Chapter 1 – Introduction

1.1 Background

The Internet has joined the mass media stable as a popular form for dissemination of information. With digital technology, authors are now either closer to the processes for production and distribution of their work, or at the coalface of production. At one time the output of the author passed through a chain of activity that included editors, designers and typesetters – and ultimately librarians to catalogue and classify that work. Authors have embraced this new process for production and distribution as a means of greater control, and foundational to this readiness has been the authoring technology that has enticed the author into the role of content creator and production assistant.

For corporate publishing, the proximity of the author to the publishing process presents a number of dilemmas, but chief among them is editorial control over the presentation of content. It is in this context that a new class of software has emerged: the "Content Management System" or CMS. The publishing of content on a website does not necessarily supplant the publication of this same content in print form and through various other forms of multimedia (for example CDROM), and the management of consistency across different media can be challenging.

The reuse of content has long been epitomised by the news-wire syndication – the commercial distribution of media news stories by major news distribution channels such as Australian Associated Press (AAP). News organisations use syndicated news-wires in their own publications (with or without attribution) and will themselves syndicate news items that they originate. The digital distribution of news and other content through syndication offers such organisations much wider international reach, and also enables smaller organisations to participate in syndication.

The CMS may serve a wider role than simple website publishing. The consistent authoring of digital content is an enabler for subsequent syndication of the information. Effective syndication of digital content hinges on both structural separation of content and consistent use of semantic metadata to aid the distribution process. The capture of news through Content Management Systems (CMS) and subsequent syndicated distribution through XML (eXtensible Markup Language) syndication is a natural extension of the traditional news-wire. Such syndication now goes beyond the traditional distribution to international news outlets. Many website portals now take syndicated news from a variety of sources.

Outside the domain of news syndication, reuse of content on the web has been hampered by a lack of consistent forms and metadata descriptions. While structured document markup approaches such as XML can provide a disciplined means of content separation in a manner that facilitates information reuse, even within a single domain, the use of agreed document "ontologies" does not assure consistent usage of sematic identifiers or markup of the content (Gruber, 1993).

XML has bought welcome relief in the management of text-based content – by increasing the focus on the content itself, hopefully enhancing preservation of the content from unnecessary change when adjusting the stylistic presentation and giving better document portability (Coombs, Renear, & DeRose, 1987). Both Knowledge Management Systems (KMS) and Content Management Systems (CMS) are Information Systems that rely on effective semantic metadata description to aid later information discovery. The CMS has joined these Information Systems to take an important place in the toolset for corporate management of information. A key conceptual element of the CMS is the separation of content elements from their presentational form to better manage editorial control and reuse of that content. In this context, the effective management of published resources becomes essential. The rapid emergence of the Content Management System (CMS) as a new class of enterprise software is a clear indication that many content publishers are placing a higher level of importance on the management of their web-based communication.

At both the national and international level, multimedia content creation is considered of immense strategic importance (M.E.S.O., 1995; O.E.C.D., 1998). The European Economic Community Multimedia Educational Software Observatory (M.E.S.O.) "Final Report" sees a "missing paradigm" for multimedia content production: it belongs neither in the processes for book publishing, film production, software development nor game development. Publishers working in the Education market have made a strong contribution to the production of multimedia content and exploration of the issues surrounding multimedia content reuse (Brett & Nash, 1999; Colbert, Peltason, Fricke, & Sanderson, 1997).

Most websites and multimedia resources go through several generations of design makeover. Considerable material and human cost goes into the development of multimedia resources that are used in only one publication. This investment can languish when issues of technological obsolescence intervene, or if the information itself is not current and is difficult to update. As a result, much of the content created for websites is discarded between generational changes.

The difficulties in achieving effective content reuse are compounded by the rapid obsolescence of both technology and content assets, with longevity of as little as 18 months in multimedia assets (M.E.S.O., 1995). This is particularly important where publishers invest significantly in developing content for a particular media. This difficulty is compounded by the need to publish in multiple formats. For instance, an encyclopaedia may be published in print, CDROM and web-based media. Each of these has different information systems required to create the published result, often resulting in separate departments creating runtime, design and content elements in isolation (Norrie & Signer, 2003). Multiple digital document standards and a rapid rate of obsolescence demand the design of content reuse systems that allow not only the repurposing of content, but also the effective regeneration in new runtime environments as part of a broader digital asset architecture and through complex workflow processes.

The diversity of authoring tools that feed into the multimedia production process compounds the difficulty of content reuse (Agoulmine et al., 2000; Duffy, 2000). Industry models such as the Microsoft "White Paper" on "Content Management" explore a largely linear view of the content management process (Reynolds & Kaur, 2000), with little exploration of the issues of content reuse. The high cost of multimedia content publication makes content management vital (Kelly, 2000).

The challenges of Internet publishing, therefore, characterise the key problems that all multimedia publishers face: editorial control of content, issues of heterogeneous authoring tools, long term persistence of information, technological obsolescence, editorial control of digital assets and version control of information.

The emergence of the Content Management System has certainly addressed some of the most problematic issues in web authoring. Functionally rich Content Management Systems began to emerge in the early 1990's to address primary issues of content authoring, version management, editorial workflow control and transformation of content for its target environment (Celentano, Pozzi, & Toppeta, 1992). The proliferation of CMS products to the market in recent years reflects the heavy demands for editorial management of a website.

The more complex the website, or the deeper the level of content, the greater are the problems of management. With the visibility of the website as the public face of an organisation, there is greater demand for currency of information; and therefore, the need for a framework through which website content can be easily updated by those most likely to supply the relevant content. Technical elements of web publishing are a major obstacle to content authors not familiar with the authoring and stylistic elements of web publishing; and the technical mechanics of delivering changes to the website have in the past been obstacles to content publishing. Design standards soon become a problem when more than one contributor is managing website content,

resulting in a demand for editorial control of information content and its presentation toward the maintenance of "corporate" design standards.

The term "content management" is in fact an attractively succinct description of the demands of managing multimedia resources. First, the term "content" is sufficiently general to describe any digital form, media or object ("content" in the sense of a descriptive label of items that are "contained within"). It also implies the elements of meta-information storage (in the sense of the "table of contents"). The term "management" can be extended to imply not only the operational management of the publishing exercise, but also the long term organisational management of the content store, the effective use of content through regeneration, repurposing and syndication, and the long term archival administration of content. A further limitation to content reuse lies in the original capture and management of the content in a form that maximises its reuse through several generations of runtime engine and content makeover. Also hampering structural design reuse is the lack of comprehensive website modelling approaches, and the disconnection between the technical architecture of delivery platforms and the business model (Gu, Henderson-Sellers, & Lowe, 2002).

Websites typically go through several generations of design. The underlying content, however, may be similar between these generations of graphical design. While an early principle of the CMS has been the separation of content from presentational style, many sites continue to experience difficulty in the transition of content through intergenerational graphical redesign of websites. Content reuse or re-purposing has a role both in the regular makeover of websites and in the syndication of content to a variety of different destinations and media - for instance the publishing of content in HTML (Hypertext Markup Language) form for web distribution and PDF (Adobe's Portable Data Format) form for print distribution. The paradigm of WYSIWYG (What You See Is What You Get) has considerable benefits for the content creator, but stands opposed to the structural need to decompose content elements and the stylistic elements for purposes of long-term independent management of content for content reuse. Evident here is a

tension between the separation of content and style for effective management and reuse of content, and the cohesion of content and style to enhance effective authoring.

The problem of content management centres not only on original publishing but also issues of content reuse for syndication and the complex regeneration of information in the context of a runtime engine (such as a web server, itself possibly hosting other information systems), and based on heterogeneous content sources.

CMS products focus on the primary generation and maintenance of website content, and their origins lie in simple content delivery mechanisms for web publishing (Fraternali, 1999; Reynolds & Kaur, 2000; Yeh, Chang, & Oyang, 2000), with a focus on effective workflow management of the production process. However, content separation from style and structure has been an axiom in the discourse concerning CMS since the emergence of XML. Content separation from design is achieved either through template-based approaches to content publication, or more recently through use of XSL (Extensible Style Sheet) templates interacting with XML document formats. Dynamic generation of personalised website using XSL from XML content sources is also gaining popularity.

In a reaction against page-oriented website management, there have been theoretical advances to formulate a consistent UML-style modelling approach to website management (Fraternali & Paolo, 2000). While this has not seen significant industry implementation, the approach points to the importance of an integrative view of the site model as well as the dissociative management of content fragments.

The task of Content Management does not necessarily end at the workflow organisation of the authoring process. Publishers also have a requirement to track the copyright clearance levels of the content they use. Similarly, from an archival point of view, the management of digital content presents the dilemma of multiple copyright ownership of a given content unit (Jewitt, 2000). Social and copyright issues explored by Cox (1998) in the context of academic journals

are equally applicable to the generation and regeneration of multimedia content in an Internet environment.

Kartcher (1988) quantifies the Return on Investment (ROI) achieved from a transition to a CMS-based production system for journal publications. This analysis illustrates the high initial cost of establishment followed by subsequent production savings:

> Over two years the CMS option is $43,000 higher -- 15 percent -- than the other publication method. But beginning in the next edition (gray cells), the CMS costs $66,000 less -- nearly 50 percent -- than the other publication method. That is a conservative analysis, because it does not take into account the potential increase in incremental revenue derived from new custom publications. (Kartchner 1988, p.4).

Even with the recent proliferation of Open Source and commercial CMS products, organisations introducing a CMS continue to face high initial implementation costs (Clyman, 2002b; Dalton, Manning, & Gardiner, 2001).

The management of online educational curricula is perhaps the leading area in which the challenges of content reuse are being actively engaged. There are manifest advantages to the effective reuse of learning objects in common teaching areas. The cost of development of curriculum content that is designed for effective visual communication in an online environment gives a clear imperative to effective reuse of curriculum materials. The need for a consistent framework has been called for in many domains (Candler & Andrews, 1999; Colbert et al., 1997) and is facilitated by the emergence of credible metadata ontologies, such as the Sharable Content Object Reference Model (SCORM) and IEEE Learning Object Metadata (LOM) (see Section 2.5.5 on Learning Management Systems).

The research presented in this thesis is directed toward the formulation of a framework for CMS development that recognises the long-term value of the content not only for original publishing but also for regeneration and reuse. In the early research phase of this thesis, in 2000, Content Management Systems (CMS) were a newly emergent class of industry software. The last three years have seen many more products released to the market, targeting different aspects of the

Internet publishing process, including several credible open source solutions. Yet despite substantial industry activity in the development of new CMS products, use of the CMS remains focussed on primary content generation (Stahl, 2003).

What is lacking in current CMS models is a paradigm for content management specifically directed to systematic content reuse. Current implementations, some of which have strong foundations in XML, need to be extended to reflect the ontological and systems issues that also affect content reuse. While industry products can claim a rich set of functions, and many are now well positioned in the Enterprise Information System market, they lack a coherent theoretical model for systematic content reuse both temporally (supporting long term reuse) and structurally (supporting reuse across various dimensions of encoding, language design). The problem of content reuse is more complex than can be addressed by the development of a single ontology or addition of functions to existing implementations. This thesis is directed to forming a richer theoretical foundation to the description, discovery, regeneration and reuse of content resources.

This thesis therefore presents a model for content management oriented specifically to regeneration and reuse of content, especially for Internet content delivery. The need for such a model is situated in a context where content reuse across generation changes of websites and across runtime environments is often minimal. The thesis also argues that effective reuse has implications for the long-term management of valuable content resources. The emerging infrastructure represented by Content Management Systems requires additional attention to the issues of content reuse. In the midst of frustration with the cost of the CMS (Dalton et al., 2001) and frustrations with achievements in reuse, this thesis develops a new model for content management targeted specifically at issues of content regeneration and reuse.

1.2 Research Questions

This thesis addresses the issue of the efficient and effective content regeneration and reuse using Content Management Systems across heterogeneous multimedia resources. It addresses the question of whether there may be a model-driven approach to support management of such resources that is more effective than current industry practise. A systematic model-driven approach to regeneration and reuse of digital multimedia content implies:

- A content composition and publishing engine which acts on the structural relations and generation rules for their output in published form.
- An Information Retrieval and content authoring user interface which presents the structure of the content as well as the content itself.
- A data management model that addresses difficulties implicit in working with loosely typed data forms and unstructured data, and addresses the issues surrounding complicated layers of meta-information.
- Effective realisation of principles for the separation of content, structure and presentational form (for example XML).

The model must deal with the difficulties of heterogeneous content resources and the need for a flexible document object model. The model should be effective to the extent that it could significantly empower system analysts and software developers in their development of CMS. The principal research question for this thesis is formulated as follows:

- What approach to content management will allow effective reuse of content across heterogeneous multimedia resources tracking version changes across language and time?

This may be stated in terms of the following objectives:

- To formulate a theoretical systems model for the management of digital multimedia resources directed to systematic reuse and regeneration of content.
- To examine the effectiveness of the model through the design and development of a non-trivial prototype as a proof of the concept and an evaluation of the prototype outcomes.

To answer this research question, as well as to clarify and focus the thesis research objectives, the research addressed the following questions:

1. What are the methods, techniques and tools currently available to support content management and content reuse? The research approach taken on this question comprises a detailed literature review (see Chapter Two).
2. What reuse models so far have been successfully applied to the process of content publishing? This question is explored in an Industry Software Review (see Chapter Two).
3. Is there an alternative model to support content reuse? A detailed research methodology (Chapter Three) underpins a synthesis of the theoretical research leading to a proposed new Content Model for Reuse (Chapter Four).
4. To what degree can the proposed model be implemented in practice? This is explored through the Software Development Research and a major field implementation (see Chapters Five and Six) of a prototype system as a proof of the concept in a complex publishing environment.
5. To what degree is the proposed model effective? This is explored through an empirical study of the reuse consequences of deploying the model (examined in Chapter Seven).
6. In what other ways does situating the content in this new model benefit the ongoing management of the CMS? This is also explored through Software Development Research (see Chapters Seven and Eight).

1.3 Significance of the Study and Target Audience

The need for an effective model for digital multimedia content is evident both in the industry context, where innovations in Content Management Systems have failed to meet user expectations (Dalton et al., 2001), and failed to capitalise on syndication and reuse (Stahl, 2003)

and in the theoretical context, where the many challenges of multimedia content management and reuse over time are yet unresolved. The need for effective content management is also illustrated by the industry growth of tools to reverse-engineer the original content from published content and structure (Tilley & Huang, 2001). The development of an effective model for content reuse demands further research. Document Management Systems have addressed for some time the issue of organisational management of large stores of documents, and Knowledge Management Systems have fruitfully addressed the need for systems to explore and discover content based on complex relevance criteria. This research has relevance to the area of information management, the semantic description of information in Knowledge Management while directing both theoretical and industry research away for the current haphazard accretion of functions in Content Management and toward a more rigorous information model that facilitates content regeneration and reuse.

New technologies such as Web Services, while emphasising software reuse, continue to fail the requirements of content reuse (Kirda, 2001; Tilley & Huang, 2001). This thesis has relevance to those researching information models for content, and reuse of content over Web Services. The thesis is also relevant to the theoretical community researching issues of digital content structure and metadata description, information retrieval, and web-based content organisation and use, with a particular focus on web-based publishing.

1.4 Intellectual Property

This thesis is entirely the work of the author and complies with the University of Sydney (Intellectual Property) Rule 2002.

1.5 Thesis Organisation

The central focus of this thesis is the formulation and evaluation of a model for content regeneration and reuse. The chapters of the thesis are organised to this end as follows:

Chapter 1 sets the background and introduces the research objectives guiding this thesis.

Chapter 2 examines the problem area with review of current relevant trade and theoretical literature.

Chapter 3 introduces the research methodology guiding this thesis.

Chapter 4 presents the Content Model for Reuse (CMR), a theoretical model for effective content regeneration and reuse.

Chapter 5 introduces the Inter-Publish prototype, its design and implementation.

Chapter 6 presents the major field test in which the prototype implementing the CMR model is evaluated.

Chapter 7 evaluates the prototype through an analysis of the results arising from multi-national field-implementation started in 2000 and completed in 2003.

Chapter 8 summarises the research results and proposes future directions to improve Inter-Publish and to further research on content regeneration and reuse.

Appendices:

- Appendix One presents the functional matrix derived from of a review of fifteen CMS implementations (including Inter-Publish) with a map of literature reviewed.
- Appendix Two contains a detailed Web Services definition for the Content Model for Reuse.
- Appendix Three documents the Inter-Publish Transformation Language, with references to other documentation resources.
- Appendix Four presents the Technology Acceptance survey design and the survey responses.

Chapter 2 - Literature Review

> *Eager though he was to tell what he had seen, he did not dare to reveal the shameful secret, and yet he could not keep quiet about it. So he went off, and dug a hole in the ground; then he whispered softly to the earth he had dug out what kind of years he had seen on his master's head. Throwing the earth back again, he buried the information he had given and, after filling in the trench went quietly away. But a think carpet of trembling reeds began to push up on the spot and, at the end of a year, when they were full grown, the reeds betrayed their gardener: for, when stirred by the gentle South wind, they uttered the words that had been buried, and revealed the truth about his master's ears. Ovid.* Metamorphoses *Book XI.*

This chapter presents a comprehensive survey of the theoretical literature pertaining to content management and content reuse. The literature review first examines the terminology that has evolved to describe content, content management and content reuse, and its intersection with associated disciplines such as Semiotics (Section 2.1). The parallels between content reuse and software reuse are then examined (Section 2.2). This discussion of the nature of content itself leads on to an examination of methods for encoding and representing content in digital document architectures and methods for content transformation for use in digital runtime engines (Section 2.3). Methods for metadata description, content discovery, and current approaches to content syndication are explored in Section 2.4. Various systems approaches to management of multimedia content are discussed, with a particular emphasis on the characteristics that differentiate Content Management Systems from Knowledge Management Systems, Document Management Systems, and Digital Library` Systems (Section 2.5). Current industry approaches to the development of Content Management Systems (CMS) are examined through a survey of representative commercial and open source approaches (Section 2.6). The systems discussion opens the way to a discussion of the economics and the management aspects as they pertain specifically to content reuse (Section 2.7). At each stage of the literature review principles for content reuse are highlighted and these principles are consolidated at the end of this chapter as twelve principles pertaining to multimedia content regeneration and reuse. These principles form the foundation for the development of the research strategy presented in Chapter Three.

2.1 Definition of Terms

> *The ontology of a theory or body of assertions is the set of things to which that theory ascribes existence by referring to them in a way that cannot be eliminated or analysed out by reduction (Bullock, Stallybrass, & Trombley, 1977,p. 605)*

The term "content" may have many possible interpretations. This thesis addresses "content" as the informational element of a multimedia resource across the universe of possible means for representation. This content may comprise the entirety of the informational content of a multimedia document, or it may be an information-bearing fragment of a larger multimedia resource. In this respect "content" is understood to mean "digital content", and invariably comprises some form of digital encoding. However, the information about the structure of information elements (for example, the XML markup ontology) can also be regarded as a form of content - "content about the content" (*i.e.* metadata). Other forms of content can include "content about the structure" (the logical organisation of information), content about specific representational forms (templates and layouts) and content regarding security and workflow.

Semiotics, or theory of signs, differentiates the substance (or semantic content) of a communication and its particular form. In Semiotics, the study of meaning of signs and symbols, we see the most cogent analysis of the transformative process that humans perform in creating signs and symbols, and the interpretive subjectivity of such signs and symbols (Holdcroft, 1991). Semiotics broadly classifies signs and symbols in terms of the *signifier* and the *signified*. The *signifier* is the visible sign or symbol itself. The *signified* is the meaning or the set of meanings that can be recognised in the symbol. Digital content is, axiomatically, encoded by some process into digital form. This is one level at which this semiotic differentiation is relevant to the treatment of digital content. If we take *content* as the substance of the communication (however the final communication is expressed) then we see common ground between traditional publications and digital publications. The digital representation of content requires an intermediary process of encoding for which explicit standards are required beyond the normal conventions of written language or physical realisation of an image or expression. Such encoding standards are particular

to the computing environment (operating system, architecture, programming language, network environment and so on) for which they are targeted. Other types of content might include, however, metadata, information about the structural interrelationships between content elements and presentational design information for delivery in different forms.

This usefully expresses the process of content reuse: the signifier is the outcome of content reuse – the particular published manifestation, be it in HTML, email or other specific final form. The signified are the content structures that have been assembled through the mechanisms of content reuse. Hjelmslev and Whitfield (1961) extend the semiotic distinction of the signifier and signified to a fourfold distinction that looks to the form and substance of both the signifier and the signified which can be paraphrased as:

- The nature of the content. Hjelmslev distinguishes between form and substance of that which is signified. In the context of digital content this can be usefully recognised as the encoding of the content (its form) and its meaning (its substance).
- The nature of the expression. The published result is a process of transformation that renders a digital multimedia document for use in a runtime engine. This final expression can be understood both in its substance (the meaning communicated and the relevant encoding of metadata) and its form (the presentational format suitable for the runtime engine).

As with any communication, content can be fragmentary or coherently organised. A finished, coherently organised set of multimedia content objects is generally termed a "document". For Wilkinson (1998), "a document records a message from people to people" (Wilkinson, 1998, p.2). The Oxford English Dictionary (OED) is more expansive, describing a document as "Something written, inscribed, etc., which furnishes evidence or information upon any subject, as a manuscript, title-deed, coin, picture, etc." (OED, 2nd edition, 1973).

There is an intangible element of *finality* in the OED definition, not present in the Wilkinson definition, which is relevant to multimedia documents. This sense of finality of

"multimedia document" fits well with the "final expression" of Hjelmslev and Whitfield (1961). That is, a document is a logically complete set of multimedia content components ready for delivery by a runtime engine. For example, in a website context a multimedia document might comprise:

- The HTML-format web launch page
- A SMIL multimedia document
- A collection of multimedia documents or fragments (text, audio and video) used by the SMIL presentation.
- The set of hyperlinks and text items used by the SMIL Presentation

In the above example, at least three runtime engines may be engaged in document delivery: the client browser, the streaming media server and the web server.

It is quite apparent that the "multimedia document" is the result of one or many transformational and encoding stages using any number of content components following a particular structure in a runtime engine to deliver the final document. It is also clear that the management of such multimedia documents must address a complex and heterogeneous set of standards that govern the encoding of the content, the supplementing of the encoding with additional, analytical narrative (metadata) that addresses the substance of the content, and the relevant generative mechanisms to create the published expression in a relevant presentational format.

Content *reuse* is understood to mean the reuse of information in any context other than a purely archival manner, including the management of semantically identical content in different language forms (language variants), syndication of content, reuse of content within different contexts of a single environment, use of the same content in different presentational forms or media, and management of content transition through generational changes of larger document or website structures.

