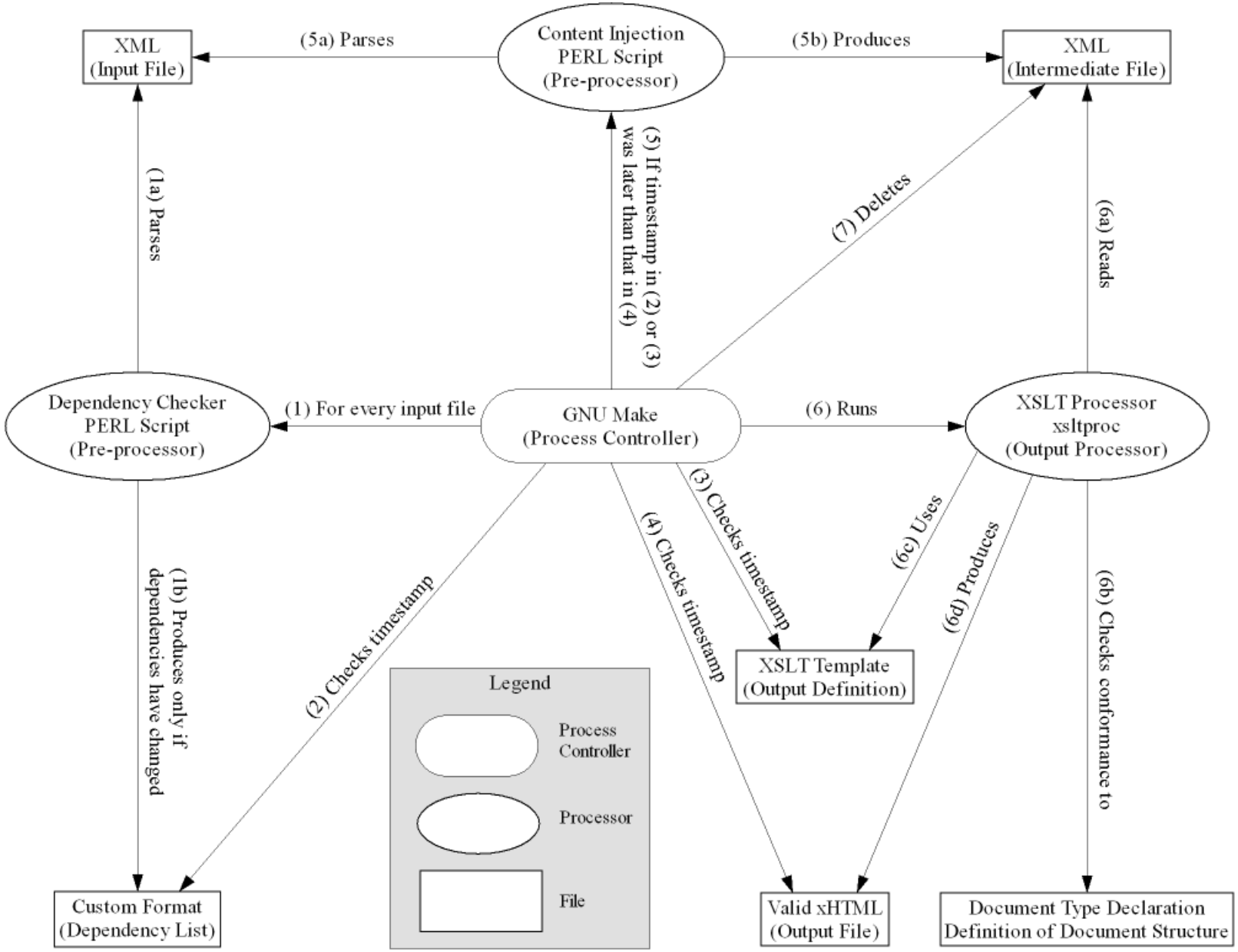

Figure 4: Example implementation of the system design using W3C standards and tools available in the public domain

If standards compliant input is not available, resulting from non-compliant output of a previous process or role, exceptions will be reported to the user of the system, such as Content Authors or Technical Staff. These exceptions are recorded with respect to the input XML documents, such as missing referenced documents or invalid structure. GNU Make moves the state of the system from one stable state to the other, till it reaches the final state, where the output is produced. Thus GNU Make acts as a Process Controller. Through use of the Makefile it is also possible to add other processes which GNU Make may call before reaching the final state and producing the final output.