Following Heitmann (1999) and many others, this thesis distinguishes between *essence* and *metadata*, together comprising the *content* of a multimedia document. Explicitly in this thesis content is discussed in terms of "content units", or the heterogeneous multimedia elements that can be bought together to form a complete multimedia presentation, however delivered, and "content metadata", comprising semantic descriptors that are used to describe and enhance access to these content components. For instance, content reuse is exemplified in the standard website "news" page. It may be surrounded by website "navigation", a company logo, associated links and other branding and navigational aids. Assuming good web page design, the user will quickly recognise the true semantic content of the page – the "text" of the news item and associated images or video. The identical text may have been distributed for use on other websites – repackaged, re-branded and wrapped in different navigation, but essentially the same "message". In its simplest form, the content unit in this example could be the text of the news item. Another form of content reuse in this example could include the very navigation on the news item. The stylistic conventions – navigation, branding, look and feel – themselves are a type of content.

Software development methodology offers practical application of reuse concepts in application design. Prieto-Diaz (1989) defines reuse as 'the use of previously acquired concepts or work products in a new situation. It involves representation of development work-products at various levels of abstraction, storing these work-products and their components for future reference, identifying the similarity of new and old situations, retrieval and duplication of already developed objects and actions, and their adaptation to suit new requirements. Such concepts for software reuse are equally applicable to elements of content presentation such as layout design. The intersections between content reuse and software reuse are examined in more detail in Section 2.2.

Reuse of content is necessarily situated in a larger context that prescribes the standards whereby content is structured and exchanged - the "ontology". The term "ontology" has gained considerable prevalence with the Information Technology community, serving to describe the

data schema or content model with which data and content elements comply or the structure and semantic description which govern the document markup process. Ontology, in broad definition, describes "the theory of existence, or more narrowly, what really exists" Bullock et al. (1977) or more specifically Gruber who defines ontology as a "formal specification of a conceptualisation" (Gruber, 1993). In this context, the Knowledge Management community has found the term useful to describe that set of axioms, taxonomies and heuristic algorithms that Artificial Intelligence lent to the task of automatic analysis and discovery of document contents.

More strictly, in the development IT language structures, formal ontology sees a set of axioms and definitions in an Information System which could be mapped to a structure in logic. These are epitomised, for example, in the semantic rules for defining a relational database schema.

The usage of "ontology" has been adopted by the XML community to describe the formal specifications underlying the document structures in which content markup occurs - a kind of existential framework for content structure, or as Erdmann and Struder describe it: "…the ontology is a kind of mediator between the information seeker and the set of XML documents. It unifies the different syntaxes/structures of these documents and can add background knowledge to the process of answering a query." (Erdmann & Struder, 1999, p.2)

The use of the term "ontology" is indicative of the confidence of the XML community in their paradigm for content representation:

> For sharing information and knowledge (that means for interoperability) between different applications a shared set of terms describing the application domain with a common understanding is needed. More flexibility is gained, if not just a flat set of terms is defined, but also relationships between these terms. (Semanticweb.Org, 2003)

In the context of the end-user representation of content through Intelligent Agents, Portals and Customisation technologies the term "ontology" is used to describe "the formal means that specify the domain of interest for the clientele of the knowledge portal" (Staab & Maedche, 2001, p.63).

Gruber ascribes to the formal definition of conceptualisations: toward the goal of "knowledge reuse"(Gruber, 1993). Following Gruber, this thesis treats the content "ontology" as formal definition of the relationships between content "objects" and framework for describing these content "objects". This ontology can play a particular role in content reuse in providing a commonly agreed means of content interchange. This thesis describes the information and metadata content "objects" that exist within a document "ontology" as *content units*. This is substantially similar to the "fragments" of the ZyX model (Boll & Klas, 2001), Digital Item Declaration of MPEG (Burnett, van de Walle, Hill, Bormans, & Pereira, 2003), the "components" described by Boiko (2002), "shredding XML" (Udell, 2003) and the "chunks" described by Byrne (2003a). The preference for the term "content units" is to give emphasis to the economic nature of the reuse purpose (discussed further in Section 2.9 below), and explicitly to distinguish this from content models directed principally to representation of the final content form.

2.2 Content Reuse and Software Reuse

Chapter One introduced the various forms of content reuse: syndication of content to other parties, regeneration of content across generational changes to a website and recompilation of content in different presentational forms.

Most websites elevate the structural context information for a page to be at least co-equal to the content presented, partly due to the technological context in which the information is presented (a page at a time with multiple launch points available through hyperlinking). In most websites, the navigational artifice can easily be separated from the unique informational element of the page. Consistency in the navigational design has obvious merits, given the inherent "random access" that is represented by hyperlinking - there is often no certainty at which page the user might arrive in the site. The navigation artifice can be regarded a separate content form from the information content on the page. The Montage website, for instance, exhibits characteristic navigational and content constructs (see Chapter Six). Given the limited screen space available

for information display, navigational constructs often "wrap" the content using a navigational menu at the top or side of the page, with the informational content specific to the page centrally placed.

The web page shown in Figure 2.1 characterises simple navigational techniques for contextual presentation of information in a website context.

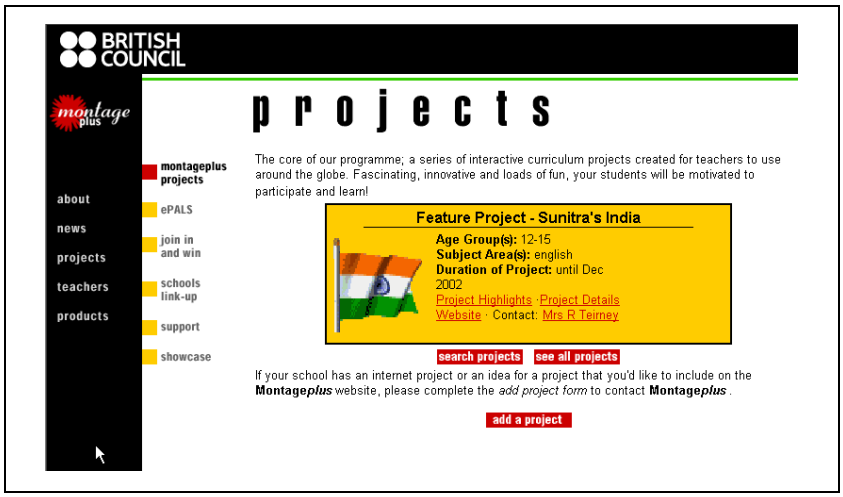


Figure 2.1: A web page from the Montage website

The simple separation of content from presentation layout could yield the template shown in Figure 2.2 below.

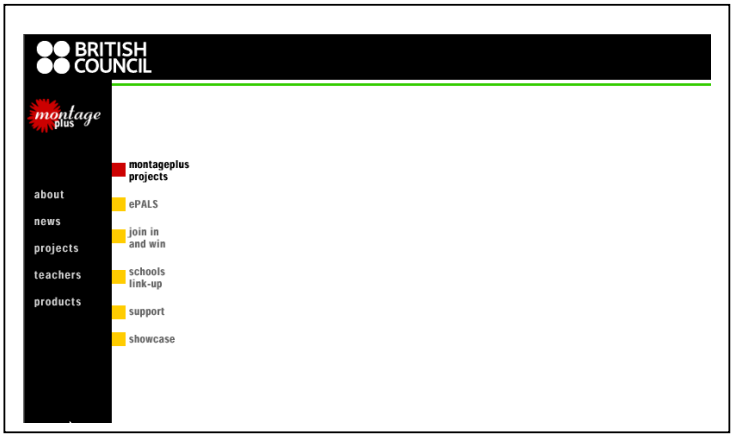


Figure 2.2: A presentation template abstracted from Figure 2.1

A coarsely granular content unit abstracted from this page is shown in Figure 2.3 below. A more

granular fragmentation of the content might involve the creation of content units for each key element on the page.

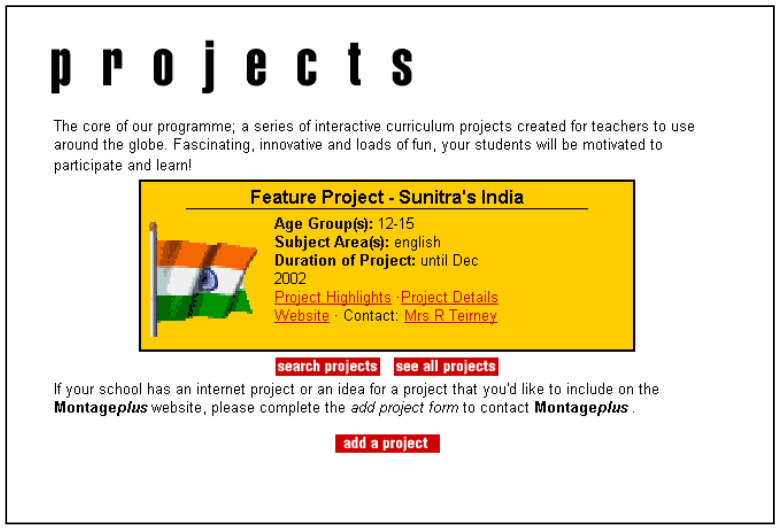


Figure 2.3: A content fragment from Figure 2.1.

The ease of publishing a simple page on the Internet belies the complex issues that surround maintaining stylistic consistency and semantic integrity in the context of a website as a whole. Separation of informational content from stylistic content has been axiomatic to text markup systems such as SGML (Standard Generalized Markup Language) and XML in order to facilitate the transformation of documents in different output styles and formats.

While it might be useful to extract the content element automatically from web pages, the simple visual differentiation of content from form is not a trivial task for automated semantic engines. Furthermore, even in the creation of a simple web page structure such as the above, some document model will be required. Such a template-based approach to content allows the publisher to reuse both design elements and content in the generation of consistently "branded" original pages, the regeneration of content across "brand" changes of the site, the regeneration of content for purposes of syndication, and the regeneration of web pages across version changes of the same content. Implicit in digital publishing is a transformational process, which assembles the content for final expression. Also part of this process of final expression is the discovery of the

semantic unit of content at which content can be effectively reused across the universe of possible means for representation of the content and an understanding of the document model, which can be used to describe this content. Selective reuse of content for syndication, website regeneration or creation of content in different formats requires that the content be organised in a manner that facilitates the selection and transformation of this content. Characteristically this process is facilitated either through the introduction of structural markup tags in the content itself, or the fragmentation of the content across database tables, or a combination of these two approaches.

This leads to two initial principles for content reuse:

> *Principle #1 - Content reuse requires the organisation of content in a manner that allows the separation of content from its final presentational form.*
>
> *Principle #2 - The delivery of digital content in its final presentational form will involve some form of transformational process.*

There are useful parallels between content reuse and software reuse. The emergence of object-oriented programming saw the formalisation of software reuse as a systematic practice in software design. Krueger (1992), observes that software reuse has been an element of software design through several different paradigms of reuse, including:

- Design and code scavenging - the ad-hoc reuse of code snippets from existing programmatic examples or design examples
- Source code components -the building and reuse of components, and in object-oriented programming the inheritance of existing components to build more complex components
- The abstraction of schemas to give a higher-level model for software design and component reuse
- Application generations - including generators that can build software from modelling languages such as UML
- Very high-level languages, transformational systems, and software architectures, all being progressively more abstracted models that rely on established

> components to allow highly abstract but coarsely granular application design. (Krueger, 1992, p.132-133)

The rationale for software reuse includes high cost of original software development, and the frequently repeating paradigms evident in the design of software applications. These motivators are not dissimilar to those that are driving the formalisation of methods for content reuse. Not surprisingly therefore, software reuse offers several lessons to the development of content reuse systems, due both to the similarities in the potential benefits of software reuse and the organizational barriers presented to implementation of software reuse techniques. Krueger (1992, p.133) identifies "abstraction" and "selection" as key elements of Software Reuse. He describes "abstraction" as the process whereby software is generalised in a manner that will facilitate its reuse by others. Selection is the abstracted paradigm whereby the software author "discovers" software for reuse.

Cybulski (1996) identifies the benefits of software reuse in:

- Savings in costs and time.
- Increase in productivity.
- Increase in reliability.
- Improvement in documentation and testing.
- Increase in ease of maintenance.
- High speed and low cost replacement of aging systems.

The Internet keeps alive the paradigm of code scavenging, with a multitude of code support websites offering small and large snippets of code for reuse in other applications. Object-oriented programming privileges more formal software reuse through systems of inheritance.

Just as there are similarities in the benefits of software reuse with those of content reuse, so also there are similarities in the barriers to content reuse. Sherif & Vinze (2003) articulate these barriers in their qualitative study of software reuse in industry. They note particularly that a key

factor in the success of software reuse in an organization is the existence of a champion within management to further the cause of software reuse.

It is not surprising therefore that the paradigms for software reuse have served as a paradigm for developers of Content Management Systems. Boiko draws parallels with software components and content components, choosing an object-oriented paradigm for representation of content fragments (Boiko, 2002). Fraser (2002) presents a view of content reuse that draws more heavily on the object-oriented paradigms, referring to content as a "component" and particular publishing instances as "specialization" of the component.

There are real similarities between content reuse and software reuse, and these similarities are already reflected in the first two principles relating to content reuse enunciated above. However, there are also significant differences between software reuse and content reuse. When dealing with unstructured "content" the object that results cannot necessarily be resolved into a simple "class" and the issues involved with the organisation of content management, authoring, selection and reuse of unstructured collections of text and multimedia resources cannot satisfactorily be characterised by typifying all content reuse as a form of "specialisation".

Fraternali presents high-level modelling approaches as a means of abstraction of design at the level of the website as a whole, drawing strongly on UML-style concepts (Fraternali & Paolo, 2000). Rockley (2003) classifies "content reuse" into two methods, "opportunistic reuse" (akin to Kruegers "code scavenging") where content is retrieved and reused in a "cut and paste" manner, while "systematic reuse" is achieved where the Content Management System automatically manages the reuse of content across target environments - what Rockley calls auto-population. Her content management view is modelled on the document management "life cycle" approach to the history of the content (Rockley, 2003). Both of these reflect an information systems view of content rather than a "component" view. This distinction is discussed further in Chapter Four. Nevertheless, both research into software reuse and its practical lessons will remain a constant point of reference in developing the information systems for content reuse.

Approaches to software reuse affirm the abstraction process that is discussed in Section 2.1 in forming the document ontology, leading to the first two principles described in that section. Krueger's emphasis on “select” as an essential element of software reuse has relevance to the issues of content reuse and are characterised in this thesis as “content discovery”. The two elements of content discovery - metadata description and content indexing - are discussed further below. This section leads to the additional content reuse principle:

> *Principle #3 - Facilities for content discovery and structured capture of semantic information to facilitate content discovery are essential to content reuse.*

2.3 Digital Document Architecture

> *The book is an exceedingly flexible invention. As a tool for studying and analysing texts, it has proved its power for well over a millennium and moveable type only enhanced it s effectiveness. Long use has added the benefit of standardization: people know how books work, and can learn pretty quickly how to use even complex forms of the book, like scholarly editions. (McGann, 1998, p.613)*

This section explores the many ways in which digital content can be encoded, annotated with additional semantic descriptors and described in a digital document structure. Content Management stands astride many standards, but particularly those pertaining to content encoding, markup, transformation and delivery. A multitude of standards exist for encoding of the content, its structure or its semantic information. Many of these encoding standards have changed dramatically over the last two decades, affected particularly by theories of presentation-neutral encoding of content. This section provides a brief overview of content standards for multimedia encoding and markup strategies, particularly as they pertain to web publishing, and their implications for content regeneration and reuse.

The storage of content in digital form necessitates some form of encoding and markup, however minimal. Content encoding standards address the methodology for digital representation of information.

Encoding standards affect the digital storage systems for content as well as the communication and runtime delivery systems. Markup of content extends the encoding layer to describe structural or metadata information concerning the content, or more specifically:

> The placement of identifiers in text from which can be inferred information regarding the presentation, formatting and structure of the text or which adds additional commentary regarding the text (but not part of the text). (Sperberg-McQueen, Huifeldt, & Renear, 2000, p.216)

There is therefore a dualism of purpose in the role of markup, and debate regarding the role of markup for structural or metadata description continues (Sperberg-McQueen et al., 2000). Markup can form part of the "raw" form of digital content as well as the final presentational form. The multiplicity of standards for encoding across different content types (text, video, audio and animation) complicates the issue of obsolescence on both sides of the transformative equation, especially for multimedia publishers who have already invested significant expense in designing content for a particular media encoding format.

For multimedia publishers, regeneration of content is a significant element in effective content reuse, and multimedia content regeneration can involve complex workflow processes before achieving the final product (M.E.S.0, 2005). Further regeneration may be mandated to suit new encoding methods, compression techniques or document delivery models. The storage of content in a presentation-neutral manner is designed to simplify the process of later rendering this content in different final encoding forms. The goal of presentation-neutral encoding of multimedia data is by no means trivial, and content reuse can demand close cooperation with multimedia content originators (Boll & Klas, 2001).

2.3.1 Encoding (text and multimedia)

The invention of the earliest compilers implied the need for encoding systems to render textual, numeric and program information in digital form. The fundamental text forms of digital encoding of English characters emerged from standards such ASCII. Specialised encoding approaches to other languages have resulted in system-specific character encoding formats.

Japanese, for instance, boasts at least 3 major character encoding variants – Shift-JIS (a Microsoft favourite), EUC (a Unix variant) and Unicode. The text encoding landscape is, fortunately, being positively transformed by the emergence of Unicode as a generally accepted standard for text encoding. The melding of Unicode with markup standards is a work in progress.

Word processing systems have revolutionised text encoding and markup, particularly with the emergence of the WYSIWYG paradigm and the ubiquitous desktop processor. Word Processing systems add annotations to the text in order to define how the text will finally be presented. The "printed" document is an "end result" of what is often a sophisticated production process. The popularity of Word Processing systems in business exemplified by Wang in the late 1970's and Tex on Unix systems soon spread to the desktop systems. Word Perfect set an early standard for widespread deployment of desktop word processing.

Text markup is a set of formal rules for the separation of underlying content from information regarding its formatting and presentation. By "markup" the author refers to anything other than the actual content. In particular markup may indicate the visual presentation of the text itself: the font, use of bold, italics, underlining, colour, etc (this is called "procedural markup") or markup may be used to denote the structure of content ("descriptive markup"). The objective of this markup is to organise encoded content in a digital document architecture.

Procedural markup has severe limitations in portability and general utility (Coombs et al., 1987). Changing the document to be rendered to different devices, printers and formats presented the first challenge. Information interchange between systems and word processing engines is particularly frustrating for content based on procedural markup (Ide & Sperberg-McQueen, 1995). Furthermore, numerous different applications were exploring procedural markup techniques proprietary to their word processing engines. More fundamentally, however, the descriptive markup process can "augment human intellect by providing concept-manipulation aids" (Coombs et al., 1987, p.932).

Procedural markup does, however, lend itself to WYSIWYG presentation of text, where content and style are brought together in a manner meaningful to the author.

If text encoding suffers from a diversity of standards, the evolution of encoding approaches for other multimedia content types has been unrestrained. On the Internet alone there exist several widely used image formats - GIF, JPG and TIF being the most popular. Within each of these image encoding approaches, there may be varying methods for compression (some of them proprietary).

The presentation of images on the Internet is generally (and ideally) a process of image transformation to render the image in a suitable format for effective and time-efficient presentation to the end user. This transformation process attempts to retain the content value sometimes at the expense of fidelity of resolution of the image. Limitations in the colour palette (in GIF) and compression techniques can yield quite adequate results for aesthetic presentation while retaining the key content value. Unlike text, in most cases the digital encoding of an image requires the choice of a discrete level of encoding quality. Other than vector-based images, the trade-off is generally between size of the encoded image and quality of the result. Encoding standards can, themselves, change over time, as can compression algorithms used with the encoding. Images are used for a multiplicity of purposes, given their natural communicative value: website navigation and structural information, communicating rich symbols and meaning, presenting multi-lingual or complex text in a universally fixed format, and so on. Metadata relating to the Intellectual Property constraints surrounding the image use are of the essence in complex multimedia libraries. Intellectual Property limitations can be complex: they may limit the media, the duration or the reach of the content.

Multimedia resources such as images, video and animation can be demanding on all elements of the multimedia information system. Video resources are typically demanding in storage requirements and have varying approaches to embedding associated metadata with the

encoded object. Client bandwidth limitations may require the generation of several versions of the same multimedia object encoded at different levels of compression. The multimedia system may also need to retain multiple versions of the content (for example, images stored at various different resolutions and sizes). Thumbnails of image represent yet another variant of the original multimedia object.

There is a rich diversity of encoding standards for audio and video, reflecting bandwidth challenges and innovation in fidelity of presentation. Several encoding methods are popular, leading many media sites to deliver video content in several formats. While presentation-neutral original content formats are the obvious solution, Boll and Klas (2001) suggest that the effective reuse of multimedia content requires careful design consideration at the early stages of multimedia document design – the consequence of which can be greater flexibility in the reuse of multimedia content in different contexts – for example, the development of a multimedia lecture which can be delivered either as a streamed object on the web or as a PowerPoint presentation, with adjustment of the media to suit the different delivery contexts.

2.3.2 Text-based Electronic Publishing

A number of standards-oriented architectures for digital document organisation of text have emerged. Foremost among these is Standard Generalized Markup Language (SGML). A key objective of SGML was to move away from the highly specific markup syntax of proprietary document systems to a universally understood set of standards for device-independent, system-independent representation of text data in electronic form. SGML was adopted as an international standard in 1986 (International Organization for Standardization, 1986).

SGML is a “meta-language” which allows the definition of the rules by which markup is undertaken in a hierarchical manner, and as such is a Descriptive markup language. SGML defines a generalised hierarchical markup syntax for content it encapsulates, with an objective of separating “content” from any particular “presentation” and embedding with the content not the

specific style but structural rules for "content" – for example chapter headings, paragraphs, etc. The presentation of content therefore takes the content and its structure, and applying a "style" sheet to the structure in order to produce the final result. The conceptual separation of the formal structure from the content has been profoundly influential in concepts of content reuse, especially in the context of XML.

The specific rules for markup are defined in the SGML Document Type Definition (DTD). Armed with a DTD to describe the encoding standards, a document can be encoded and exchanged and interpreted by anyone receiving that document. The outstanding example of the document sharing enabled by such standardisation is the Text Encoding Initiative (TEI), an SGML standard. The TEI has been the pre-eminent vehicle for textual markup of texts in the Humanities for the last decade. The TEI serves as a framework for consistent digital encoding of large bodies of literary works (Casillas, Abaitua, & Martinez, 2000). The TEI is a finely granular markup ontology, serving to facilitate the integration of disparate collections of literary resources under a consistent markup ontology.

However, the TEI experience with SGML illustrates some of the theoretical difficulties with a strictly hierarchical markup approach. Renear, Hockey and McGann (1999) explore the ontological status of text markup, in the context of overlapping hierarchal structures and highlight that text markup may serve two discrete roles:

- The description of the structure of the content for purposes of presentation.
- The semantic identification of semantic structures in the text, which may be entirely non-hierarchical and can include very complex networks of symbolism.

SGML is in this sense antithetical to WYSIWYG as a method for content capture. SGML privileges the formal structural definition of the content. WYSIWYG privileges the final expression.

SGML has been most successful where a strong community supported the elaboration of a DTD that achieved widespread adoption - for example the Text Encoding Initiative (TEI) allowing the critical mass to justify the development of specific parsing technology. The very flexibility of SGML was an obstacle to its adoption as a universal approach to text markup. Its weaknesses for such a role include the overloading of syntactical conventions such as minimization features and the resetting of the meaning of any delimiter, and parsing peculiarities such as 8-character naming limits (DeRose, 1999). While SGML notionally supports multiple hierarchies through the CONCUR device (Barnard et al., 1995), implementations of the concept have proven complex and frustrating, a limitation that also applies to XML. First HTML and then XML emerged with a more limited markup structure in order to simplify the process of markup and content parsing.

HTML is the document architecture of choice for web-based presentation of content. The accessibility and ease of markup offered by HTML have made it both popular and the pre-eminent standard for encoding text in websites. Client Browser implementations are generally marked by their simplicity, robustness to errors in markup and ease of content development. Many authoring tools are available and it is widely understood as a markup method. Unfortunately, quite early in its use, procedural encoding was introduced as a quick and simple means for the specification of fonts, colours and styles. The <font> tag with explicit presentational directions, the use of font descriptors such as <b> (bold), <i> (italic) and <u> (underline) are all typically applied in HTML as procedural markup code. The embedding of such coding in HTML has limited the portability of web content to other environments, although in many cases client email and text authoring tools have been modified to support HTML content presentation. While HTML, the errant child of SGML, established a central role as the markup language of choice for web content, the essential principles underlying SGML have consistently struggled to emerge. The World Wide Web Consortium (W3C) has “deprecated” much of the procedural coding elements of HTML, these elements remaining popular use, and the forgiving

nature of most web browsers has facilitated the popular adoption of HTML (Haustein & Pleumann, 2000).

XHTML, based on a strict DTD, is achieving gradual recognition in industry practice. The XHTML definition presents a strict SGML-compliant DTD, which is fully procedural and honours a more standard SGML encoding structure (Raggett, Le Hors, & Jacobs, 1999). Another rejoinder to the W3C in its effort to reclaim the separation of content from presentation in HTML is the cascading style sheet (CSS). Cascading style sheets allow the creation of a separate style guide that may be consistently applied to all web pages, and the HTML can, theoretically, be cleansed of specific presentation markup tags. While conceptually sound, the CSS has been plagued by browser incompatibilities, varying browser defaults and software versional differences in implementation over time: all profound examples of software obsolescence. It has taken many years for CSS to achieve a general industry acceptance, and many HTML pages continue to use a hybrid set of CSS and procedural markup elements to achieve an acceptable level of presentational consistency.

In parallel with the electronic publishing of documents, commercial document management systems facilitated a set of standards for digital document description. Historically, Electronic Data Interchange (EDI) has relied on a strict document ontology designed for a particular domain. Retailers have long relied on EDIFACT and ANSI X.12 for exchange of business documents. EDIFACT defines the document structures for the interchange of invoice documents, delivery documents, payment records and receipts. EDIFACT, like ANSI-X12, is a highly detailed specification oriented specifically for the interchange of commercial transactions. The controlled vocabulary is therefore also very structured. These structured document formats have the attraction of a well-standardised ontological framework. However, such a framework has not necessarily assisted the uptake of EDI, which remains slow for many reasons, not the least of which are behavioural resistance and trust issues (Love, Irani, Warren, Coldwell, & Warren, 2001).

EDI gateways have traditionally been expensive value-added services. The highly structured nature of the specification has limited their utility in the interchange of complex or loosely defined document entries (such as book records). EDIFACT and ANSI X.12 use a terse code-based coding format in the traditional 80-column with header totals, line details followed by summary totals.

The early EDI gateways, with proprietary exchange technologies, proved expensive and were confined to larger business and government systems. However, even variable content forms such as bibliographic data have been exchanged using the BISAC document standard, a variant of ANSI X.12 oriented to bibliographic data interchange (Boss, 2000). The transactions in these EDI formats are terse and efficient, particularly when compared to the protocol-heaving XML exchange ontologies. However, the minimal encoding used in EDIFACT places a greater burden on the programmatically encoded specification of the ontological framework, and the development of parsing engines is correspondingly more complex.

XML was developed initially as an SGML-based encoding standard for information interchange between applications, primarily for transient data interchange and as a potential successor to EDI standards. However, the definition of a specific SGML-based architecture with an immediately recognisable encoding standard proved immensely popular. XML axiomatically defines the encoding standards for the tags, and focuses on the breakdown of the elements of the information.

XML has the advantage of giving a universal structure for the hierarchical traversal of textual and other content in a reliable manner. An XML document will comprise two fundamental elements: the content itself, marked up using “Elements” and a structural definition, the DTD or XML schema, describing the syntactical usage of Elements, and the hierarchical structure of Elements.

The DTD defines the strict hierarchical standards for construction of a particular XML content set - essentially the markup rules for a particular context. The open nature of XML allows flexible, hierarchical, document structures. While such flexibility is advantageous for the one-off design of content exchange formats, this flexibility is an issue when XML documents are shared in a wider community of systems. The inclusion of a "namespace" resolves a final dilemma – the potential for conflicts in element naming between two different XML sources. The XMLNS attribute instructs the parser to interpret any elements prefixed with the abbreviated code supplied using the nominated DTD. The location of the DTD must be a URL. Obviously, such structures depend greatly on the accessibility and persistence of the nominated namespace URL.

Apart from the ease with which parsing systems can interpret XML, its attraction as a markup language is the lightweight simplicity of the encoding format. While a DTD is valuable, the tags themselves carry considerable information on the ontological structure of marked up text both implicitly in the naming of elements and explicitly in the hierarchical ordering of the elements.

Its success over the more generalised SGML lies in the simplicity of the tagging structure. XML makes axiomatic much of what would otherwise have to be defined in a DTD in SGML – the tagging structure in particular. XML is now understood as the pre-eminent method for text information interchange between systems. It has also gained wide acceptance as the preferred text markup strategy.

The privileging of content separation from both structural markup and stylistic presentation has been accompanied by a sustained theoretical critique of WYSIWYG as a method of authoring. Sowa is representative of such criticism, coining the term WYSIAYG: "What you see is all you get" (Sowa, 2000, p.55). However, two proprietary formats for electronic publishing have widespread Internet usage: Microsoft Word and Adobe PDF (portable data format). Microsoft Word has such a presence in the desktop that many CMS products inevitably face the

challenge of integrating WYSIWYG capability with their systems (Byrne, 2003a), despite the challenges that this can present to good separation of content from presentational style and structure. PDF has also achieved widespread use as a method of content distribution where a high fidelity in the print presentation of the content is also required. It also enables security over the end-user capability to print, copy or view particular content elements.

2.3.3 Multimedia Document Ontologies

The organisation of complex documents containing elements of text, image, video and animation have emerged as an essential element in the final delivery of content. The production of technical manuals for delivery of training courses was an area of early innovation in multimedia document architectures. DSSL is an ISO-specified generalised document transformation language for operation with SGML documents (ISO, 1996). Like XSL, DSSL supports a style sheet and a richly defined transformational language. Day, Liu & Hsu (2001) explore DSSL for content reuse of existing documents in the production of multimedia training manuals. DSSL remains popular in a narrow domain, but it has been vastly overshadowed as a document transformation standard by XSL.

SMIL 2.0 represents the fruition of the W3C efforts to define an XML framework for special and temporal integration of autonomous, heterogeneous multimedia objects in a coherent presentation. The logical design of SMIL provides an attractive and elegant framework for presentational delivery of complex media objects (Bulterman, 2001). Bulterman is careful to differentiate SMIL from content media types such as Flash, situating SMIL as the temporal ontology for content presentation.

The SMIL document provided a framework for XML description of the execution sequence of multimedia objects making up a full presentation and includes:

<seq> - allowing the specification of sequential multimedia objects

<par> - for the parallel delivery of multimedia objects (this can include the synchronised delivery of text, audio and video).

<excl> - for precise timing of object activation.

The elegance of SMIL is the familiarity of most object references (<img> for images) with an embedded simple scripting language for the management of sequencing.

The major implementation of SMIL in a runtime environment is from Real Networks within their Real Media player. Microsoft chose to embed SMIL in Internet Explorer 6, rather than their own Media Player (Rutledge, 2001).

MPEG-21, like SMIL, is an XML document ontology that can span many multimedia resources across different server platforms. In parallel with the emergence of SMIL has been the increasing complexity of the integrative multimedia content models such as MPEG. The emergent MPEG-21 specification includes many of the elements available in SMIL, and extends the concept of a complex digital document description with the Digital Item Declaration (ISO/IEC 21000-2). It specifies a schema for declaring the markup or composition of digital items.

ZxY is a content model aimed at the generalised treatment of multimedia objects in a presentation neutral manner. Boll & Klas (2001) present a set of definitions for acyclic linked list document trees that can be used for presentation-neutral management of content for purposes of reuse in final published presentations. Their model explores the presentation-neutral management of content in a rich multimedia environment (Boll & Klas, 2001). They identify a content fragment as the lowest unit of information. Their model was implemented as a Sybase data-blade. The definitional framework they build for presentation neutral document structures in acyclic linked structures represents a flexible treatment of the document ontology.

Originally sponsored by the US National Institute of Standards and Technology, Open eBook is an XML standard for markup of multimedia documents for use in Digital Talking Books (Open eBook Forum, 2001).

2.4 Content Discovery and Transformation

The multiplicity of standards for dealing with complex document structures is complicated even further when dealing with multimedia objects. Many Content Management Systems continue to use the O/S file system repository to locate their multimedia resources because of the complexity of database management of these resources. The complexity surrounding the management of unstructured text documents is further compounded when dealing with the heterogeneous formats of the digital multimedia document, particularly where the content encoding may not be amenable to simple fragmentation. Some multimedia documents, if in proprietary encoding format, may need to be treated as complete document objects (for example PDF or Word documents) or may be pre-processed to extract content and metadata. The result, once again, is a challenge for content reuse:

> Many high-quality multimedia standards are being created to meet the needs of different communities. However the result is an array of standards offering parallel or competitive solutions. (Burnett et al., 2003, p.60)

2.4.1 Metadata Standards

Content encoding addresses the storage of content in a digital environment. Metadata standards are the means by which content is described for purposes of ongoing management, discovery and distribution. Section 2.2 introduced the importance of "selection" as an element of software reuse. Reuse of content equally relies on effective content discovery. Metadata is the encoding of descriptive terms that typify or characterise the content that they describe. Metadata enhancements can serve both to aid the end-user discovery of content through runtime engines and the discovery of content for purposes of content reuse. For instance, subscribers to a syndication service will characteristically identify particular interest areas. These interest areas will generally be in the form of a "controlled vocabulary" - for example geographic region, subject area, etc. The discovery of content that matches these interests relies on effective metadata enhancement. The development of effective Metadata standards for content is an area of active

research, as is the development of automatic classification systems to parse and describe content in a structured manner. A further challenge to the software developer is that of eliciting the interest and participation of the author in the semantic annotation process, where such annotation can seem an additional burden to the authoring process.

MARC is one of the earliest widely adopted digital metadata standards. Libraries have several decades of experience in the interchange of metadata relating to documents. MARC remains a key encoding formation for library cataloguing metadata. MARC uses hierarchically structured numeric codes to tag descriptive information such as author, title, publisher, and so on. Like XML it is highly suitable for incomplete and loosely defined material. It draws on external standards such as AACR (Anglo-American Cataloguing Rules) and Library of Congress Classification for elements of its controlled vocabulary. No matter how popular many syndication systems using XML are, there are few that can demonstrate such proven effectiveness as MARC over the long term. The value of a sustained and consistent ontological framework for bibliographic data lies precisely in its ability to provide continuity through generational changes of system architecture:

> A few libraries have used the same database for more than two decades on as many as four different automated library systems; each time transporting it to the new computer system by outputting tape from the old system and loading it into the new. Files that conform to MARC have been transferred without reformatting, nonconforming files have required thousands of dollars worth of processing to support each migration. (Boss, 2000, p.31)

MARC has never had significant use outside its base for library systems, and is firmly grounded in bibliographic description. Even booksellers have tended to prefer the use of EDI architectures (such as BISAC) to the MARC format for exchange of bibliographic information. A move to recognise MARC in an RFC (RFC 2220) has been made by Library of Congress. While it could be argued that its tagging system is less intuitive than XML, its success as a vehicle for metadata exchange over very large bodies of documents is possibly unrivalled. There is a recent move to establish a DTD "MARC as XML" for MARC content interchange (Guenther, 1997).

Library Systems still place USMARC as the central standard exchange of bibliographic metadata information (Boss, 1995). This illustrates the advantages that can be gained from a long-term stability in the document exchange ontology.

The Dublin Core Metadata Initiative was formulated to encourage consistent meta-identification of website publications (DCMI, 2001). The encoding standard is described in the W3C RFC 2731 (Kunze, 1999). The elegance of the Dublin Core lies in its minimalism and the ease of encoding within the existing HTML metadata framework, through which it has achieved widespread (if not consistent) use. Its elements are more or less similar to standard HTML metadata with extensions to more fully describe the content, its ownership, copyright and classification.

RSS is a widely used XML specification for website structural description. Netscape originally developed RSS for web site mapping. RSS, like the DCMI, has achieved a success borne of its minimalism. It has been a popular content syndication vehicle, and the XML definition now focuses on this element of the standard (Libby, 1999). It defines a very simple XML format for describing a web page - its title, URL and summary content as a minimum (see Figure 2.3 below).

From its origins as a “site map” RSS has evolved into one of the most popular syndication techniques, particularly popular for news syndication. The emergence of RDF has seen a recent release of RSS 1.00, a rework of RSS in RDF syntax explicitly intended for syndication (and dubbed “Really Simple Syndication”).

```
<?xml version="1.0"?>
<rss version="0.91">
 <channel>
  <title>Montageworld Projects</title>
  <link>http: //www.montageworld.co.uk/</link>
  <description>Internet curriculum projects for teachers. An education initiative from the British Council</description>
        <item>
        <title>Flat Stanley Project</title>
        <description>
        This is a very simple activity. Students make a paper Flat Stanley and mail it
        </description>
        <link>http: //www.montageworld.co.uk/projects/showcase/Flat_Stanley_Project.htm</link>
        </item>
 </channel>
</rss>
```

Figure 2.3: RSS Example

The RDF specification aims to provide a formal model using directed graphs to describe the semantics of metadata and of cataloguing web-based resources (Lassila & Swick, 1999). RDF is very much focussed on the semantic interpretation of the web content and relationships between content items – directed particularly to content discovery (Steinacker, Ghavam, & Steinmetz, 2001). RDF is the core metadata descriptive framework adopted by the Semantic Web movement (see OWL below). Unlike DCMI and RSS, both of which are explicitly oriented to website content, RDF is a generalised semantic structure for describing what can be complex website resource descriptions. While the principle momentum for RDF comes from the Semantic Web, its practical implementation has been at a more minimal level in the adoption of RDF as part of the RSS 1.0 specification. The generalised vocabulary of RDF, like SGML, has limited its adoption, and has seen rival approaches to semantic description such as Shoe (Heflin & Hender, 2000).

The IEEE Learning Object Metadata has gained some popularity in the Information Technology community for the encoding of curriculum resources (Steinacker et al., 2001). The ontology is not dissimilar to the DCMI metadata set, with extensions for the specific domain of learning resources. Of particular interest to this thesis is the explicit recognition and encoding of metadata to identify the "granularity" of the educational objects - the "aggregation level". This is defined by a scale of "1" for raw media data) to "4" for collections of documents (IEEE, 2002).

There have been recent efforts to align the IEEE LOM with an XML specification and RDF specification.

The SCORM framework is similar to the IEEE/LOM, but providing a richer framework describing the metadata ontology not only for content description but for its framework for delivery of Learning Objects, with a particular focus on effective content reuse and a model for integration of content across different systems. It goes as far as defining application interfaces and architectures for use of the SCORM model and has an XML-based API. Like the IEEE LOM, it explicitly recognises the granularity of content that it addresses (Advanced Distributed Learning, 2001). SCORM is also an advanced metadata ontology specifically for reuse of learning objects. Developed by the US Department of Defence, it has a particular presence in Governmental organisations. SCORM is examined further in Section 2.6.5 below.

The Semantic Web is one of two key initiatives emerging from the W3C in recent years, the other being Web Services. The Semantic Web seeks to provide an ontological framework for effective semantic description and discovery of web-based resources. This ontology is particularly realised in the Web Ontology Language "OWL". OWL defines an ontology for describing content structures with an emphasis on achieving better semantic uniformity, and therefore transforming the web into a more effective knowledge base.

The Semantic Web can broadly be described as a movement to a consistent content description and discovery ontology for the web. RDF is a key element of the Semantic Web framework, as is the metadata ontology OWL (see Section 2.4 above). A further element in the Semantic Web programme is the W3C working draft query language construct called XQuery. Published in June 2001 (as a draft standard), XQuery constructs for sorting, query constructs and logic. It is conceptually closer to the relational query language than a Z39.50 search object (Chamberlin et al., 2001). Nevertheless, it may offer the platform for implementation of a standardised vocabulary for search definition as it comes into maturity. The persistence of search

results over time and an effective distributed vehicle for search processing would complement this standard. When dealing with large, heterogeneous, text resources such as the Internet, another major challenge is the heterogeneity of document classification systems and controlled vocabularies, or the lack of such metadata enhancement entirely. The developers of the Google search engine have demonstrated how effective automated classification and the use of linkage information can be used in the delivery of smart search results.

The experience with software reuse emphasises the importance of building effective abstracted models that can be relied on over time. This particularly applies to the arms-length exchange of content. The versatility of XML belies the difficulty of agreeing on standards in the document ontology supporting the XML for exchange. The example of MARC illustrates the effectiveness of reliable document ontological frameworks within particular communities. While the emergent ontologies of the Semantic Web such as OWL may change the situation, the current trend is for particular communities to develop or adopt ontologies for their own particular needs.

2.4.2 Content Discovery

The importance of content discovery has already been discussed. Content discovery can hinge on several factors: adequate metadata description of the content, use of effective automatic classification systems, and intelligent search engines. There are remarkably few standards that govern Information Retrieval, and yet content discovery is of great importance to most web publishers. The multiplicity of content markup and semantic description standards has compounded this difficulty. Very few institutions have sufficiently large heterogeneous distributed collections that require a commonality of search interface. Yet this is an important component of content reuse over large content databases. The poverty of many search engines (against the capability of a few) is illustrated by the fact that many companies index their own sites using the public Google search engine.

XML has been widely adopted as a vehicle for content exchange. This wide adoption is both proof of its effectiveness as a simple vehicle for information interchange, and of its weakness – the speed with which competing standards for information description can emerge. Content exchange has been of particular interest to news organisations, website portals and website content publishers. The result is a heterogeneous set of industry-sponsored XML definitions, many with overlapping ontological structures but with little in common in their DTD. If each of these specifications targeted distinct content areas then the existence of overlapping specifications would present little difficulty. However, specifications such as NewsML, PRISM (discussed below) and RSS (a metadata format which has evolved into an exchange format discussed above) serve essentially similar purposes, backed by different sponsoring industry consortia.

The XML-RPC standard is an early form of Web Services definition that has reached popular acceptance for provision of "phone-book"-style lookup services. It remains a (functionally) lightweight and easily implementable Web Services approach that is appropriate for information interchange where personalisation and security are not elements. Web services approaches such as XML/RPC have the capability of accessibility and transparency through proxy servers and firewalls – both a strength and a weakness of the approach. XML/RPC reuses the HTTP protocol framework and XML, with the resulting verbose protocol layer (St.Laurent, Jonston, & Dumbill, 2001).

The emergent industry of Web-services based document interchange relies on a more established XML-based specification for communication with web sites - SOAP (Simple Object Access Protocol). Web services add to SOAP layers of service discovery and service description, both important components of robust content exchange (Chung, Lin, & Mathieu, 2003). Building on the SOAP API for web-based functional and content exchange, Web services look to be an interesting framework for distributed content delivery. In particular, the Web services implementation leverages the Internet browser client platform to provide a rich application communication framework with host services that can readily traverse the Internet. Web Services

extends this framework to provide facilities for resource discovery and as standardised service description. As such it is an attractive framework for the development of services that are delivered in an Internet framework, and has seen rapid adoption by major suppliers as a vehicle for encouraging service-based subscription to their products. A Web Services Description Language (WSDL) ties the services description together in one document ontology (Ogbuji, 2003).

Web Services inherits several intervening XML ontologies, and adds four more layers of its own. In that sense, WSDL documents have become distant from the self-explanatory roots of XML markup, and are often generated rather than hand crafted. The protocol burden of Web services is also much larger than explicitly defined and terse EDI exchange ontologies such as EDIFACT (Geng, Gopal, Ramesh & Whinston, 2003).

The rich melding of service discovery, service description and content exchange exemplified by Web Services make it likely to be the model for ongoing development of arms-length content exchange systems.

Early keyword-oriented crawling and indexing engines (WAIS with some input from Z39.50 protocols) are the conceptual precursor to current Internet search engines (Needleman, 2000). Gopher, WAIS and WWW protocols were all released in 1991. Of these, the only effective survivor is WWW, having subsumed both Gopher and WAIS in current Internet search engines.

Internet search engines have opened a new frontier in document indexing and classification, particularly in the way they take discovery techniques to a new scale and explore indexing concepts in the context of the massively hyperlinked environments (particularly the Internet). Search engines are remarkably successful in abstracting underlying metadata and resources based on linkage information, semantic metadata and contextual information of the site itself. Search engines touch not only on the content and metadata of web pages, but are also beginning to

explore the underlying database models behind the website resource - a kind of large scale reverse engineering (Cope, Craswell, & Hawking, 2003). Part of the credibility of web search engines lies in their ability to yield useful results rapidly across a very large collection of documents. Their popularity stands despite the fact that their breadth of coverage rarely exceeds 10% of sites, and despite the failure to build search interfaces which reflect lessons in search design that have already been learned (Shneiderman, Byrd, & Croft, 1998). Apart from the semantic information represented by formal encoding of metadata on the page, and the text content of the page, search engines also make extensive use of linkage information to derive useful ranking information for the page - put simply, the more a page is pointed to, the higher its value as an item to present in the retrieval list (Page et al, 1998; Craswell, Hawking, & Robertson, 2001). Such approaches can also help the design of authoring systems to facilitate the elicitation of semantic data from the author, and were an element of the prototype research in this thesis (see Chapter Five).