In our example implementation, we have Input Files that depend on other Input Files for content, hierarchy, or for instance, title of links, and summaries. This relationship is managed by PERL scripts acting as Pre-processors. In GNU Make's terminology, every Output File is a Target, an Input File the Source, and all Input Files that the Source may depend on are the Dependencies. We used a PERL script to parse an Input File to produce an Intermediate File called Dependency Lists that contains a list of all Dependencies for the Source. Other Pre-processors use the produced Dependency Lists to produce the Intermediate Files that conform to the Definition of Document Structure represented by a Document Type Definition (DTD). A Target Output File is reproduced only if the Source Input File it is produced from, or one of the other Input Files that are Dependencies for the Source Input File, are modified.

GNU Make provides mechanisms for central configuration. A Makefile is a collection of commands to be run using the system shell */bin/sh* according to specified rules. It is possible to create rules that use commands to identify the user id (uid) and group id (gid) of the user executing GNU Make, thus applying different XSLT templates for users with different group affiliation, and determine the appropriate location of the xHTML web-pages produced. Security against tampering of the production system can be implemented using Unix file system permissions or access control lists. Other Unix tools such as Superuser do (sudo) may also be used.

Only the roles mentioned in Figure 2 and related to the Dynamic System in Figure 3 need to create or modify components in our example implementation. That is, Technical Staff create and maintain the Definition of Document Structure Document Type Declaration (DTD), the PERL Scripts, the Output Definition XSLT (template), and the Makefile used to configure and control the behaviour of GNU Make; the Content Authors create and maintain the Input Files; and the End users view and use the Output Files.

All components of the example implementation of our system design are available in the public domain and are generally also part of any academic Unix installation. The example implementation is available for download [4] and free for academic or personal use. The authors retain copyright of the code and the idea.

5. DISCUSSION

As he re-designed Sun Microsystem's website, Nielsen [19] demonstrated that it is possible to obtain a generic document structure and web-page layout for a website comprising a variety of web pages and content. Thus, it is possible to define the website design and layout once and implement it centrally using the Definition of Document Structure and Output Definition, as indicated in Figure 3 in our proposed system. In this system, time and resources are spent on the once-off design of the organisation's website to ensure its compliance with standards. Although the responsibility for creation of the content of the website remains distributed across the organisation the system ensures standards compliance by making it possible to control the output centrally. This removes the need for time-consuming processes to ensure that policies for enforcing standardised inputs are adhered to across the organisation. As indicated in the example implementation, the Input Files will not get processed unless they adhere to the Definition of Document Structure. That also means that if any dependencies are missing, the Output File will not be produced. This allows Content Authors to have real time feedback from the system with respect to missing documents and links. In effect, a technical enforcement is implemented rather than a business process solution to standards compliance.

Although the centralised website design control may appear to be somewhat restrictive the design allows for more Pre-processors to be added to be run by the Process Controller. Thus, programs and scripts can be added to add characteristic features to websites. The Output Definition can be used to provide a common structure and layout for the Output Files, and can enforce some good design features, such as page length and limits on level depth, that may encourage the adoption of good design principles to assist in making the website more usable and accessible. The system effectively removes the burden of implementing changes of procedure and practice across an organisation if changes to the website presentation are required as organisational needs change or additional standards are introduced. A once-off design change makes this possible without any need for alteration in business processes.

Loeser [16] investigated production of static HTML using databases, rather than the production of HTML dynamically by websites, and found many benefits including reduced server load and a reduction in the accumulated time for editing many pages. This system does not restrict the means by which the Input Files are produced and implementors of the system design may choose to use a versioning system such as Concurrent Versioning System (CVS), Subversion (SVN), a database, or a CMS with appropriate connectors to provide them. We would recommend use of Loeser's methodology in our proposed system so that the Content Authors are engaged in authoring the content of the Input Files for the website management system, while structuring content as dictated by Definition of Document Structure happens automatically.

The example implementation relies completely on tools available in the public domain, W3C standards, and approximately 900 lines of code including the PERL scripts, the Makefile, the DTD, the XSLT template, and the CSS. This increases the maintainability of the system as it does not rely on proprietary standards, tools, and code. Scalability of the system is dependent on the tools used, which have a proven track-record of being used in very large open-source projects collaboratively developed by people from around the world.