Z39.50 is the only specification for provision of distributed Information Retrieval services that has a demonstrated track record across service platforms and networks. Z39.50 sets a standardised protocol for state-based response to complex search queries in such a way that search results can persist over several searches. The consistency in the query formulation protocol allows the establishment of a federation of search servers, which are consistent in their protocol level of operation. It defines a generalized search and document delivery protocol that allows definition of domain-specific *profiles* that specify the query syntax, limitations and content structures for query results. A key element of Z39.50 is the ability to maintain persistent query states between query sessions. Accepted as ISO23950, the standard is widely used in the Library community for distributed catalogue resource sharing. It has had only marginal interest outside this community of users, and lacks any significant uptake by those developing internet-based retrieval engines (Needleman, 2000).

Z39.50 is best suited inside the library domain for bibliographic information resource sharing. However, its architecture points to features that could be improved in current Internet-based information retrieval engines, particularly:

- A standardised query protocol supporting distributed processing of searches across multiple systems.
- The ability to persist search content queries and their resulting document collections.

2.4.3 Content Sharing and Syndication

Capitalising on a growing standardisation in methods for metadata description using XML, a number of standardised approaches to content sharing and syndication have emerged, particularly to support the transition form traditional methods of news-wire syndication to digital syndication of content.

The International Press Council sponsors NewsML, a widely used specification for syndication of news items. Developed by Reuters and adopted by many international news agencies (such as Agence-France Press) this is a format for distribution of syndicated news content. Its controlled vocabulary predates XML as a specific encoding method. The DTD for NewsML can be used to describe all types of multimedia. While NewsML is generally deployed between traditional news organisations, IDEAlliance sponsors a number of proposed industry standards, with a particular focus in the area of content management.

In particular, two specifications have wide industry usage:

- ICE (Information and Content Exchange). A convention for the protocol-level exchange of information relating to syndicated content. Recognises a subscriber and a publisher and defines XML-based information on currently available content awaiting collection, and its status. ICE purports to be a simple, easily implementable standard for interchange of information regarding the status of syndicated content (Souzis, Popkin, Khoury, & Hunt, 2000). ICE does not specify the format of the content; it merely acts as an XML-format protocol

operating in an HTTP (HyperText Transfer Protocol) framework for the content interchange process between the syndicator and the client.

- PRISM (Publishing Requirements for Industry Standard Metadata). An XML DTD focussed on the ontological specification of semantic information required to facilitate content reuse. The PRISM standard is designed to facilitate the semantic recognition of editorial and classification information of content items. Unlike NewsML, PRISM makes extensive use of Dublin Core in its specification. It is more generalised in scope and so less specific in detail than the equivalent DTD for NewsML (PRISM Working Group, 2001).

2.4.4 Content Transformation

In HTML, most browsers will make a reasonable effort to interpret the tags that are used to mark up the content, because a fixed set of known tags are defined. In this sense, irrespective of any style sheet, the user defines an implicit style sheet to HTML. XML has no pre-determined tags. XSL is the style sheet for XML. Like CSS, XSL has a set of rules that determine the stylistic presentation of XML in a way that removes the need for procedural markup. Unlike CSS however, XSL is a template-oriented solution with some scripting-like constructs for sorting, selection and looping based on conditional constructs. XSL and Content Style Sheets (CSS) are deployed for rendering XML in a presentational form. XSL extends the capability of the CSS not only the rule-based presentational information (available also to CSS), but also a "scripting" language for much more structured control of the presentation. XSL was accepted for W3C for recommendation on 16th Nov. 1999 with XSL 1.1 currently in working draft. The elements of XSL are XSLT (a language to transform XML to a final presentational form) and XPath (a scripting-like language for conditional selection and sorting of content). XSLT is used to transform an XML document into a final presentation format, or indeed, into another XSL-format document. A popular use of XSL, for instance, is to render XML into HTML so that the content can be viewed in a browser. This may include the addition of headers and footers, and may involve the sorting or re-arrangement of the source XML content. It is a hierarchical transformation:

> A transformation expressed in XSLT describes rules for transforming a source tree into a result tree. The transformation is achieved by associating patterns with templates. A pattern is matched against elements in the source tree. A template is instantiated to create part of the result tree. The result tree is separate from the source tree. The structure of the result tree can be completely different from the structure of the source tree. In constructing the result tree, elements from the source tree can be filtered and reordered, and arbitrary structure can be added. (Clark, 1999)

XPath maps the source document to one or more predefined XSL templates based on selection rules. XSLT does the work of then rendering the content in its final form. Such a parsing process is common to all Content Management Systems, whether implementing a template-based approach or an XSL approach to rendering the final form of content.

Internet Explorer 5.0 (and subsequent versions of IE) embedded XSL parsing features in the client web browser. It is possible with IE 5.0 to send XSL and let the browser render this in the format that has been selected. It is important to note however that many other browsers do not yet support XSL. For this reason many XSL transformations are done on the Server rather than by the Client browser for those systems currently implementing XSL transformations.

XSL remains the principle standardised content transformation language. Its industry acceptance is, however, constrained by issues of complexity of the scripting design and relatively poor performance in dynamic content generation. While much of the theoretical discussion of content transformation centres on XSL, the industry review in Chapter Two illustrates the ongoing popularity of proprietary scripting and template-based tagged insertion approaches to document transformation.

2.4.5 Content Reuse = Multiple Ontologies

The drive to automatic classification is in part related to the difficulty of effective semantic description of content. Increasingly complex ontologies have emerged, such as DAML+OIL - a merging of two other ontologies comprising DAML-ONT (DARPA Agent Markup Language) and OIL (the Ontology Inference Layer) (Fensel, van Harmelen, Horricks, McGuinness, & Patel-

Schneider, 2001). The more complex the ontology, the greater the challenge eliciting participation from the author - as Haustein and Pleumann (2002) observe in regard to the Semantic Web:

> Simplicity is only partially given. The mixture of RDF and DAML+OIL is understood in all its details only by people that have a background in AI or related fields. Novices will only be able to use basic concepts of RDF and might thus have problems to see the real advantages of the Semantic Web. (Haustein & Pleumann, 2002, p.1)

There is little doubt that the drive to interoperability and scalability implicit in the goals of the Semantic Web will continue because the benefits in the resulting knowledge discovery are attractive. While there are emergent signs that RDF will play an important role in metadata description, it could not yet be said that a satisfactory metadata ontology yet exists which is embraced by all stakeholders. The Semantic Web has yet to see significant organisational or industry backing. Described as a "vision" still in progress (Lu, Dong, & Fotouhi, 2002), lack of satisfaction with progress with the Semantic Web has seen some emergent criticism of the slow progress (Ogbuji, 2003), and the development of alternative strategies (Heflin & Hender, 2000). There are also clear industry efforts toward consolidation to a few, well accepted encoding and metadata annotation standards for documents.

Regeneration of content can benefit from standardisation of approaches to content encoding, while content reuse can be improved through the industry adoption of standardised document ontologies within given communities.

> *Principle #4 - Content syndication and reuse rely on agreed ontologies for the description of semantic content. Industry implementations remain domain specific, but there is a convergence toward RDF. The ability to address multiple semantic ontologies remains important.*

These issues of content discovery and exchange lead to three principles relevant to content regeneration and reuse:

> *Principle #5 – A consistent document exchange ontology with strong community support is an essential prerequisite to content exchange. The CMS should*

accommodate the flexible mapping of content and metadata to accepted document ontologies.

Principle #6 - Stability of the document ontology over time will improve content discovery and reuse.

Principle #7 - Content discovery is facilitated by:

- *Standardised semantic descriptors (controlled vocabularies) supplementing the original content.*
- *Support for complex query structures.*
- *The ability to persist search content queries and their resulting document collections.*

The emergence of rich, but competing, standards for complex multimedia documents, is indicative of the need for ontological flexibility for content reuse. The multimedia publisher may well have to map to one or more overlapping ontologies. The description of the content reuse objects should be amenable to producing content to suit these different published multimedia content models. The result is a need for flexibility in the Content Management System in dealing with a heterogeneous mix of encoding forms and document models for the final published expression. Some will be tractable to content fragmentation at a more granular level, while some may have proprietary formats that discourage such fragmentation into content units.

The discussion in this section leads to an important principle in the context of content regeneration and reuse:

Principle #8 - Multimedia content reuse characteristically requires flexibility in the support of a heterogeneity of document models for authoring, publishing and syndication.

In many instances the transformational task is customised through programming or Operating System workflow scripting to achieve a particular publishing task. These custom transformations may be particular to a system context and may themselves involve stages of manual intervention. Content reuse make require transformation for single sources to multiple

different presentational forms. Content regeneration may take disparate resources through several phases of transformation and with a greater focus on the workflow for this transformation toward a single published resource. For instance, the EBEL project at Sydney University explores the multi-stage publishing transformations, which are required to generate a digital Basque Grammar from original text and audio source documents (Patrick, Balnaves, Kam, Ler, & Yeates, 2000). Nor is there a single transformation approach suitable to all media types, some of which involve proprietary encoding methods.

The Content Management System must therefore expect to accommodate a task-oriented approach to transformation with its own workflow process:

> *Principle #9 - The transformation engine should be designed to support a flexible, possibly multi-stage, sequence of transformation processing - that is, a flexible generative process. This can be regarded as the "publishing workflow".*

2.5 The Systems Context

The management of multimedia content can be considered the preoccupation of at least five categories of industry systems:

- Document Management Systems
- Digital Library Systems
- Learning Management Systems
- Knowledge Management Systems
- Content Management Systems.

It is relevant to analyse the degree to which these product appellations represent primarily a marketing segmentation or truly different conceptual models for management of digital content. Each of these categories of systems has an evolving body of research literature. This section will explore a comparative analysis of the functional differences between these systems.

2.5.1 Digital Library Systems

The Library Science focus on consistent classification and fidelity of presentation has seen a cautious approach to the development of Digital Library Systems. Concern has been expressed both for their effect on research methodology (Crawford & Gorman, 1995; Downs, 1999; Jewitt, 2000; McGann, 1998; Weiderhold, 1995) and with technological issues of obsolescence (Crespo & Garcia-Molina, 1998; Smith, 1997).

Notable developments in the public release of Digital Library resources include the impressive Library of Congress site (Marchionini, 1998) and the Blake archive (Eaves, 1997). Both libraries have directed special attention to the realisation of an effective delivery platform in the process enhancing research through the online paradigm of delivery. The issues of cost-economical delivery of digital library content is profoundly important to the library community, in a context where the focus remains on equity of access and freedom of movement of information. There are considerable costs entailed in the presentation of content in digital form, both in the authoring process and in the delivery of the service. Issues of persistence are therefore particularly important in the context of Digital Libraries (Crespo & Garcia-Molina, 1998).

Significant international digital library resources are also emerging, an example being the "Networked Digital Library of Theses and Dissertations" (NDLTD) which provides both metadata search resources and a union list of masters and PhD dissertations with a large international membership of universities (including University of Sydney). The focus of the NDLTD is illustrative of the role of the Digital library - with a strong concern for cataloguing and preservation having a primacy in the concerns for the NDLTD service (Phanouriou, Kipp, Sornil, Mather, & Fox, 1999).

2.5.2 Content Management Systems

The term "content management" is a succinct description of the demands of managing multimedia resources. The term "content" is sufficiently general to describe any digital form,

media or object ("content" in the sense of a descriptive label of items that are "contained within"). It also implies the components of meta-information storage (in the sense of the "table of contents"). The term "management" can be extended to imply not only the operational management of the publishing exercise, but also the long term organisational management of the content store, the effective use of content through regeneration, repurposing and syndication, and the long term archival administration of content.

CMS products have evolved in the first instance to reduce the technical demands of publishing to the new media (Celentano et al., 1992). The new media places a premium on currency of information, creating a need for a framework through which website content can be easily updated. The technical aspects of publishing, including the mechanics of delivering changes to the website (FTP, WebDAV, dynamic delivery, etc), can be intimidating to the content author. Finally, as web-based publishing permeates an organisation the importance of the CMS increased as a response to maintaining consistent design standards. The editorial control of information content and presentation may not be an issue for casual web publishing, but has increasing importance as web-based communication gains the same standing as the printed media.

Apart from utilitarian sites (such as banking gateways), the attraction of a website (in an e-commerce sense) was founded on what effective content value it could offer (Belle, 2001) - although Odlyzko (2001) cogently argues against the elevation of the place of "content" as a perpetuation of older paradigms of mass communication, and that the Web, in particular, represents more closely a form of individual communication. While the DotCom community is now littered with stories of commercial failure, many of its successes are founded on the effective reuse of content through syndication, cost-efficient personalisation and facilitation of site make-over (Dalton et al., 2001). The Australian Broadcasting Corporation, for instance, uses a Content Management System (developed by the author) to both publish its website, generate its website navigational aids (subject and site indexing) and to syndicate the resulting content to other organisations with a database of more than 400,000 content items.

The content author will often address a particular audience, and neither the content author nor the intended audience are necessarily concerned with issues of content encoding and markup. Neither is the author generally interested in the mechanics of content delivery. However, the effective separation of content from presentation and the elemental identification of content components have overwhelming value for the publisher, researcher and archivist. In this context, there is little hope of imposing a generic standard for content capture, given the separate relationships that are implied in the publishing process. In book publishing, the traditional role of the bookseller and library was as a mediator between the content author, publisher, researcher, archivist and finally, of course, the reader. The particular role of the library was to organise and classify content in a manner that made it effectively accessible to its client population of readers.

There are factors affecting content reuse that go beyond the mechanics of identifying content components for reuse. Overby (1999) argues that authors compose from a particular media, and the fragmentation of content risks losing the stylistic and overall semantic direction of the author. These criticisms are cogent and mirrored by the discussion of the limitations of XML as method of markup. They indicate a contingency in the useful application of concepts for content reuse that must be reflected in the authoring process and considerations of reuse at time of publishing. Content reuse in the context of regeneration still has considerable value, even where the content maintains its essential integrity but is regenerated as a result of graphical presentation makeover or versional changes. This contingent element in content reuse is reflected in the areas where content reuse has been most effective: especially news syndication, online learning, and versional regeneration of websites.

The paradigm of WYSIWYG is highly intuitive for the content creator, but stands opposed to the structural need to decompose content components and the stylistic components for purposes of long-term management. WYSIWYG content editing features are universal to commercial CMS products (see Section 2.7). The expressive power of WYSIWYG pertains to the natural

inclination of the "user" to operate in and with the final expression rather than the semantic or structural markup.

Websites typically go through several generations of design, sometimes with considerable stylistic change but little change to the underlying semantic content. The transition of stable content through intergeneration graphical redesign of websites simply emphasises a trend already implicit in the early needs of Content Management Systems: the separation of content from style and presentation.

The mantra of Content Management Systems from early times has been "content separation from style and structure". There is now sufficient published discussion of the Content Management Systems to discern key functional elements in common to the systems view of content management. Abbey, Ellis, Suh, & Thiemecke (2002) characterise the CMS in terms of "workflow", "transformation" in the context of an "asset" based strategy. Fraser, (2002) discusses the design of a CMS in terms of "version control", "workflow" and "personalisation". Rockley has an "enterprise" view of content with perhaps one of the widest views of the CMS as it is situated in the organisation, but emerging once again are the common themes of "workflow", and "delivery systems", including content transformation (Rockley, 2003). A detailed functional dissection of content management systems appears in Appendix One.

Consistent across the greater proportion of Content Management Systems are functional principles that pertain to effective content reuse:

> *Principle #10 - The end user should be masked as far as possible from the technical aspects of the publishing process to encourage currency of content.*

2.5.3 Knowledge Management Systems

Knowledge Management is itself a convergent discipline that has seen the integration of expert systems and AI enhanced information classification and retrieval. Much has been written on the strategic importance of knowledge within an organisation and the processes needed to

describe and disseminate existing knowledge and to elicit new knowledge (Liebowitz & Wilcox, 1997). Knowledge management systems are *organisationally* focussed and *problem* oriented (Liebowitz & Wilcox, 1997) and must often deal with the information retrieval and the classification of loosely structured text documents, particularly in the context of Internet and other search engines. Others have looked for a consistent methodology for annotating and mapping the conceptual structures implicit in documents (Gruber, 1993). Valuable motivators in the evolution of techniques for Knowledge Management are the economic cost of formal classification of documents through traditional classification approaches and the economic benefit arising from efficient discovery of highly relevant material. Business consulting firms have led the way in uptake of Knowledge Management Systems.

Knowledge management shares several common causes with Content Management. Knowledge management has a particular focus on the effective identification (automated or manual) of semantic metadata to facilitate knowledge discovery. It is also essentially directed to knowledge reuse: the economic cost of encoding the knowledge resources of an organisation is warranted by the strategic benefit yielded by the efficient application of these knowledge resources in other contexts.

The Ontologies (or formal specification of conceptualisations) used in Knowledge Management are used to map the knowledge structures and provide a consistency in annotation and discovery of documents. Gruber (1993) situated this ontological commitment to common terminology centrally in the development of "knowledge services". Davis (2001) describes the importance of a global perspective of information flows in the design of knowledge management systems in a collaborative workflow setting. In particular, Davis identifies both the importance of the controlled vocabulary for keywords, and the difficulties in encouraging end-users to supply this important metadata information.

The theory of markup properly attributes a role for both structure and meaning (Sperberg-McQueen et al., 2000). However, the inferential power of standard HTML metadata and the DTD for XML have not been sufficient to provide an adequate vehicle for inference of meaning. This difficulty has seen the emergence of more sophisticated document ontologies that "speak of concepts and semantic relationships rather than the elemental nesting or sequential order" (Erdmann & Struder, 1999, p.2). Lauer (2001) situates the knowledge management process as a Hegelian inquiring system – that is, the central role of the Knowledge Management system as an enquiry system that teaches users "how to look" as distinct from a repository of knowledge resources in a product-oriented information paradigm. Althoff, Bomarius and Tautz (2000) similarly locate Knowledge Management as part of the Learning Organisation. That is, the specific focus of the Knowledge management system is in facilitating discovery of knowledge resource. This focus on discovery is characterised by a concern for the semantic identification of knowledge resources and the use of intelligent agents and personalisation to characterise the specific information needs of the inquirer. The semiotic focus of Knowledge Management Systems is on the substance of the content, and the role of content reuse in this context is primarily as a vehicle for fulfilling the needs of the inquirer. Like Knowledge Management, Content Management has a semiotic focus on the substance. But unlike Knowledge Management this is directed toward the effective regeneration and reuse of the substance of the content for purposes of syndication and economically efficient versional regeneration of content – rather than toward effective content discovery.

2.5.4 Document Management Systems

Document Management Systems generally focus on the management of a document in its final form, or digitised in a form that retains reasonable legal and presentational fidelity with the original document. There is considerable current focus on establishing the legal standing of such documents (J. Wilson, 1997). In some cases this will include a focus on the archival issues of

document management (for legislative or other reasons), facilitating the cost-efficient workflow management of large volumes of documents, with demonstrable efficacy (Wright, 1996).

Wilkinson (1998) provides an overview of the nature, use and control of documents. He characterises the document as a *message* (Wilkinson, 1998). He situates Document Management Systems in the high-context role of business enterprise document management. Wilkinson also differentiates the "document" as a finished item from the workflow process of drafting and preparation and characterises the message as comprising of *content*, *structure* and *metadata*. Wilkinson also explores technologies that support document description, production, delivery, publication, discovery and removal – a document "life cycle." Document Management can serve to enhance document discovery in order to maximise Document utility for reuse in the organisation (Song, Clayton, & Johnson, 2002). In this case, the overlap with Library Systems and with Knowledge Management is clear. The deployment of technology in the construction of Document Management Systems will naturally draw on concepts of workflow management and knowledge management. However, Document Management Systems primarily focus on the commercial document assets of a particular institution (Wilkinson, 1998), as distinct from the heterogeneous mix of documents managed in a Library context, and are often concerned with the management of the full "life cycle" of the document from its inception to its archiving or destruction.

2.5.5 Learning Management Systems

A particular class of Content Management centres on Distance Education and Online Learning. The issue of content reuse has been widely recognised as central to the cost-economical development of curriculum materials for online use in almost every educational context (Candler & Andrews, 1999). In particular, educational institutions and publishers have often invested considerable resources in development of online education materials, only to find considerable costs in generational technological changes and in versional revisions to the content. Brett and

Nash (1999), for instance, describe the development of authoring tools to facilitate the regeneration of content across six separate releases of CDROM instructional material in a manner that avoids repeating the high production costs with each separate CDROM production (Brett & Nash, 1999).

The US Department of Defence has funded the Advanced Distributed Learning Initiative (ADL*)* for the specific purpose toward the specific goals of "content reusability, accessibility, durability and interoperability (Advanced Distributed Learning, 2001, p 1-2). The objectives of the ADL centre on content reuse:

> Demonstrate the ability to move Web-based courses from one learning environment (learning management systems) to another; demonstrate the reuse of learning content 'objects' across different platforms and learning environments; provide searchable learning content across different learning environments or media repositories. (Advanced Distributed Learning, 2001, p.1)

Their content model, Sharable Content Object Reference Model (SCORM), is one of the most recent, and comprehensive ontologies for describing curriculum resources. Their model describes a comprehensive XML ontology specifically oriented to Online Educational objects. In the SCORM model, the Advanced Distributed Learning Initiative defines an Application Programming Interface (API) for standardised access to these content objects from disparate content sources in a manner that facilitates their reuse. The purpose of this API is to provide a consistent framework for providers of runtime engines for course delivery (which they refer to as Learning Management Systems) to integrate with a disparate set of learning content repositories).

In building a web-based online learning and courseware system, (Yuan, Abiza, & Karmouch, 1999) similarly differentiate between the media content server and the courseware server, with the courseware server assuming the role of the runtime engine for course delivery, and central to reuse.

Another approach, the IEEE Learning Object Metadata (IEEE LOM) also focuses on content reuse of educational resources. Like SCORM, an XML metadata ontology is central to

the approach to content reuse, enhanced through the use of component-based tools to visualise the course design (Saddik, Fischer, & Steinmetz, 2001). Cameron, Beam, & Beam (2000) follow a simpler XML model for course content generation, presenting the content through a workflow production system for course content that follows a simple standard "bookshelf" metaphor. A consistent theme among the LOM ontologies is the explicit recognition of content fragmentation. The IEEE LOM refers to this granularity of content as the "aggregation level" (IEEE, 2002).

2.5.6 Comparative Analysis Of Systems

Most CMS products have at least some element of workflow management. The life cycle of web-based content is in some respects similar to that of the standard print document, and for this reason it is tempting to categorise the Content Management System with Document Management Systems. While there is commonality in the functions addressed by Document Management and Content Management, particularly in their elements of workflow and multimedia document management, there are also clear differences. In the semiotic treatment of content the Document Management System focuses on the management of the content in its final expression. In this respect, the CMS is akin to the Digital Library System.

The delineation between Content Management Systems (CMS), Knowledge Management Systems (KMS), Document Management Systems (DMS) and Digital Library Systems (DLS) can be mapped across several dimensions. Table 2.1 contrasts these systems across eight critical dimensions in relation to content reuse for the Knowledge Management System (KMS), the Document Management System (DMS), the Content Management System (CMS) and the Learning Management System (LMS) and Digital Library System (DLS).

	KMS	DMS	CMS	LMS	DLS
Content Focus The level of content and metadata fragmentation (from whole of document to part-of-document).	Part of document	Whole Document	Part of Document	Part of Document	Whole Document
Semiotic focus– Whether focussed on the substance of the content (to aid content discovery) or the particular expression of the content (to aid publishing in a particular manner and format).	Substance	Expression	Substance	Expression	Expression
Reuse focus. The focus for retrieval, in particular whether the target is generally the "whole" document or fragments of documents that are useful for the task at hand, can vary according to the nature of its use and reuse.	Exploration on the basis of semantic elements. The KMS may facilitate content retrieval both across controlled vocabularies and free-text terms.	Authoritative delivery of final expression	Similar to the KMS, but with syndication and re-constitution of content in different forms or for content regeneration. Unlike KMS the objective is reassembly of this content as a new resource.	Repackaging and reuse of content units in different curriculum contexts.	Authoritative delivery of final expression
Utility of Content - The utility treatment can affect whether the information system retains only metadata, whole documents or content fragmented in a finely granular way.	"Knowledge" as a commodity – strong focus on the semantic value	Documents as a research resource in a homogeneous document environment. Commercial record keeping.	"Content" as a commodity – strong focus on the economic value of reuse	"Content" as an educational resource.	Documents as research resources in a heterogonous document environment.
Metadata Ontology - and the different approaches to the definition of formal categorizations and markup.	Semantic-meaning focussed, often complex semantic networks	Thesaurus-focussed, often organisation specific.	Domain-specific For instance, the Dublin Core to describe metadata for web pages, or NEWSML for syndication.	Formal ontologies such as SCORM, Learning Object Model (IEEE/LOM).	Hierarchical Extensive use of Controlled Vocabularies & Classification standards – for example Library of Congress Subject Headings, Dewey.
Discovery - The focus for retrieval, in particular whether the target is generally the "whole" document or fragments of documents that are useful for the task at hand, can vary according to the nature of its use and reuse.	Internet Search Engines & Proprietary Clustering and association techniques	Application search engines & Internet search engines	Internet Search Engines	Internet Search Engines, Learning Management Systems	Z39.50 through Catalogue interfaces
Identity and the characteristic method for internal and external identification of documents.	Internally-generated document identifiers	Internally-generated document identifiers	Internally-generated document identifiers	Internally-generated document identifiers	Standards-oriented identifiers description (for example codes of Cataloguing and Classification)
Personalisation of content (for example through personalisation, Intelligent Agents).	Intelligent Agents	Profiles	Intelligent Agents	Learning Management Systems (for example WebCT and Blackboard)	Selective dissemination of Information

Table 2.1: Contrasting Systems Approaches

The concern of Content Management is the efficient reuse and regeneration of content, sometimes in a highly dynamic environment. While there are common challenges to Document Management in all these contexts, the particular challenges of each are founded on this different focus The Document Management System has a particular focus on the integrity of the

management of a document through its life cycle. The Content Management System may be tasked with the regeneration of content in different forms and the distribution of content in document structures through syndication.

Content publishing of complex multimedia can span many client authoring and server delivery systems: "Modern web applications are conveniently described as a hybrid between a hypermedia … and an information system" (Fraternali, 1999, p.228). In the context of Television Production, Heitmann (1999) situates the Content Management System as an Information System central to addressing multimedia management issues:

> "The convergence of audio and video technology with IT and telecommunications technologies promises a number of advantages and new possibilities, but it also raises a number of questions:
>
> a) How can the data resulting from television production and distribution – agency feeds, scripts, rundowns, storyboards, systematically be collected, logically be connected to its associated video and audio material and finally be made available to desktop computers in an organized manner?
>
> b) How can the images and sounds be made available on as many desktop computers as possible without having to equip them with expensive multimedia technology?
>
> c) How can the image and sound material held in different storage devices be administered in a coherent way?
>
> d) How can sounds and images, once digitised, be archived with computer assistance and be kept ready for fast and efficient retrieval?" (Heitmann, 1999, p.24)

The emphasis of the Document Management System is workflow and archival management of the document life cycle; where the content that preserves the essential elements of its authoritative final form. The level of content treatment, that is, the "granularity" of the content, is generally that of the "whole" document. The Information Retrieval focus is on discovery of the whole document and preserving the archival fidelity of the original document. This can be contrasted with Content Management and Knowledge Management, whose focus is on the substance of the communication and the provision of efficient systems for content regeneration and reuse in variant forms and contexts. The concern of Knowledge Management is the effective semantic discovery of useful "pieces" of knowledge.

Content Management Systems can be considered most akin to Knowledge Management Systems, with a key differential focus in the manner of reuse. This differential focus explains the utility value of Content Management Systems in News organisations and for the management of dynamic, complex, websites.

2.6 Current Industry Practice

2.6.1 Context of the Review

There are now several hundred products called "Content Management Systems" available in the market. This represents a staggering rate of evolution in the three years over which this thesis has evolved. This market survey was conducted through evaluation of white papers, product profiles, direct supplier discussion, discussion lists and demonstration versions of software available for download. The survey provides the basis of a functional decomposition of CMS features against the model presented in Section 7.2 and a framework for classifying attributes of current industry practice against the model. The classification of functions against the model also serves the objective of testing the degree to which the model describes important existing elements of industry. It also serves to highlight potential deficiencies in industry practice in the area of content reuse, and provides a context for the functional evaluation of the prototype. A more detailed functional decomposition and document map of supplier resources is provided in Appendix One.

2.6.2 CMS product classes

There are many classes of authoring systems that stake a claim to the title "Content Management Systems", from simple website editors, to enterprise applications.

Fraternali (2000), in a detailed survey of "data-intensive" web development tools (that is, systems for delivery of content from large database sources), classifies existing Website management systems into six categories:

- **Visual Editors** (complex web page editing tools with some site management capability)
- **Hypermedia Tools** (site management tools with no integral database element)
- **WebDPL integrators** (database programming language extensions to existing web infrastructure). In this he classifies Microsoft Active Server Pages, Vignette's Story Server, HAHT Softwares HahtSite, Informix AppPages, Allaire's Coldfusion and others.
- **Form Editors.** Fraternali classes a broad range of existing software design products with extended web-based form editing and generation capabilities in the category of Form Editors. That is, they have the capability to generate schema-based forms and a scripting-based capability to delivery the dynamic content. It this class he includes Oracle Developer 2000, Inprise IntraBuilder, Sybase's Powerbuilder, WebObjects, and Allaire's Cold Fusion Application Wizards.
- **Multi-paradigm tools**. Applications that integrate elements of database management, site management, site modelling and website control. In this category he includes Microsoft FrontPage, Hyperwave Server 4.0 and Oracle's Web Development Suite.
- **Model-driven generators**. He explores many aspects of data-intensive website publishing through the AutoWeb system for website design and delivery. In fusing an entity-relationship modelling approach to a web site modelling language (HDM), he presents a top-town conceptual model for design of language and device independent navigational and structural components of the website (Fraternali et al, 2000).

The first five categories described by Fraternali characteristically situate the Web Server as both a content repository and a runtime engine. Such authoring tools are generally targeted to specific classes of content – for example Homesite for HTML development, Dreamweaver for HTML and Javascript development, Adobe Photoshop for image and animation development, and Macromedia Flash for development of Flash animations. The WebDAV protocol for

collaborative authoring is also oriented to such a dualist role for the runtime engine. More recently WebML has emerged to provide a synthesis of site modelling and database integration for data-intensive content delivery (Ceri, Fraternali, & Bongio, 2000). The WebML model offers a valuable perspective in approaches to a Computer-Aided Software Engineering direction in website design and there is considerable value to the WebML perspective on modelling as an approach for the consistent management of the runtime delivery paradigm. While WebML has attracted a reasonable theoretical community, its industry implementation is limited. Such modelling still retains a web-centric focus, although deriving the benefits of a presentation-neutral way of describing in a UML-like manner the logical structure and data interfaces of a dynamic data-driven website.

In analysing Television production systems, Heitmann (1999) classifies products in four categories: Integrated systems, Studio automation systems, Middleware libraries (class libraries, etc) and Search engines (Heitmann, 1999, p.26). Heitmann's "Integrated Systems" typify the "Content Management System", while the studio automation system centres on the management of the runtime delivery systems. Several products have made a transition from a collection of loosely federated components to an integrated solution – a recent example being the Macromedia "Content Management" solution.

2.6.3 Industry reviews and reports

This section presents a brief overview of existing treatment of Content Management Systems in technical literature and software. There are several industry reports comparing content management functionality and price. Until recently, most have been directed to assisting the purchasing decisions of organisations and aiding in the software selection process. Recently, however, the level of interest in CMS products has seen the emergence of a body of literature addressing the broader functional portrait of Content Management Systems in both theory and practice.

Boiko (2002) recently published the "Content Management Bible", an industry review directed to the selection and development of CMS products. In his selection criteria (presented in Chapter 16 of the book), he presents a detailed model for a Request for Purchase, focussing principally on issues of workflow, management and deployment. The functional dissection of the CMS represented by this presents a traditional view of the Content Management System – directing its strengths to the editorially managed publishing of content. Of 179 questions he presents, only 6 pertain to the nature of the content ontology that drives the representation of information. The task of selection when choosing a Content Management System has become a daunting exercise, with more than 225 substantial products in the market.

Byrne (2003b) presents in the "CMS Report" a detailed evaluation of 25 of these products, with a greater content focus on this evaluation. Both books situate XML centrally in the management of content, with Boiko treating content objects as classes and components (in an object-oriented view of the content) and Byrne focussing on the XML handling of the content.

The PCMag website (produced by PC Magazine) provides brief product overviews of CMS products, with a useful comparative pricing matrix (Clyman, 2002a). The pricing shown in the Table 2.2 below is based on the published information in this pricing matrix. Beyond the simple product cost, implementation consulting has been estimated at 4-5 times the actual product price (Clyman, 2002b; Dalton et al., 2001).

There are clear functional elements across this literature that are attributed to the CMS:

- Workflow management - the ability to create editorial and authoring processes that manage the full life cycle of content authoring, review and delivery, sheltering the user from the technical aspects of this process
- Version Control - the ability to review and recover content across different versions of this content.

- Content Separation - the separation of content from style and structure to allow its presentation in an appropriate form guided by template or in a style-sheet driven manner at whatever location is deemed appropriate for the content.
- Content Transformation - the ability of the Content Management System to transform the fragmented content into a published form (Addey et al., 2002; Boiko, 2002; Rockley, 2003).

The analysis presented below differs from industry surveys in its explicit focus on content reuse. A content analysis leads to finely granular functional decomposition of the CMS. This function dissection is explored further in Chapter Five.

2.6.4 Systems Reviewed

Rather than repeat the reviews already available from industry website, this survey evaluates a few representative products for the ways in which their systems map to the reuse model. CMS products were selected for the review into three broad categories:

- Category 1 - Larger CMS suppliers with integrated database management and versioning control, site modelling and multi-target content publishing, and generally with an implementation cost exceeding $100,000.
- Category 2 - Smaller CMS and site editor suppliers using the Web Server as the content repository supplemented by template-based approaches to web page creation and authoring enhancements. These products have an implementation cost that is characteristically under $100,000.
- Category 3 - Open source solutions that are often heavily influenced in functionality by their community of developers (for example, blogging features highly in some of the open source CMS products, but rarely in the commercial products).

Table 2.2 below shows the systems selected for review, representing a cross-section of large, small and open source CMS solutions. Systems were selected if they looked to have a representative set of functions that fulfilled at least a few of the content reuse principles described above. This functional analysis therefore explicitly excludes simple website editing tools such as

Macromedia Dreamweaver 4 and Microsoft FrontPage 2002. Systems were preferred that had good published technical information available. Information pertaining to these products was gathered through content analysis of their published literature and where relevant reference to industry commentators and discussion lists - and in a few instances after direct discussion with suppliers. The process was iterative, as functional elements were identified as part of the industry review.

	Category 1	**Category 2**	**Category 3**
Characteristics	-Proprietary >$100,000	-Proprietary < $100,000	Open source
Products reviewed	3rd Mill Documentum Interwoven Teamsite Vignette Obtree	Gravitymax Sytadel Strongpoint Microsoft CMS 2002 RedDot	AeGir/MidGard v1.4 Zope Lenya/Cocoon Redhat CCM

Table 2.2: Systems reviewed

In the process of creating the functional deconstruction of CMS products (and excluding those products directed principally at website editing only) the author has classified Content Management Systems in three broad categories.

Systems are categorised as "Page Centric" if their design privileges the individual web page for delivery to one or more servers, with the web server acting principally as a file delivery agent. Systems are typified as "Server Centric" where the system has its own web server or web site model and delivers content dynamically, often with plug-in functionality such as Bulletin Board Systems, Blogging and integrated FAQ facilities. Systems are described as "Content Centric" where they focus on the content rather than the target-publishing environment. All content units are retained within the CMS product for transformation and delivery to relevant target runtime environments. Some CMS suppliers refer to this as "object-oriented".

2.6.5 Category 1: Proprietary > $100,000

2.6.5.1 3RD MILL

- Software Base: Cold Fusion
- Content method: Web centric
- License: Commercial
- Website: http: //www.3rdmill.com.au/3rdgen

3rd Mill is a mid-level proprietary CMS developed by an Australian company. Being a relatively recent offering, it has a relatively strong content focus, with a template driven focus on publishing which appears to be similar to the template-based approach taken with the Inter-Publish prototype. It has a web-centric orientation and a simple role-oriented workflow framework. Figure 2.4 shows the user interface, with the left-hand navigation presenting the content in a manner reflecting the published website.

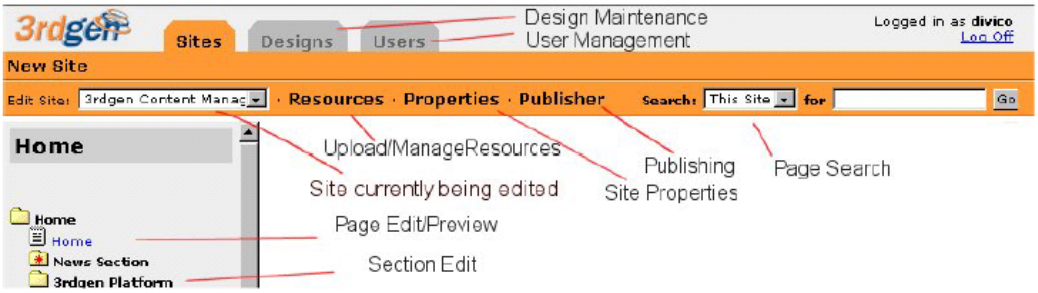


Figure 2.4: 3rdMlil home page – 3rdGen (2003a)

Figure 2.5 shows the logic of their workflow engine. Large multimedia resources are stored on the file system rather than within the CMS. The workflow engine is linear.

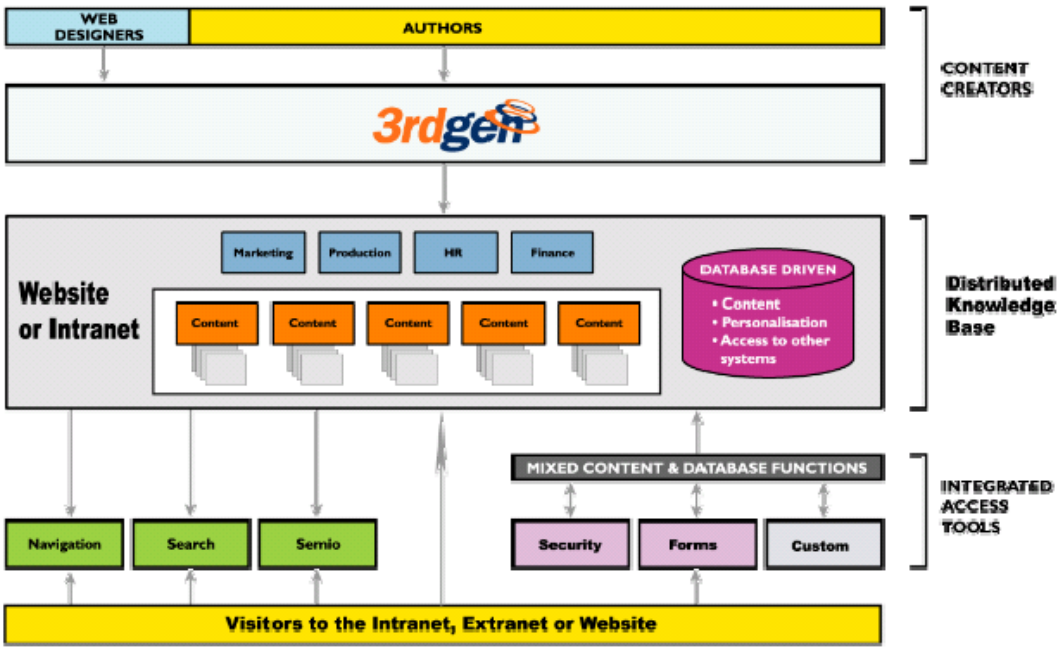


Figure 2.5: 3rdGen technical architecture – 3rdGen(2003b)

2.6.5.2 DOCUMENTUM

- Software: J2EE
- Content method: Page centric
- License: Commercial
- Website: http: //www.documentum.com/

Documentum provides a "framework" for implementation of an organisational CMS, bringing together many components that integrate with their Document Management solution.

The strength of Documentum appears to lie in the rigorous workflow management of the CMS, drawing on the substantial existing Document Management product set and customer base. Documentum have leveraged their existing customer base to achieve a strong presence in the CMS market despite the lead in the large corporate market that Vignette has established. Nevertheless, their first release of a CMS solution, essentially an extension of their Document Management solution, was soon sacrificed to a ground-up redevelopment that can support alternative web servers, and marking a transition from a Server-Centric approach to a Page Centric approach.

The most recent release of Documentum purports to be a ground-up redevelopment, moving to an environment that will operate with alternative web servers. The vendor provides optional product components, including syndication options, Web Services support and a wide range of "add-on" components, which together comprise a comprehensive but expensive CMS solution. Given its corporate cost profile, even for the basic product, Documentum remains a product for the large corporate arena.

2.6.5.3 INTERWOVEN/TEAMSITE

- Software: Perl
- Content method: Page centric
- License: Commercial
- Website: http: //www.interwoven.com

Interwoven, like many of the major CMS suppliers, provides a complex toolset of integrated products. The workflow and other aspects of Interwoven may require quite low-level scripting. This element of the software has been a point of criticism in discussion groups as it has implied some difficulty in transition between versional releases of the software platform.

One of the major strengths of this product appears to be its virtual content file system (the Teamsite DAM) that gives a desktop-oriented content view that is in fact personalised to each user's data view.

2.6.5.4 VIGNETTE

- Software: J2EE
- Content method: Content centric
- License: Commercial
- Website: http://www.vignette.com

Vignette makes very little technical documentation available online, and the assessment of this product relies principally on published assessments. Vignette is well situated in the Finance and Commerce markets, with strong integration capabilities with existing DBMS systems. Byrne (2003b) is not complimentary of its authoring interface, and its component-based approach can lead to complex infrastructure requirements for publishing. As one of the more mature commercial offerings it has particular strengths in the full life-cycle management of content, and particularly the tracking of usage specific to particular events.

Like some open source solutions, it has a toolkit orientation, with considerable customisation characteristic of a particular installation. It also supports (often as add-ons) a broad range of delivery vehicles. Due to lack of detailed technical literature on the product, Vignette is not analysed in detail in the functional dissection. It remains, however, one of the market leaders in the large end of town.

2.6.6 Category 2: Proprietary < $100,000

2.6.6.1 GRAVITYMAX

- Software: ASP & .NET
- Content method: Page centric
- License: Commercial
- Website: http: //www.gravitymax.com.au/
- Server limitations: Windows Server platforms

GravityMax is an Australian commercial CMS that offers a good functional breadth in web publishing, although its syndication capability appears to be limited. Its workflow implementation appears to be solid, with role-oriented allocation of tasks and a linear workflow model.

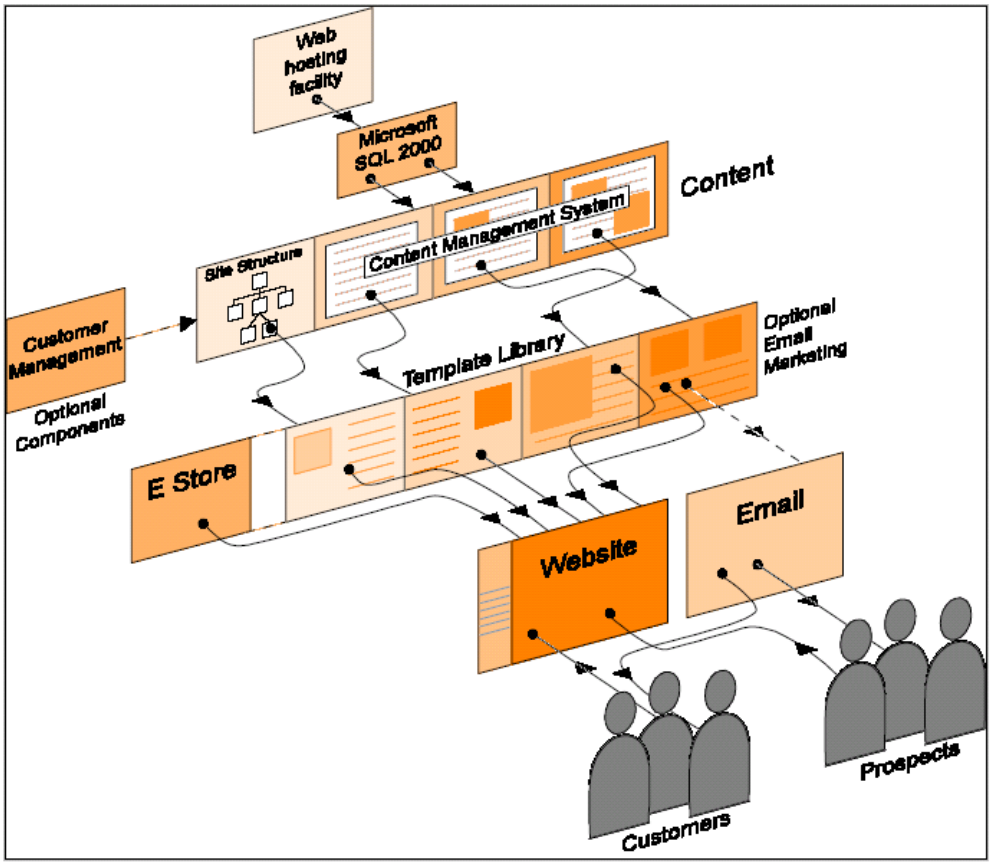


Figure 2.6. Gravitymax technical architecture - Gravitymax (2003)

The product now has a tight integration with the Microsoft Web server framework, with IIS its main web server base. The depth of support for complex syndication and indexing was not clear in the GravityMax product documentation, however the product appears to have good content separation (see Figure 2.6). The design of the CMS is essentially Page Centric with some extensions for Email marketing.