As a future development we would hope to extend our example implementation to include a Graphical User Interface to edit the source XML files. Currently, the Document Type Declaration (DTD) is used to define the structure of XML documents which will produce valid output. We would like to change over to the use of an XML Schema. Another extension would require investigating the possibility of integrating version control into the design of the system. Currently, there is no provision for version control. However the the system does not place any restrictions on version control of source documents using traditional file version control methods.

The development of the example implementation was carried out on Unix. Thus we were reliant on the built-in file access permission system [29] provided by Unix and Unix-like operating systems. Investigation is required to find a way to define and enforce an Information Architecture [31] as decided by the Strategic Decision Level of an organisation. Further, Grado-Caffaro and Grado-Caffaro [10] highlight problems related to security and future compatibility with respect to XML, which will need to be investigated with respect to

our system.

Apart from the shortcomings mentioned above, that require further investigation and improvement, our proposed website management system design still requires the support of Technical Staff to create, configure and maintain some aspects of the Dynamic System. This is seen in our example implementation, however, this may not necessarily occur in other implementations of the design. And, although we are have created the system from a combination of tools available in the public domain, there will always be a cost to organisations involved in implementation of this system in that these tools need to be coupled, configured and installed as specified in this paper.

6. CONCLUSION

In this paper we present a design for a website management system from existing standards compliant software that is available in the public domain. Our system is based on the premise that standards compliance and encouragement of adherence to good design principles facilitates the implementation of standards compliant, and usable and accessible website design. In this system standards compliance can be automatically enforced centrally, while maintaining the distributed nature of content creation and, hence, also the web-development roles existing within organisations. Such a centrally controlled model removes the reliance on workflow processes to achieve consistency and compliance and provides the possibility that standards compliance strategies may be developed and implemented centrally by usability and accessibility experts within organisations.

7. ACKNOWLEDGEMENTS

We are grateful to Iman S. H. Suyoto and Keith Frampton for their feedback on reviewing earlier drafts of this paper.

8. REFERENCES

[1] D. Alexander. How accessible are australian university web sites? In *Proceedings of The Ninth Australian World Wide Web Conference, Gold Coast, Queensland, Australia*, July 2003.

[2] D. Alexander. Redesign of the monash university web site: A case study in user-centred design methods. In *Proceedings of The Ninth Australian World Wide Web Conference, Gold Coast, Queensland, Australia*, July 2003.

[3] D. Alexander. Preparing web content for migration. `http://www.its.monash.edu.au/staff/web/slideshows/content-migration/`, February 2004. Viewed 09 April 2005.

[4] N. Arora. Project xnarora: XMLized narora.net. `http://www.narora.net/projects/xnarora/`, November 2004. Viewed 11 April 2005.

[5] J. Clarke. XSL Transformations (XSLT). `http://www.w3c.org/TR/xslt`, November 1999. Viewed 11 April 2005.

[6] J. Clift. University of south australia's user driven web redevelopment. In A. Treloar and A. Ellis, editors, *Proceedings of The Tenth Australian World Wide Web Conference, Gold Coast, Queensland, Australia*, July 2004.

[7] Commonwealth Scientific and Industrial Research Institute (CSIRO). `http://csiro.au/`, May 2005. Viewed 03 May 2005.

[8] Contributors to Wikipedia. Document Type Definition (DTD). `http://en.wikipedia.org/wiki/DTD`, March 2005. Viewed 10 April 2005.

[9] Free Software Foundation (FSF) Inc. GNU Make. `http://www.gnu.org/software/make/`, October 2004. Viewed 10 April 2005.

[10] M.-A. Grado-Caffaro and M. Grado-Caffaro. The challenges that xml faces. *IEEE Computer*, 34(10):15–18, 2001.

[11] V. K. Gupta, S. Govindarajan, and T. Johnson. Overview of content management approaches and strategies. *Electronic Markets*, 11(4):281–287, December 2001.

[12] G. Harman. New directions in internationalizing higher education: Australia's development as an exporter of higher education services. *Higher Education Policy*, 17(1):101–120, March 2004.

[13] J. Hietaniemi. Comprehensive perl archive network. `http://www.cpan.org/`, April 2001. Viewed 10 April 2005.

[14] L. Hull. The whiteboard: Accessibility: it's not just for disabilities any more. *ACM Interactions*, 11(2):36–41, 2004.