2.6.6.2 SYTADEL

- Software: PHP
- Content method: Content centric
- License: Commercial
- Website: http: //www.synop.com
- Server limitations: Linux (RedHat or Debian)

Sytadel is a PHP-based "community-based" product directed to a member-based or blogging community of publishing. The product appears to have leveraged the PHP platform effectively to allow the integration of functions into the software.

While it has a strong content focus, Sytadel has focussed on the Intranet market. It has a simple serialised workflow process. Sytadel is rare among its cousins in that its underlying content base is XML with all content transformation being dynamically produced using XSL. It does not support XHTML, and does not appear to support the major syndication technologies (PRISM, RSS, NEWSML), although the underlying XML approach for text content would probably facilitate such syndication. Sytadel lacks Unicode and double-byte support, a limiting factor in internationalisation of the product. The product, being young to the market, nevertheless brings a sophisticated approach to content handling, with an ECU-style management of content units, although limited in its depth of functions compared to larger commercial products.

2.6.6.3 OBTREE

- Software: J2EE
- Content method: Web centric
- License: Commercial
- Website: http: //www.obtree.com

ObTree provides a "full lifecycle" framework for content publishing, with flexible granularity in the content definition, although with a web-centric framework for content delivery, as illustrated by the technical production diagram shown in Figure 2.7 below.

Being ground-up developed as a CMS (as distinct from the Document Management roots of Documentum and others) the product appears to have a tight focus on a workflow oriented to web-based publishing, with the traditional use of staging servers in the workflow production process. Obtree includes WYSIWYG authoring capabilities presented in a highly structured user management form.

The Obtree claim to be "object-oriented" centres on the fact that the product stores all content objects within the application and manages the content inter-linkages with no external website or file system references (Williamson, 2001).

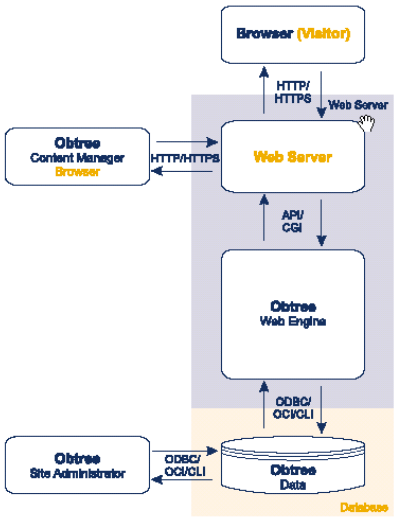


Figure 2.7: ObTree "object-oriented" engine - Obtree Technologies Inc, (2003)

2.6.6.4 STRONGPOINT

- Software: .Net
- Content method: Page centric
- License: Commercial
- Website: http: //www.strongpoint.com.au
- Server limitations: Windows Server platforms

Strongpoint is another Australian CMS serving the mid-to-low price range. Strongpoint generates all documents dynamically. The integrated search capabilities of the product appear at present to be limited. The software appears to have a flexible, though linear, workflow paradigm.

2.6.6.5 MICROSOFT CMS 2002

- Software: .Net
- Content method: Web centric
- License: Commercial
- Website: http: //www.microsoft.com/cmserver/
- Server limitations: Windows Server platforms

Microsoft purchased the *Ncompass Resolution* CMS. They have extended this largely web-centric CMS product through integration with the Microsoft DotNet framework, with a component orientation to the system design (see Figure 2.8 below) that makes it amenable to integration with other systems. Not surprisingly, the product has strong integration with the Microsoft desktop application toolset. Its workflow is largely linear in nature, and syndication is principally supported through the Web Services framework - which in itself might offer an effective framework for mapping to other syndication methods.

The Microsoft CMS solution appears principally oriented to the efficient delivery of content to their CMS web server, with some extensibility offered through XML syndication and Web Services. Web Services appear to be strategic to the plans for integration of this product with other systems.

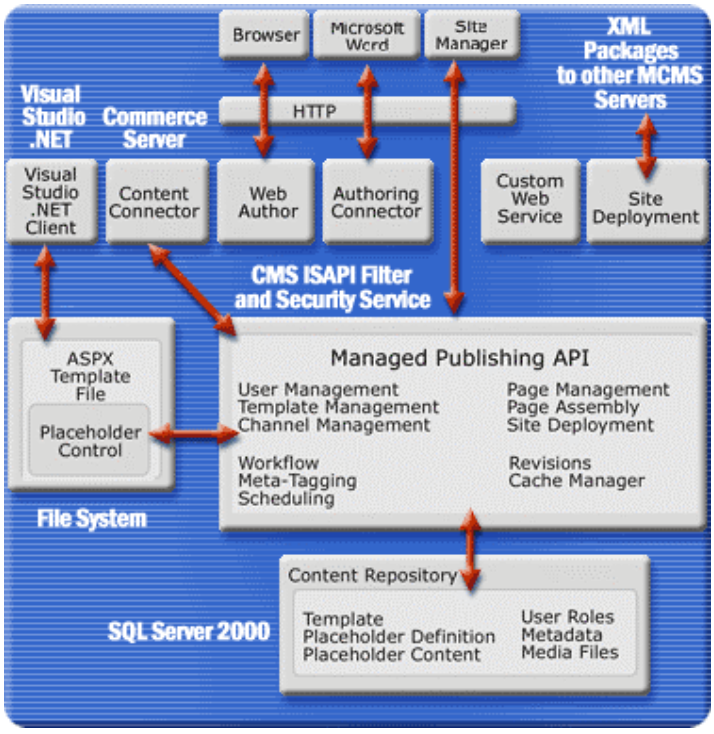


Figure 2.8: Microsoft CMS 2002 technical architecture - Microsoft (2003)

2.6.6.6 REDDOT

- Software: J2EE
- Content method: Content centric
- License: Commercial
- Website: http: //www.reddot.co.uk
- Server limitations: Windows Server platforms

RedDot ECMS provides a component-based framework for content publishing and personalisation. While RedDot supports syndication, this is achieved through an optional syndication engine. The product appears to have a strong underlying XML support, with an XML content import engine which has a user-moderated "briefing" process in order to process the XML data into the RedDot equivalent of the ECU. The workflow capability also appears to be a strong element of the software with language variants also supported.

RedDot, like Sytadel, generates content dynamically in its own server environment and does not support static generation to other content servers, although it can integrate with other server-based runtime delivery engines (see Figure 2.9 below). While the product supports a range of web servers, RedDot uses, like Vignette, a component interface to be back-end CMS to "fetch" content.

RedDot appears to have a significant presence among the US universities. The CMS includes not only internationalisation support but also a translation editor to facilitate multilingual content editing. RedDot, like Vignette, has several years more maturity to market. It has a strong workflow design with some workflow visualisation capability. Publishing is achieved through a template-based approach with their own markup approach.

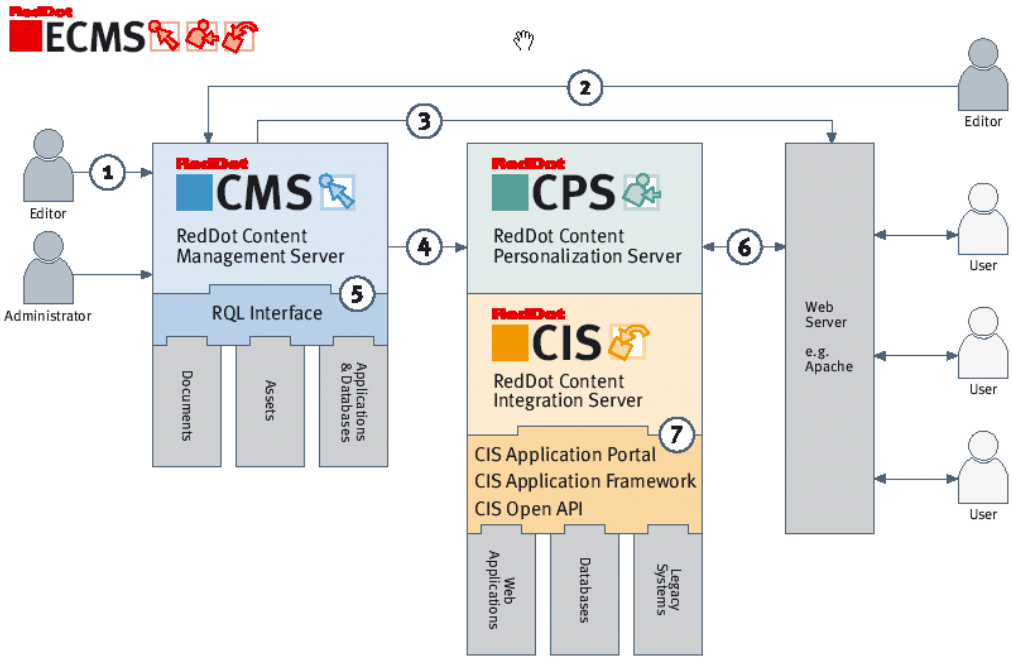


Figure 2.9: RedDot technical architecture - RedDot (2003)

2.6.7 Category 3: Open Source

2.6.7.1 AEGIR/MIDGARD V1.4

- Software: PHP
- Content method: Server centric
- License: Open source - GNU Library General Public License
- Website: http: //www.aegir-cms.org/

Midgard is a PHP-scripting-based open source solution. It has integrated WYSIWYG and basic workflow management. It is server centric, coupled with Apache web server to provide a web-based view of the content operating in an application-server framework. Deployment of the product is relatively straightforward and the PHP model makes it amenable to local customisation, with an extensive Application Programming Interface (API).

Midgard uses MYSQL as its database backbone. MySQL is probably the most popular open-source database, although probably better oriented to content serving rather than a high frequency of content updates, with a relational engine better tuned for retrieval than high-volume updates. The MySQL link brings with it limitations in Unicode support, perhaps limiting its support for internationalisation, particularly in countries requiring double-byte / Unicode support. It is designed as a dynamic site generator, and is template oriented and server-centric in its design. There also appears to be limited version control – versioning and more sophisticated workflow control would probably be needed for large organization support.

Aegir extends the Midgard solution, with a focus on the community of news organisations. It has a simple, serial, workflow system of editorial approval of articles. This workflow cannot be modelled or significantly modified. Tagging of content through the style-sheets requires some PHP scripting, and the system as a whole can be regarded as a framework rather than a solution. This open source project is most popular in Europe.

2.6.7.2 ZOPE (Z Object Publishing Environment)

- Software: Python
- Content method: Web centric
- License: Open source - BSD
- Website: http: //www.zope.org/

Zope is a Python based open source application server framework. It has an integrated web server and is oriented to dynamic page generation. Unlike Midgard, Zope supports a complex object database and has an international spread of popularity. Zope and RedHat are probably among the most "polished" of the open source solutions. Zope includes relatively mature integrated help and has well internationalised core. Like many CMS products it uses its own tagging system called DTML, as well as a scripting language "Zope Server Pages".

Zope is open to a number of authoring access modes, with FTP and WebDAV support. Zope, like Midgard, is oriented to a web-centric dynamic page generation model, with its own "zserver". This server also integrates a search engine "zcatalog".

As one of the more mature open source solutions, Zope also supports LDAP authentication and has a reasonably configurable role-oriented workflow model. The PHP/Zope platform has spawned a number of CMS variants, both open source and commercial, based on this underlying model, including *Nuxeo* (www.cps-project.org).

Nuxeo CPS (Collaborative Portal Server) (open source) extends the collaboration/workflow aspects of Zope. A commercial release - *EasyPublisher* - extends the capability of Zope in workflow management and simplified user interface.

2.6.7.3 LENYA / Cocoon

- Software: Java on the Apache server platform
- Content method: Web centric
- License: Open source - GNU Library General Public License
- Website: http: //cocoon.apache.org/lenya/

Apache Lenya (formerly Wyona CMS) is an Open-Source Content Management and Publishing System written in Java. It is based on open standards such as XML and XSLT. Lenya extends the Cocoon framework from Apache and includes XML WYSIWYG editing capabilities and a task-oriented linear workflow capability.

Like Midgard, Cocoon is server centric, coupled with Apache web server to provide a web-based view of the content as an application server-style of operation. The Lenya roadmap has a strong element of community publishing. Like Zope, the Cocoon framework has spawned a number of CMS variants similar to Lenya. Figure 2.10 illustrates the roadmap for Cocoon, including its direction toward web services integration.

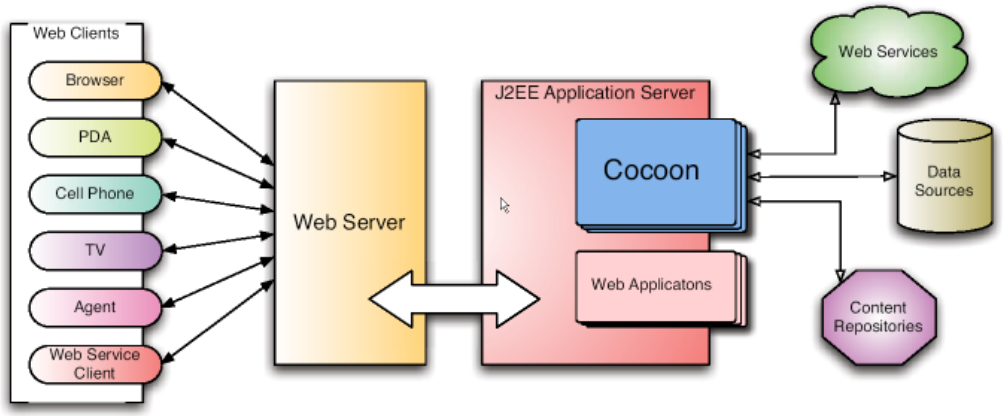


Figure 2.10: Cocoon roadmap

The "component based" approach to Cocoon and its sophisticated Application Programming Interface, along with the extensive experimental work associated with XML in the Apache project make this product one of the most functionally rich Open Source solutions. The loose federation of components comes at a higher implementation cost – while many functions are available, extensive scripting may be required to realise these components as a finished CMS.

2.6.7.5 REDHAT CCM

- Software: J2EE
- Content method: Page centric
- License: Open source, managed by RedHat
- Website: http: //www.redhat.com/software/ccm/

The RedHat CCM is a "thread" from the Ars Digita CMS product, released as part of its Application framework in Open Source. The RedHat CCM system gives a representative range of functionality for the CMS, although with a somewhat rigid document ontology (instantiated as database objects). An additional "thread" from the original Ars Digita CMS is the UK Local Govt originated CMS "APLaws", with the latest version built on RedHat CMS 5.2. (http: //www.aplaws.org.uk/plus/pages/download/software_cms.cfm)

RedHat has committed to integration with its other Enterprise products, creating in the process probably the best open source solution available. While not as innovative as Tiki, the credibility of the support surrounding the RedHat products gives it greater presence in the market. The recent commercial release of the RedHat server may indicate a trend to commercial rather than open source delivery at RedHat.

2.6.7.6 Other Open Source products

Several other Open Source products have a strong presence in community-based publishing, and in particular OpenACS and Tiki. Tiki is a relatively new PHP-based CMS open source project (starting in 2002) but based on an older source code core called Wiki. Tiki has an active community, resulting in a large number of features implemented in a relatively short timeframe: such as newsletter, blogging, file/image gallery, poll/survey, quiz, FAQ, chat, banner, webmail, calender-based event management and other useful community-based website functions.

The PHP base may be a factor in the rapid community contribution to the functional growth of this product, which appears to have a large and active Open Source community. Its community

focus has seen it popular in more recent content publishing forums such as blogging. The integration of several plug-in style modules also facilitates its utility. Tiki is often compared with a similar open source product, "OpenCMS". It provides a basic set of functions for a web-centric CMS operation with broad template-driven content authoring but limited syndication. OpenCMS does not appear to have been as "nimble" in community-driven growth of functionality as its younger rival Tiki.

OpenACS (Open Architecture Community System) is another "thread" from the Ars Digita original, and continues the toolkit-based approach, with significant customisation for implementation.

2.6.8 Industry Practice and Content Reuse

The last three years have seen the emergence of a second generation of proprietary and open source approaches to Content Management. Both communities demonstrate considerable innovation and experimentation. The systems reviewed above indicate the functional richness of the Content Management System products currently reaching the market: systems that clearly demonstrate their functional capability in original publishing of web-based resources.

Some CMS products take a strict XML approach to content, storing other multimedia content types in the operating system. Others take a database approach, expressing each content form as a document "class" or table schema. An emergent theme, which this model exemplifies, is the disaggregation of content at an economically effective level - a process variously called "chunking" (Byrne, 2003a), fragments (Boll & Klas, 2001) and components (Boiko, 2002).

CMS solutions may integrate search features or rely on search-engine functionality of run-time engines. However, important to the long-term effectiveness of large content stores is the role of the search engine and the degree to which it supports complex query generation across the content collection.

As has been discussed, XSL provides many features that facilitate the generative presentation of content in its final form (for example HTML). However, there remains a need to scan databases for content against complex selection criteria. These selection criteria may be fixed, may be the result of persistent user search specifications, or may include features for heuristic content clustering. Intelligent agents take such search features a stage further in maintaining a self-tuning and persistent search agent that is constantly scanning for content based on complex rules.

As Internet publishing escapes its English-language origins, multilingual issues are emerging as an important issue, as web content management meets the multilingual challenge. The CMS can therefore serve to manage reuse in the context of both language variants and format variances in the published content, organising multi-lingual versions of the same content in a way that facilitates the co-location of semantically similar multilingual fragments. This can require the presentation of multilingual content elements "in parallel" for effective comparative review.

All products demonstrated some methodology for formal separation of content from structure and presentation. Sytadel and Obtree exemplified a rigorous template-based approach with underlying XML database. Zope and Cocoon exemplified scripting-based solutions to the content management problem – providing a framework into which additional functions and functional examples could be built. PHP itself has been a popular language framework for the engineering of a variety of Open Source systems.

There are varying approaches to content fragmentation. Only a few (and particularly Sytadel) preferred an underlying XML database or document structure – with most still opting for a relational database for content storage. Most continue to manage video or other complex multimedia assets on the file system or web server rather than within their database framework.

Appendix One has a detailed functional decomposition of the fourteen products discussed above. This Appendix also has a document map of technical literature used in reviewing these

products. The functional matrix is re-interpreted in Chapter Five against the model for content reuse proposed in Chapter Four and plays a part in the evaluation presented in Chapter Seven.

The complexity of the functional list emerging for current CMS products hides the immaturity of the products in some areas:

> It is important to recognise – and many analysts have already pointed out – that WCM (Web Content Management) is a fairly immature space. Antecedent products that addressed Document or Asset Management may have bequeathed some key concepts like Workflow and Library Services, but the WCM problem domain remains fluid enough that true product categories are only now emerging. In nearly every case the products themselves are only a few years old, and even then, may have been completely rewritten from scratch. (Byrne, 2003b, p.61)

The analysis of the Content Management System in theory and in industry practice indicates that there is a recognisable set of intrinsic functions recognisable as the "Content Management System". These principles are also axiomatic for good practice in content reuse and are summarised in Principle 11 below.

> *Principle #11- The issue of content reuse goes beyond specification for a document exchange ontology and requires a systems approach that is currently characterised by the Content Management System. Core functions for content management to support content reuse are:*
>
> - *The modelling of the document ontology in a manner that separates the informational element from the presentation and structure.*
> - *The management of additional semantic metadata with the documents for content discovery.*
> - *The management of the workflow processes whereby content is authored and updated in a manner that meets the editorial requirements of the organisation (including the use of staging servers to review content in its proposed context).*
> - *The version management of content at the micro (content unit) and macro (web site) levels.*
> - *The management of rules pertaining to content transformation, distribution and personalisation.*
> - *The processes associated with archival management of the content after its currency expires to allow content discovery and reuse in other contexts.*

2.7 The *Economics* of Content Regeneration and Reuse

There is an inherent tension between the efficient authoring of content for the once-off regeneration of information and the markup of content for long-term archival or research purposes. Rigorous structural and semantic markup can be an expensive and labour intensive process with a process focus that may detract from the creative elements of content composition. While in many cases authoring systems go some way toward facilitating this semantic and structural identification, the situation of content in a "Content Management System" often entails several stages of content enhancement both of structural and metadata elements.

There is a also a contingent factor in the markup of content that centres on a cost-benefit between the demands placed on the content author and the likely later uses of the content in other contexts. There are several approaches to the elemental analysis of content. The content capture system can guide the user to content identification in a manner that explicitly recognises some content structures. Automatic semantic analysis can detect both obvious structures (sentences, paragraphs, etc), and more complex components (links, images, etc). Finally, markup may be added manually to existing content through standards such as TEI. While XML editors have begun to appear on the market, content capture using such tools needs further improvement.

The utility of the Content Management System is founded on a notion that content has value in elemental as well as complete form: that meaningful identification of components of content can yield economic benefits to the publisher in reintegration of disparate content components in new forms and the regeneration of content components in new versions or reassembled in new media in what (Heitmann, 1999) refers to as the *essence*. The Content Management System serves a key integrative role between the content author, publisher, researcher and archivist into an information systems context. The fundamental transition this represents is from traditional "fixed" versions of publications to the efficient means of generating content for different runtime engines for effective dissemination and version control.

The experiments with rigorously marked up literary texts arising from the Text Encoding Initiative have yielded valuable insights into the nature of text itself. There was considerable initial enthusiasm that SGML elicited a fundamental hierarchy implicit in all text. Websites themselves follow an implicit hierarchy that often reflects the file system structure of web servers. Hypermedia linking, however, introduces networked models of the content structure that once again introduces concepts of efficient economic units of information for purposes of inter-linking of content. DeRose et al (1990) and Coombs et al (1997) discussed text documents in terms of 'hierarchies of content objects", and Renear restates this argument as late as 1999 (Renear, Hockey, & McGann, 1999), while moving away from the description of the hierarchical model as a complete explanatory device for text. Hierarchical models of content provide a valuable element to management and exploration of content, and often characterise the content structure of websites. While hyperlinking creates inter-relationships between textual components on web pages, web sites are generally overlayed with basic navigational structures that categorise and organise the web page objects. Similar content forms are expressed in Chapters, Sections and structural devices for printed organisation. The difficulties with hierarchical markup structures lie in the complex semantic relationships that occur in complex texts. Terse forms of communication, such as poetry, may contain overlapping semantic structures that cannot be adequately described by a single hierarchical structure. Nevertheless, the utility value of imposing a hierarchical overlay on the text remains strongly defended, in a modified form, by Renear, Mylonas and others. While couched now in the contingent situation of perspectives, they still assert that "non-hierarchical perspectives can often be decomposed into hierarchical sub-perspectives." (Renear, Durand, & Mylonas, 1996). The hierarchical sub-perspectives of text achieve an additional dimension when multimedia content is introduced. Similar organisational hierarchies are evident (Boll & Klas, 2001), although once again it is not credible to describe these hierarchical relationships as essential to the structure of multimedia documents.

However, the disparate encoding formats for different types of multimedia by this very virtue require some content segmentation along the lines of content types.

The utility value of content reuse extends this concept to the segmentation of particular content items into what Boll & Klas (2001) refer to as *fragments* or what this thesis refers to as *economic content units*. A more systematic approach to content reuse can serve to:

- Address issues of inconsistency in the presentation of the same semantic information across different media.
- Ameliorate the economic costs of content recapture during generational changes of content.
- Reduce the opportunity cost represented by the failure to deploy content in all relevant content forms (for example through syndication).

The term *economic content unit* has been adopted in this thesis to describe that particular granularity of content organisation, which yields optimal results for content reuse.

> *Principle #12 - Fragmentation of content is essential to achieve content reuse, although the granularity of this fragmentation is contingent on the reuse objectives. The results of this fragmentation are "content units". The choice of a level of fragmentation has resource considerations for the ongoing management of the content. A finely granular fragmentation of content necessarily adds complexity to the document description of content and the ongoing management of content relationships. A "chunky" or coarse treatment of content at the whole document level simplifies the document ontology but limits the extent of reuse. Fragmentation to the word or phrase level may not, in many contexts, be semantically meaningful. The balance between the utility finely granular fragmentation and coarse fragmentation leads to the concept of the "economic content unit", which is the optimal level of content fragmentation for the ongoing management of content in a particular context.*

2.8 The *Management* of Content Regeneration and Reuse

> *We still do not know nearly enough about the properties of digital information, particularly its permanence, its integrity and its persistence. (Ekman, 2000, p.10)*

Digital content remains a new innovation in the context of the many thousand years we have been managing document archives. The original management of digital documents presents enormous challenges to the library and the archivist despite the fact that organisations that have traditionally managed print collections (such as libraries) are moving rapidly to digital only document collections (Balnaves, 2003a).

CMS products have borrowed key conceptual elements of Workflow Control from the Document Management systems. They combine this with elements of authoring, publishing and distribution (Boiko, 2002; Hackos, 2001). Effective editorial control is a hallmark of the better Content Management Systems, but the content issues extend well beyond this. Effective website management can require the partial or complete "staging" or versioning of the website across multiple media forms so that new generational versions of the site can be viewed fully in context.

Content reuse adds further complexities to the management challenges of dealing with digital content. The effective management of content regeneration and reuse has the extended complexity of the management issues relating to the distribution, often to multiple different forms of media, of the original content sources. This entails also the possible need to track the usage of content through its distribution channels. Both syndication and email distribution may require management of content resources to ensure content delivery is not duplicated.

Text, image, audio and video content each have very different challenges in the management of large content databases. Digital content in the Internet arena has grown to some degree independently of the intellectual rigour surrounding classification systems. Metadata classification of content needs also to address the classification of content at the elemental level, not just of content in its final form, and in a manner that enhances the distributed content identification and dissemination of the content. The Text Encoding Initiative (TEI) has

demonstrated that the advantages of standardised markup in Digital Libraries go well beyond the realm of information exchange and offer new dimensions of textual analysis and criticism.

Problems with Information Retrieval in the context of multimedia are highlighted by Wood (1998), and there have since been many experiments in the automatic discovery of semantic information in audio and video. The success of one-shot queries in web sites and similar systems depends greatly on the specificity and relevance of metadata associated with the content. The structured collection of metadata itself can be a time-consuming task, and is itself very subjective. Santini (1998) explores the use of User Interface design to maximise the contextual information implicit in the content database to enhance the process of retrieval. Central to the role of a Content Management System is therefore the flexible capture of the content itself, and the effective meta-description of the content at the elemental level.

As with document management systems, there are clear life cycles in the management of web-based content through the authoring and publication process. There are also functional elements of this life cycle that are distinctive to the CMS, including:

- Content separation from structure and style (as distinct from Document Management which may give preference to the preservation of the original document content, structure and style). The separation of content from style and structure is essential to the later presentation in an appropriate form guided by template or style-sheet to whatever location is deemed appropriate for the content.
- Workflow management - the ability to create editorial and authoring processes that manage the full life cycle of content authoring, review and delivery, sheltering the user from the technical aspects of this process.
- Support for collaborative authoring, accommodating unstructured documents and simplifying the technical process of publishing (Bergeron & Bailin, 1997; Dalton et al., 2001).
- Version control and the ability to manage review and recover content across different versions of the content.

- Content Transformation - the ability of the Content Management System to transform the fragmented content into a published form. . (Addey et al., 2002; Boiko, 2002; Rockley, 2003)
- Content Visualisation. A key weakness in current systems at the micro (content) and macro (website) area is that of effective content presentation to facilitate design and management of content resources and their relationships (Ceri et al., 2000; Fraternali & Paolo, 2000) (Colbert et al., 1997) - and the integration with content authoring tools (Thomas, 2000).
- Improved content identification. This especially pertains to the effective semantic markup of content to assist discovery of the content. Effective content identification also facilitates the systematic reuse capabilities supporting the effective regeneration of content to runtime engines in revised presentational forms or in different presentational media (Boll & Klas, 2001; Dalton et al., 2001).
- The integration of scripting and database systems for management of the content with systems for the content delivery (Fraternali, 1999; Thomas, 2000).

2.8.1 Content Modelling

Fraternali (1999) has emerged as a key proponent of a formal development approach to website design and implementation. Subsequent research into the effective integration of data-intensive applications with websites has seen the development of a coherent web modelling and visualisation approach to Content Management.

Ceri and Fraternali (1999) propose WebML as a modelling methodology for the management of complex data-intensive websites. They define an XML methodology for generalising the structures of the website and the interfaces to data sources, in the context potentially of the use of Case-like tools in a formal development environment for website design. They stress composition primitives not unlike the content units discussed above but with an integral treatment of compositional and presentational aspects of the content management. WebML is server-centric in the sense that it does not recognise multiple presentation or

distribution channels for the content, and has to date been principally directed to the delivery of text-based informational content.

Fraternali & Paolo (2000) explore the visualisation and modelling of websites at the "macro" level. A content management model, to achieve the goal of both content reuse and content preservation, will necessarily provide visual presentation of the structure of information represented in a site.

It was clear from the survey of current industry practice that few CMS products support WebML or feature it in their development priority. There are clear indications, however, that they are experimenting with modelling concepts in website design that may well benefit from a formal, coherent, Website development.

2.8.2 Content Identification, Persistence and Discovery

The URL has considerable value in the short term, both as a brand and an identifier. It is not, however, perfect as a permanent locator of content resources. Dahl & Hedberg (1996) express the frustration of the Library community in the evolution of this identification standard:

> It is very typical of the situation within the ISO work that the main actor at this meeting was a representative of the Internet organization, the Internet Engineering Task Force, which had established what they called the URI Working Group. What was presented was more or less a fully developed technical standard, distributed for comment on the Internet and on its way to become a de facto standard without having the status of ISO behind it. It is a fast process, and related to a definite need, but the chances for a broader library world to influence the process is very restricted. (Dahl & Hedberg, 1996, p.128)

Of particular relevance is the experience of the library community in the use of disciplined cataloguing and Controlled Vocabularies in order to consistently describe organisations and subjects. They are not reflected in the standards for web-based resource identification, which have grown in a disjointed manner around the various naming authorities that administer the allocation of key domain identifiers.

Citations in printed works are themselves links. The identification, access and archival characteristics of a traditional journal citation and a URL citation are very different. While there is no certainty that a citation is discoverable, the likelihood of obtaining a well-cited work through a citation is much greater than the discovery of a URL citation. Table 2.3 below illustrates the contrast between Journal and URL citation.

JOURNAL CITATION	URL CITATION
Content is static	Content is variable
Content can be sourced uniquely through international delivery mechanisms	Content may only be available through one source (the website)
Content is widely distributed (multiple repositories)	Content is widely available but only through one medium.

Table 2.3: Print citations vs. URL citations

The transience of URL citations is a metaphor for the transience of published information on the Internet as a whole. In the long term the persistence of resources associated with content thus presents a dilemma. The integrity of a website may rely not only on the content within the local resource by inter-linkage with other resources. There have been experiments in maintaining versional histories of websites (Eaves, 1997) and in archiving new media sites, such as the Pandora project and the National Library of Australia (Phillips, 1998), which if nothing else demonstrate the cost and effort in long-term versioning of content. For instance, the Pandora project represents little more than 1,200 Australian sites, despite the cultural significance in the long term of preserving more of this material (Smith, 1998). It could be that Internet content delivery can be regarded as essentially ephemeral. The final form of a book or a journal had the virtue of a static nature: their content is the same for all readers. Personalisation, on the other had, dispenses with any degree of finality of information delivery: the content delivery may be different for each individual. Without a fixed point of reference in which content can be thought to have reached a "final" form – it is *essentially* dynamic and therefore ephemeral. Versioning of the entire website architecture when some elements are data driven and dynamic becomes an issue spanning multiple information systems. However, organisations do have an interest in

maintaining the integrity at least of their current resources, and the demonstrable transience of links to external resources creates an ongoing requirement to manage the creditability of current resources, and an issue of versioning both at the micro (content unit) and macro (website) level.

Content persistence over time is not helped by the speciation of software authoring tools. The widespread availability of multimedia content authoring systems allows authors to draw on unprecedented resources in the process of their analytical and creative endeavour. Yet it could be argued that never has so much information been managed so poorly by so many. The rapidity of the evolution of new means of content creation has resulted in considerable software and hardware change. Minor incompatibilities between versional releases of software compound themselves over time. It is sobering to reflect on the most common word processing formats just a decade ago: the ubiquitous Word Perfect, Word for DOS and others. Just fifteen years earlier sees the supremacy of the Wang Word Processing Systems and the central typing pool. Radical changes in hardware architectures, operating system platforms, and content storage formats are evident, with the effective speciation of some content authoring formats. A number of reports have highlighted the difficulties such speciation presents over medium to long time spans (Crespo & Garcia-Molina, 1998; M. Jones & Webb, 1998; Richards, 1997).

The move to dynamic and personalised content delivery places even greater emphasis on the archival management of the original content stores and their archival capability. There is a strong trend in the discussion of scholarly archiving of digital published material to place the archiving role back on the original publisher (Ekman, 2000).

Library Science has a respectable tradition in the engineering of systems and resources for the management, and disciplined approaches to cataloguing and classification of information. Cataloguing has one fundamental purpose: to fully and accurately describe an item in a way that enables it subsequently to be found at a proximate or precise location. The Library Catalogue describes the author, title, publisher, classification, shelf location, and sometimes the table of

contents, physical book information, book identifiers and other attributes. Strict disciplines of description (for example the Anglo-American Cataloguing Rules) allow recognition of catalogue items in a consistent manner: rules for handling multiple authors, lack of authors, consistent description of authors across multiple variants, etc. Central to cataloguing practice are three fundamental axioms:

- The unique identification of a document "shelf" location
- A consistent document-description ontology
- A consistent meta-description ontology.

Cataloguing therefore aims to give us a means of both describing and locating an item. Classification is an intellectual exercise of ascribing meta-information relating to an item in a structured and consistent manner. The emergence of online catalogues saw the development of sophisticated and flexible digital markup strategies for content identification, the most important of which is MARC (MAchine Readable Cataloging). Still the basic vehicle for document information interchange among libraries, the MARC format has a solid three-decade history of utility in flexible description of traditional documents.

Classification schemes evolved as a response to the escalating difficulty of managing and organising large collections. Online library catalogues and online journal databases represented the first meta-encoding of large document collections. Early online indexing research and trials by Lancaster (Lancaster, 1968) led to the first large bibliographic databases such as MEDLARS in the early 1970's. Such online databases began with the use of formal classification systems. However, the opportunities of keyword-based retrieval and content-based analysis saw the use of KWIC (Key Word in Context), KWOC (Key Word Out of Context) retrieval paradigms and complex experiments in semantic clustering of retrieval results based on relevance. More recently Z39.50 has evolved as a standard for distributed information retrieval across large and internationally dispersed bibliographic metadata collections.

In contrast, Internet-based search engine development has rarely moved beyond KWIC-style retrieval with ranking rules. Until recently the development of such Indexing Engines has tended to remain isolated from the discipline of Information Retrieval in Library Science. The economic benefits of semantic content identification have been more fruitfully explored in the discipline of Knowledge Management. Recent discussion framed around the "Semantic Web" heralds a move to the construction of semantically effective online Information Retrieval engines and search agents (Staab & Maedche, 2001).

Library Science has also contributed toward the evolution of complex searches across multiple database resources, particularly in the development of the Z39.50 protocol (Needleman, 2000). While little used outside the Library community, and essentially a means for resource location in a federated, heterogeneous set of library catalogues, it exemplifies the features required for effective content discovery:

- A standards-based interface for query-based interrogation of collections
- A complex query language
- The ability to discover and interrogate multiple collections
- The ability to retain persistent search collections.

The evolution of distributed, international, search capabilities across Library Catalogues represents a success in information integration at a high level of semantic consistency that is unrivalled. In the Internet space, organisations such as *Looksmart* have carved a niche in filling the gap in consistent meta-identification, but generally the focus on the web-environment and in knowledge management has been in encoding methodologies and automated structural discovery, with very little attention to the intellectual disciplines at the international level of consistent semantic description.

The transition of content from print publication to a hypermedia context has also seen the evolution of a new content identification dilemma: the transience of linkages with external resources. The Internet Uniform Resource Locator (the URL) is an apt metaphor for the

difficulties of unique content identification in the digital era. Digital content design and publishing systems lend themselves to a rapid rate of content design and delivery. The corollary to this is the rapidity of the entropy effect on this content. Lawrence et al. (2001) examined 270,977 computer science journals, conference papers and technical reports, extracting 67,577 URL references. They demonstrated both a dramatic increase in the use of URL's in citations and the substantial increase in broken links over time, peaking at 53% after 6 years. This again highlights the ephemeral nature of URL referencing, even within refereed journal publications:

> First, personal homepages tend to disappear when researchers move. Second, many who restructure Web sites fail to maintain old links. These problems are likely to persist without improved citation practices. (Lawrence et al., 2001, p.28)

Library Science has a term for loose-leaf items and irregular publications: ephemera. Ephemera often suffer from the difficulty of unique identification and the problem of description for cataloguing purposes.

In the long term, addressing the issue of obsolescence requires continuous transformation of multimedia content to ensure it is available through current delivery vehicles. Document ontologies alone will not suffice to deal with the problem of obsolescence. Content Management can play a significant role in long-term preservation – in providing precisely the architecture for continuous content transformation in an economically effective environment, as can the integration of effective digital document management as part of the contractual delivery process (Balnaves, 2003a).

2.8.3 Workflow Control of the Publishing Process

Newman (2001) situates collaborative management of information resources in the context of information as a "sign vehicle" (emphasising the semiotic element of knowledge management):

> Whereas early Information Systems placed emphasis upon the provision of information to management, there is now increasing emphasis upon the use of IT to create a media space or virtual environment to support task-oriented group communication and collaboration across time and space. (J. Newman, 2001, p.165)

Workflow management of the publishing process is an essential characteristic of the Content Management System. Websites supported by a single individual are commonly maintained using page-editor tools such as Dreamweaver. The CMS is characteristically deployed in an environment supporting collaborative authoring and workflow control of the publishing process. The architectures established to moderate content submitted to the *Stanford Encyclopedia of Philosophy* illustrate the ways in which collaborative workflow management can be achieved in an online context to solve a variety of research problems. In this case an online publishing resource allows submission of articles and uses email-based alerts and online editorial access to manage the peer-review process.

Obviously, in a voluntary online publishing environment, efficiencies in the moderating and peer review process are of the essence. Thus the architecture of their Content Management Systems is focussed on an intuitive publishing workflow that matches the peer-review paradigm, situated in an online environment. Hammer's paradigm for presenting the *Stanford Encyclopedia* comprises:

- The use of online CMS-style collaborative tools to facilitate the peer review
- The delivery of the content in a standard "fixed" form that is amenable to mirroring at multiple sites
- The generation of "fixed" versions of the encyclopaedia in order to address the issue of citational persistence. (Hammer & Zalta, 1997)

Such a paradigm for publishing is specifically oriented to a free-access online collaborative publishing project. The lack of access restrictions and the use of non-dynamic content forms in the published results make the choice of runtime engines much more flexible: their focus is in the implementation of collaborative tools for the content management and version control stages. Davis et al (2001) suggest that an effective global perspective of information flows is required in order to achieve effective collaborative workflow systems. Necessarily, therefore, the dynamics

for information sharing are highly contingent on the environment for which the workflow is modelled.

Heitmann, exploring this workflow process in the context of Television production, extends the intermediary role of the Content Management System to encompass all elements of pre-production and post-production work:

> Through the integrated management of audio, video and metadata, Content Management Systems can support the workflow of television production in all stages and thus render it more efficient. This requires the CMS to provide a suitable front-end application for each partial process – for recording, in-house distribution, research, browsing, rough editing, post-production editing, distribution management and for the cataloguing of essence in the archives. (Heitmann, 1999, p.25)

In a survey of research in Workflow Systems, Stohr and Zhao (2001) categorise workflow systems into two types: "production and administrative" workflow management systems suitable for routine, clerical tasks with a high degree of standardization in the process; and "ad-hoc" workflow systems, and collaborative tools, which require a high degree of user control over the workflow processes and sequencing (Stohr & Zhao, 2001, p.286).

Perez and Rojas (2002), in a detailed exploration of Workflow systems provide a set of functional indicators (both technological and organisational) for the evaluation of workflow technology. These provide a useful benchmark for evaluating the organisationally contingent factors in the workflow orientation of a product. These are explored further in the prototype evaluation presented in Chapter Seven.

Content Management System workflow could generally be characterised as *ad-hoc* in the sense described by Stohr and Zhao (2001). Of particular note is the emphasis by Stohr and Zhao on the managerial issues surrounding workflow, which he places co-equal with the technical design issues, with as many as 50% of workflow projects failing for management-related issues such as the lack of sustained management support, rigid bureaucracies, poor change management. In particular, the workflow role for a Content Management System will derive from a knowledge management paradigm more evident in engineering document sharing systems and in that context

needs to manage the flexibility required for effective integration of disparate document types (J. Davis et al., 2001).

The complex authoring and user-interaction environment involved in Online Learning development represents one of the richest examples of complex workflow systems targeted at effective reuse. As discussed, the SCORM model from ADL defines a rich metadata vocabulary combined with systems elements for integration with Learning Management Systems. This represents a workflow environment for creating (and sometimes delivering) the Curriculum framework. Online Learning authoring systems such as WebCT incorporate a highly structured workflow environment for structuring the curriculum development in an online environment, and a runtime engine for personalised delivery of the content.

Most CMS packages provide at least a simple workflow management environment. A simple web-based workflow may be characterised by the separation of the roles of content author and content editor, giving separate rights to edit the content and initiate the "publishing" process. More complex editorial workflow can include security levels to differentiate types of editing access: to specific parts of the site, specific pages, designer access, and who can publish new or modified content.

The integrated workflow paradigm of the software is an important "value-add" to the publishing. It is one of the key elements that CMS suppliers emphasise in differentiating their offerings. All CMS products examined above provide at least a linear workflow in which:

1. Designers supply template-based designs.
2. Authors supply content, sometimes collaboratively.
3. Editors review this content and initiate publishing.

The Content Management System either publishes this content at arms length to the web server, or marks it as available for use by a dynamic content server.

Where a content runtime engine entails restricted access to particular groups, the CMS may also serve a role in managing the Authentication restrictions of the runtime engine itself in terms of defining end-user access to content.

Collaborative authoring requires the capability to *check in* and *check out* or lock multimedia components as they are being developed. Some authoring tools include such capability, either by following formal standards (such as WebDAV) or through their own conventions. For instance, the Macromedia authoring tool Dreamweaver includes WebDAV facilities to check in and out content, although the feature is not strictly enforced. Dreamweaver implements an informal check in/check out facility through ftp by creating a ".lck" file to indicate the lock status and lock author on a file. Again the convention is not strictly enforced, and will of course only be recognised by other Dreamweaver users.

The EBEL at University of Sydney illustrates the complexities of workflow when dealing with heterogeneous document sources, with a strong intersection between editorial workflow and production workflow. The construction of workflow systems in this context can comprise multiple phases of content parsing, semantic discovery, linkage building between multimedia elements and various phases of content editing and transformation, yielding a complex and challenging workflow sequence (Patrick et al., 2000). Each of these stages may require a set of workflow rules with relevant intervention and task management. The contingent nature of workflow is illustrated by the EBEL project. This project involves the multimedia regeneration of multi-lingual dictionary resources from heterogeneous multimedia sources. The editorial intervention in this publishing system involves the transition of content through workflow stages mediated often by a single editor.

2.8.4 Summary

It is apparent that content reuse can serve a role in the long-term management of content and the mitigation of difficulties associated with obsolescence of particular encoding and runtime

presentation formats over time. No digital encoding form or markup ontology has remained stable over more than a few years. While the fragmentation of content at the point of economic utility will serve the needs for content management in a particular organisational context, further content transformation may also be required to meet changing requirements for finer granularity of content management. Transformation is therefore integral to the ongoing management of content:

> *Principle #13 - Content reuse over long periods requires a framework that facilitates transformation of content into new media formats.*

2.9 The Challenge

This literature review has explored the theoretical issues surrounding content management specifically in the context of regeneration and reuse of content. In the process of this review a number of principles relevant to content reuse have been derived and are summarised below. Key theoretical problems include the heterogeneous nature of digital encoded content, the need for ontological structures to coherently organise content for reuse, the need for workflow systems to facilitate content authoring, and the reality of continuous transformation that is demanded by regeneration and reuse of content.

The systems challenge is to provide a framework in which these principles for content reuse can be addressed effectively towards establishing an industry model guiding future implementations of a CMS and a benchmark for evaluation of existing systems.

> Principle #1 - Content reuse requires the organisation of content in a manner that allows the separation of content from its final presentational form.

> Principle #2 - The delivery of digital content in its final presentational form will involve some form of transformational process.

Principle #3 – Facilities for content discovery and structured capture of semantic information to facilitate content discovery are essential to content reuse.

Principle #4 - Content syndication and reuse rely on agreed ontologies for the description of semantic content. Industry implementations remain domain specific, but there is a convergence toward RDF. The ability to address multiple semantic ontologies remains important.

Principle #5 – A consistent document exchange ontology with strong community support is an essential prerequisite to content exchange. The CMS should accommodate the flexible mapping of content and metadata to accepted document ontologies.

Principle #6 - Stability of the document ontology over time will improve content discovery and reuse.

Principle #7 Content discovery is facilitated by:

- Semantic descriptors (using controlled vocabularies) supplementing the original content.
- Support for complex query structures.
- The ability to persist search content queries and their resulting document collections.