[15] E. Joyce. Study: Content management tools fail. `http://www.atnewyork.com/news/article.php/1690881`, February 2003. Viewed 01 May 2005.

[16] H. Loeser. Shift it to the server! - let the database server update your websites. In *Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00) - Volume 1, Hong Kong, China*, volume 1, pages 50–54, June 2000.

[17] V. Lombardi. Managing the complexity of content management. `http://www.boxesandarrows.com/archives/managing_the_complexity_of_conte%nt_management.php`, February 2004. Viewed 27 April 2005.

[18] D. M. Nichols and M. B. Twidale. The usability of open source software. *First Monday*, 8(1), January 2003.

[19] J. Nielsen. Sun's new web design. `http://www.sun.com/980113/sunonnet/`, January 1998. Viewed 09 April 2005.

[20] J. Nielsen. Usability 101: Introduction to usability. `http://www.useit.com/alertbox/20030825.html`, August 2003. Viewed 01 May 2005.

[21] J. Nielsen and K. P. Coyne. A useful investment: Usability testing costs - but it pays for itself in the long run. *CIO Magazine*, February 2001.

[22] J. Nielsen, K. P. Coyne, and M. Tahir. Make it usable. *PC Magazine*, February 2001.

[23] J. Nielsen and D. A. Norman. Usability on the web isn't a luxury. *Information Week*, January 2000.

[24] K. Patterson and A. Ellis. Usability and meeting the needs of educational web site users. In A. Treloar and A. Ellis, editors, *Proceedings of The Tenth Australian World Wide Web Conference, Gold Coast, Queensland, Australia*, July 2004.

[25] L. Quin. The Extensible Markup Language (XML). `http://www.w3c.org/XML/`, April 2005. Viewed 09 April 2005.

[26] L. Quin. The Extensible Stylesheet Language Family (XSL). `http://www.w3c.org/Style/XSL/`, March 2005. Viewed 10 April 2005.

[27] RMIT University. About the rmit web site. `http://www.rmit.edu.au/help/about/`, March 2005. Viewed 09 April 2005.

[28] D. Sloan, P. Gregor, P. Booth, and L. Gibson. Auditing accessibility of uk higher education web sites. *Interacting with Computers*, 14(4):313–325, 2002.

[29] G. Socher. File access permissions. http://tldp.org/linuxfocus/English/January1999/article77.html, October 2000. Viewed 09 April 2005.

[30] C. M. Sperberg-McQueen and H. Thompson. XML Schema. http://www.w3c.org/XML/Schema, October 2004. Viewed 11 April 2005.

[31] S. Subramanian. An introduction to information architecture. http://www.sitepoint.com/article/information-architecture?ct=1, June 2004. Viewed 11 April 2005.

[32] The Information Architecture Institute. The Problems with CMS. http://iainstitute.org/pg/the_problems_with_cms.php, 2003. Viewed 27 April 2005.

[33] The World Wide Web Consortium (W3C). The global structure of an HTML document. http://www.w3c.org/TR/REC-html40/struct/global.html, December 1999. Viewed 11 April 2005.

[34] The World Wide Web Consortium (W3C). Cascading Style Sheets (CSS). http://www.w3c.org/Style/CSS/, April 2005. Viewed 09 April 2005.

[35] The World Wide Web Consortium (W3C). Hypertext Markup Language (HTML) Home Page. http://www.w3c.org/MarkUp/, March 2005. Viewed 11 April 2005.

[36] The World Wide Web Consortium (W3C). The World Wide Web Consortium (W3C). http://www.w3c.org/, April 2005. Viewed 11 April 2005.

[37] The Zope Corporation. Zope.org. http://www.zope.org/, 2003. Viewed 02 May 2005.

[38] S. J. Vaughan-Nichols. Tempest over web-authoring tools. *IEEE Computer*, 34(10):12–14, 2001.

[39] D. Veillard. The xsltproc tool. http://xmlsoft.org/XSLT/xsltproc2.html, February 2005. Viewed 10 April 2005.

[40] L. Wise. The evolution and devolution of the university webmaster role. In A. Treloar and A. Ellis, editors, *Proceedings of The Tenth Australian World Wide Web Conference, Gold Coast, Queensland, Australia*, July 2004.

[41] A. Wright. Content management systems evaluation. http://www.atnf.csiro.au/computing/web/cms_eval.html. Viewed 02 May 2005.