Principle #8 - Multimedia content reuse characteristically requires flexibility in the support of a heterogeneity of document models for authoring, publishing and syndication.

Principle #9 - The transformation engine should be designed to support a flexible, possibly multi-stage, sequence of transformation processing - that is, a flexible generative process. This can be regarded as the "publishing workflow".

Principle #10 - The end user should be masked as far as possible from the technical aspects of the publishing process to encourage currency of content.

Principle #11 - Fragmentation of content is essential to achieve content reuse, although the granularity of this fragmentation is contingent on the reuse objectives. The results of this fragmentation are *content units*. The choice of a level of fragmentation has resource considerations for the ongoing management of the content. A finely granular fragmentation of content necessarily adds complexity to the document description of content and the ongoing management of content relationships. A "chunky" or coarse treatment of content at the whole document level simplifies the document ontology but limits the extent of reuse. Fragmentation to the word or phrase level may not, in many contexts, be semantically meaningful. The balance between the utility finely granular fragmentation and coarse fragmentation leads to the concept of the *economic content unit*, which is the optimal level of content fragmentation for the ongoing management of content in a particular context.

Principle #12- The issue of content reuse goes beyond specification for a document exchange ontology and requires a systems approach that is currently characterised by the Content Management System. The core functions of content management that support content reuse are:

- The modelling of the document ontology in a manner that separates the informational element from the presentation and structure.
- The management of additional semantic metadata with the documents for content discovery.
- The management of the workflow processes whereby content is authored and updated in a manner that meets the editorial requirements of the organisation (including the use of staging servers to review content in its proposed context).
- The versional management of content at the micro (content unit) and macro (web site) levels.
- The management of rules pertaining to content transformation, distribution and personalisation.
- The processes associated with archival management of the content after its currency expires to allow content discovery and reuse in other contexts.

Principle #13 - Content reuse over long periods requires a framework that facilitates transformation of content into new media formats.

These principles are used to formulate a systematic model for content reuse. The methodology for the research guiding the design and evaluation of this model is presented in the next chapter.

Chapter 3 - Research Design and Methodology

"The lump of earth lost its own form, and acquired that of a man, who opened his new made lips to reveal the shape of things to come." Ovid. Metamorphoses. Book XV.

Chapter One introduced the research issue addressed in this thesis: developing a model for systematic content reuse and regeneration across heterogeneous multimedia objects. Chapter Two then explored the theoretical and industry context of the Content Management System, providing the foundation for the design of the research design and research methods employed in this thesis.

The area of Content Management is one of rapid industry innovation. In this context it is a challenge to formulate a research project that goes beyond purely technical innovation to offer a coherent theoretical view relevant to current developments. The opportunity presented in this thesis is to provide a systematic theoretical underpinning to an important area of information management in conjunction with a strong foundation for further research.

3.1 Research Methodology

3.1.1 Methods of Inquiry

Senn (1998) discusses the dichotomy between Academic Research and Practice, particularly contrasting the two different roles:

Academic Research	Practice
Rigor overshadows relevance Emphasis on the research process Values broad representative samples	Solving current problems Emphasis on best practice Relevance overshadows rigor Values situational cases (Senn, 1998, p.26)

Table 3.1: Contrasting academic research and practice

This thesis attempts to address this dichotomy by proposing an effective model for long-term content management that is informed by current industry developments and evaluated through a substantial prototype implementation in an industry context framed in the Software

Development Research model proposed by Nunamaker, Chen, & Purdin (1991). A detailed literature review informs the development of a model for content reuse. For this purpose a multi-methodological approach to evaluation is applied. A theoretical model is formulated drawing on the principles for content reuse explored in the Literature Review. A prototyping process of Software Development Research is used to provide a "proof of concept" examination of the theoretical model. The proof of concept prototype is evaluated qualitatively to elicit the completeness of the prototype as a representation of the model.

Historically, the predominant method of inquiry in Information Systems (IS) research was the investigation of efficient computational methods (Hirschheim & Klein, 1994). Such a research endeavour is suited to a strictly empirical mode of enquiry with an essentially functionalist focus on Information Systems Development. The subsequent discourse on the interpretivist and Social Relativist situation of Systems Development within organisations challenged the notion that all information systems development was directed to the efficient implementation of computational systems and saw the emergence of a range of information modelling approaches that attempted to capture the contingent role of information systems in the workplace.

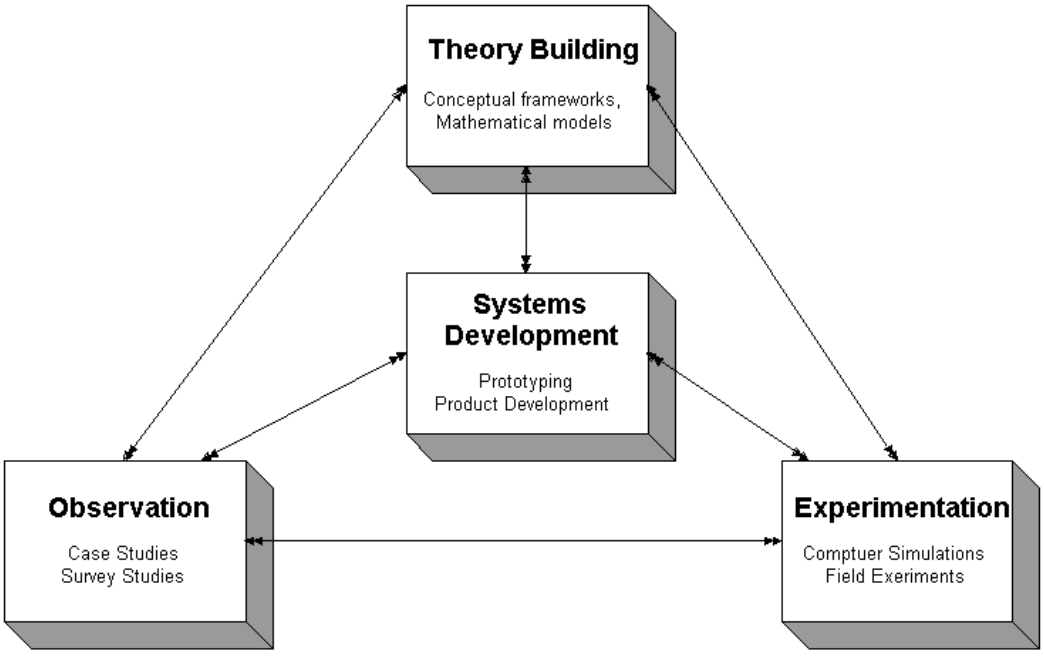
Xu (2000) characterises two approaches to Information Systems Research: the science paradigm and the systems paradigm. The science paradigm roughly equates to the empiricist tradition in research with a focus on analytical thinking of a reductionist nature and an empirical approach to inquiry. Xu's systems paradigm roughly equates to the interpretive traditions in research. Of particular relevance to this thesis is Xu's focus on the missing cognitive and evolutionary dimensions in traditional Information Systems Development Methodologies:

> The ISD process can be viewed as a semi-structured cognitive activity of a social group; therefore the communication ability of an ISDM is so important because of the need for ongoing interactions between designers and users. (Xu, 2000, p.109)

The focus of the science paradigm has been analytical modelling of Information Systems, the benefits of which are well established in the formulation of computationally effective Information Systems. However, Xu also explores the "systems" paradigm of Information Systems Development:

> Human factors and social processes which contribute to software process dynamics have been modelled and represented by pure analytical models. In fact analytical modelling cannot adequately address the behavioural aspects of social processes. (Xu, 2000, p.111)

Kuhlthau (1993) explores the contingent role of behaviour in the use Information Retrieval systems. For economic reasons if nothing else, library systems (cataloguing, online database and digital database) tend to have a monolithic interface. Kuhlthau observes that the implementation of the systems assumes a greater consistency among the population of users of these systems than actually exists and their actual usage across different disciplines is very different. The difficulty in Information Systems Development is the realisation of the contingent aspects of a model across different populations of users. This aspect of Information Systems Development has also been heavily emphasised in areas of Interface development and Workflow theory (Sonnenwald & Pierce, 2000). In particular, Wilson extols "epistemological pluralism" in order to gain a richer insight through Information Systems research (D. W. Wilson, 1998). Nunamaker, Chen, & Purdin (1991) discuss the ways in which Information Systems can affect and change the way in which people think and approach problem solving. Their model for a systems development research approach situates the process in a multi-methodological context, placing systems development and prototyping in a framework which emphasises its interaction with the process of building abstractions ("Theory Building"), the process of computational and lab experimentation ("Experimentation"), and qualitative analysis through case studies, surveys and field studies ("Observation") - in a multi-methodological context with complex feedback between each of these (see Figure 3.1).



Adapted from (Nunamaker et al.*, 1991, p.94)*

Figure 3.1: Systems Development Research

They are not alone in valuing a multi-methodological approach in Information Systems research:

> Each methodology empirical (survey and case study), descriptive, literature review and modelling plays a useful role in furthering international operations management research, be it through discovery, discussion or integration (Prasad, Babbar, & Calis, 2000).

Galliers similarly affirms the positive value of using such a multi-methodological approach in order to achieve effective Information Systems research (Galliers, 1995).

The research in this thesis grew organically through a process of inquiry through theoretical research and field implementation (and its evaluation) in an effort to enhance the evolutionary feedback of a prototype in informing a conceptual model. The attraction of prototyping in this context is the ability to adapt the model implementation in a manner that continues to communicate its intent while meeting the behavioural and workflow aspects of users in a particular context. The research approach employed in this thesis follows a multi-methodological approach as a means of enriching the enquiry process.

3.1.2 Research Approach

The background to this research has been established through a comprehensive literature review that forms the background to the formulation of a model for Content Management, and explores the definitional, ontological and information systems aspects of the problem statement that this thesis addresses. The literature review addressed both relevant conceptual literature toward formulation of this information in pragmatic categories (Iivari, Hirschheim, & Klein, 1998), and current industry practice. In the process, the literature review provides a theoretical background for a systematic model based on an analysis of the requirements for content reuse and regeneration presented in this thesis, and incorporates conceptual elements drawn from relevant Information Systems (IS) domains. The literature review situates Content Management Systems in the context of other Enterprise Information Systems: Document Management Systems, Knowledge Management Systems and Digital Library Systems.

Emerging from the literature review were thirteen principles for effective content regeneration and reuse. These principles set the stage for the development of a model that provides a systematic, model-driven approach to content regeneration and reuse. The author does not aim to chart a homomorphic systems model embracing *all* applicable domains. Rather, the thesis seeks to draw on elements of convergence in each of these areas that are as yet poorly realised in current models for content reuse and regeneration – nicely characterised by Hofstadter's "Conceptual Skeletons" (Hofstadter, 1979) - a set of homomorphic features which will be the invariant core of a complete model for content management for regeneration and reuse. The actual formulation of the model described in Chapter Four is based on the theoretical analysis resulting from the literature review, and informed by the outcomes of an extended prototype implementation.

The second stage of the research comprises a process of Software Development Research comprising the development and evaluation of a prototype to provide a mechanism for evaluation

the theoretical model. Chapter Five presents the design of a prototype to test the proposed model. Chapter Six introduces the Montage project, the context in which a prototype exploring the model was developed and tested. Chapter Seven then evaluates the model in the context of industry practice explored in the literature review and by quantitative evaluation of the results derived from the field-testing of the prototype implementation. The evaluation also draws on the survey of current CMS implementations to evaluate the functional completeness of the prototype in describing these current applications. Wynekoop and Russo (1997) argue that industry practice should be included in the Information Systems Research process – given the obvious fact that practice has often led theory:

> The IS research paradigm has shifted to recognize the legitimacy of the interpretive tradition (Walsham, 1995). We propose another shift to include practice descriptions as an acceptable supplement to rigorous research included in journals and conferences. (Wynekoop & Russo, 1997)

Chapter Five describes the detailed prototyping design – a rich process in which the theoretical research informed the directions of the software development and was in turn informed by the outcomes of the field implementation of the prototype in the Montage project. The model is evaluated through a prototype implementation and evaluation of the results of the field test quantitatively and qualitatively.

3.2 Evaluation

3.2.1 Software Development Research

The credibility of an Information Systems model is, of course, best tested through some form of implementation. Fetzer (1999), discussing the formal role of the model and the implementation in the philosophy of computing science situates it succinctly thus:

> "While the conceptualization of specifications falls within the context of discovery, as we have found, the process of empirical test falls within the context of justification." (Fetzer, 1999)

The key research element of this thesis therefore is the design, field implementation and evaluation of a prototype using Systems Development Research Methodology to test the sufficiency of the model in meeting the needs for content management, regeneration and reuse in a specific industry context. The prototype focuses on the particular exigencies of Internet publishing, with a focus on text, image, audio and video content collection management for web- and email content delivery.

Nunamaker, Chen and Purdin (1991) give prototype system development a key role in Information Systems research:

> Implementation of a system is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project. The process of implementing a working system can provide researchers with insights into the advantages and disadvantages of the concepts, the frameworks and the chosen design alternatives. (Nunamaker et al., 1991, p.100)

Nunamaker and many others have mapped the software engineering stages of effective software development research: constructing the conceptual framework, developing a systems architecture, analysis and design of the system, construction of the prototype and observation, feedback and evaluation.

Apart from the testing specific algorithmic solutions to problems, Software Development Research generally tends to follow an iterative process, which reduces the risk and allows greater contingencies for change in the direction taken than would be characteristic of a pure empirical test of hypothesis. There are interesting connections between the process of prototyping and that of Action Research. Action Research addresses issues of the experimental nature of action taking, and of the iterative feedback and learning cycles implicit in this research model. Just such feedback and learning cycles are characteristic of the prototyping development process (Baskerville, 1996). The prototype development path is one of active engagement in which the process itself serves to enrich the understanding of the participants and the developer regarding the

nature of the function being prototyped. Aspects of action research are remarkably relevant to the prototyping process:

> The cycle of subject learning is also critical for developing the knowledge necessary to claim any idiographic usefulness for the theory under test in the action research project. During the learning cycle, the subjects acquire learning about the immediate problem situation. This leads to imperative modification of action-taking and sustains the action research cycle. Without subject learning, the action research cycle ends prematurely. (Baskerville, 1996)

There are clear similarities between the research cycle of Action Research and the iterative cycles of development during prototype software development, particularly in the context of a field implementation.

This thesis therefore engages Software Development Research to test a model for content reuse, comprising a process of:

- Constructing a theoretical framework through the literature review.
- Developing a conceptual model representing the system context of the project.
- Analysis and design of the system.
- Prototyping of the system.
- Observation and evaluation of the results.

Hirschheim, Klein, and Lyytinen (1995) characterise prototyping as "…an abstract, simplified model of a future, more comprehensive system" and as such situates it as a functionalist approach to Information Systems Development with strong elements of Social Relativism, particularly in the degree to which it emphasises cooperative design and learning, communication and user understanding" (Hirschheim, Klein, & Lyytinen, 1995, p.93). Prototyping, and particularly the delivery of a rapid evolutionary implementation of the software can facilitate addressing a situation where there is a rapid rate of change in the human perceptions of requirements and needs. The issues associated with prototyping as a means of experimental design once again bring it closer to the paradigm of Action Research:

> Social relativism suggests that IS are symbolic interaction systems and any intervention in the living organization produces irreversible effects in the minds of the affected people. In that sense, there can be

> no prototyping because prototyping is a form of social intervention whose effects are irreversible. (Hirschheim et al., 1995, p.94)

Information Systems prototyping exercises, being multifactorial in their very nature, present large elements of risk in all stages of the project specification, coding and implementation, often from elements that are uncontrollable (Ewusi-Mensah, 1997; Newman, 1996, Newman & Sabherwal, 1996). What the prototype can achieve is a measure of the degree to which the final theoretical model is "implementable" as well as a learning environment for enhancing the theoretical understanding of the model. In the context of the necessary constraints applied to the scope of the prototype, this process still yields a measure of the sufficiency of the model in a particular context: that is, the degree to which the model adequately includes all essential elements that would be required to realise a specific implementation. Thus, while prototyping cannot, in the empirical sense, be regarded as a test of the hypothesis presented in this thesis, it is used in an interpretivist sense to measure and test the implementability of the model and enrich the understanding of the model, or in Fetzer's sense of a "justification" (Fetzer, 1999).

Of course, the linear nature of thesis presentation belies the iterative nature of the enquiry process, and the ways in which the literature review, prototyping and industry survey informed each other in a complex feedback cycle of research. It remains, nevertheless, an effective way of organising and coherently presenting the fruits of this research.

One of the most difficult tasks in undertaking a field-based prototype implementation as an element of Software Development Research is the selection of a case that will appropriately address the issues of interest to the researcher while meeting the needs of the relevant community. Other difficulties include the selection of a project that is achievable within the limits of the available time and means, and that is amenability to progressive review as the project develops; and yet can still be driven by the conceptual model (rather than concern with the degree to which the particular case can be generalised).

The advantage offered by a substantial field trial (as against a lab experiment) was the significant opportunity for exploring the external validity of the model. While the approach has inherent weaknesses (in the degree to which the internal validity of the model is established), the scope of the field implementation was significant and can provide a stronger validation of the external aspects of the model than would be possible through laboratory-based experimentation. This stage of the research extends the nomothetic (generalised theoretical) modelling by testing the sufficiency of the model in fulfilling the functional (technical) and behavioural (workflow) needs of a demanding web-publishing environment. A suitable field trial can also act as a "critical" exemplar that points the way to generalisation of findings (Miles & Huberman, 1994). The field trial explored in this thesis, the Montage service, is discussed further in Chapter Six.

The functional dissection that emerges from the literature review also forms an element in the Software Development Research, serving as a measure of the completeness of the theoretical model against industry practice. That is, each of the functional elements described in each of the surveyed systems should homomorphically map to the theoretical model presented irrespective of the specific terminology or industry characteristics of the systems as they pertain to content reuse. This process is not so much that of measuring the "degree of fit" of existing software to the model, as the extent to which the model effectively captures the current state of development while setting a framework for understanding this class of software in a thorough systems modelling sense. This also serves to test the value of the model in realising a generalised framework for understanding the role of Content Management in effective content reuse and regeneration. This qualitative evaluation therefore plays a part both in the Software Development Research (mapping functions to a new model for content reuse), and in the subsequent evaluation of the research outcomes.

The qualitative evaluation provides a measure of the degree of completeness of the prototype implementation as a specific semantic interpretation of the model. In order to measure the effectiveness of the proposed theoretical model, the final stage of the research is a quantitative evaluation of the effectiveness of the model in enhancing content regeneration and reuse.

Drawing on the model, generalised metrics and quantitative measures are proposed for the measurement of information currency and content reuse. These metrics form the basis of a quantitative evaluation of the prototype and the degree to which the field implementation demonstrates the key aspects of content regeneration and reuse. A post-implementation review of users of the prototype implementation using the Technology Assessment Model provides an indication of the end-user satisfaction with the prototype as a tool to facilitate the publishing process.

Chapter 4- Theoretical Model

> *"In the centre of the world, situated between earth and sky and sea, at the point where the three realms of the universe meet, is a place from which everything the world over can be seen, however far away, and to its listening ears comes every sound…" Ovid.* Metamorphoses *Book X.*

Chapter Two introduced the "Content Management System", a new industry category of enterprise software, and presents a set of theoretical principles underpinning content reuse. This Chapter takes the set of principles for content reuse derived from the literature review in Chapter Two toward the synthesis of a model for content reuse. These principles are underpinned by several factors: the heterogeneous nature of digital encoded content; the need for ontological structures to coherently organise content for reuse, the need for workflow systems to facilitate content authoring, and the reality of continuous transformation that is demanded by regeneration and reuse of content. Content Management Systems currently in the market exhibit a broad range of functions. Good document management practice necessitates content versioning of documents and in some cases of entire websites. Most CMS products have facilities for workflow control of the publishing process, staging of content and facilitation of the publishing process. The arbitrary accretion of functions would not itself be of significance if it were not for the growing importance of content reuse across generational changes of publishing targets, language versions and media forms. This Chapter presents a conceptual model directed at achieving systematic regeneration and reuse in a manner guided by the principles discussed in Chapter Two. The model is organised as a layered presentation, with each layer building in complexity on the underlying framework provided by the lower layer. It aims to direct the mind of the software architect, implementer and publisher toward the long-term benefits of content reuse and regeneration in the context of the Content Management Systems that they deploy. The purpose of each layer is to present the minimally discrete set of functions that could be "substituted" by different standards encompassing a given layer. In particular, the objective of the model is to provide a systematic approach to content reuse.

4.1 Relevance of the Model

As the importance of effective content reuse is elevated by the growing value of knowledge as an organisational asset (particularly for publishers, business consultants and educational institutions), organisational planners are faced with the dilemma of effective selection and development of suitable architectures for managing this asset. A good conceptual model can assist software developers by providing a clear framework for selection of technological approaches at each layer and can encourage the use of standards-based interfaces. Such a model can serve as a basis for guiding further development of systems in this area and as a benchmark for judging the functional completeness of products. This process can involve the interaction between business planners and technical specialists. In many organisations content authoring is dispersed, and its capture cuts across lines of business and organisational layers. This selection process can be facilitated through the use of a good functional model for Content Management informed by relevant standards. Privileging content reuse rather than the explicit feature set of the CMS is the most important characteristic of this model.

The objective of the model is also to facilitate the understanding of technological options at each layer both to reduce complexity and to encourage software modularity and standards based implementation at each layer. Organisational managers are most likely to be focussed on the requirements of the top layers of the model: the issues of publishing, editorial control and content syndication. Technical staff may be more focussed on issues of content capture and the integration of heterogeneous tools for content authoring. The conceptual model provides a common ground for planners and software programmers. The objective of this model is to be sufficiently technical that programmers can relate it to their implementation environment, yet sufficiently general that analysts can make sure the business processes derived from the model are meaningful in many organisational contexts. Finally, the conceptual model is a starting point to

ensure the overall objective of a Content Management System is satisfied in a manner that addresses flexibility of content reuse and mitigation of issues of technological obsolescence.

Industry efforts in building Content Management Systems characteristically focus on the discrete functional challenges they are addressing: original publishing, content update, efficient content delivery, etc. Any model to fully describe a Content Management System must reflect not only the database elements of the content store but also the systems elements of the production process. Content Management in a web site context can architecturally be described as a hybrid between hypermedia and an information system (Fraternali, 1999).

The explicit focus of this model is content regeneration and reuse. In this respect, simple website maintenance of content could be regarded as a trivial application of the model. This model evolved through the review of literature, current software implementations and through the deployment of a practical prototype that was field-tested over a substantial period of time. The prototype was used for both primary generation of website content, site replication based on "white site" models, secondary syndication of content in different forms (e-zines, mail outs and content syndication) and regeneration of the website over several generations of design. The lowest two layers focus on the organisation of content in a manner that facilitates the organisation of content for systematic reuse. Layers three and four focus on the presentation of these content structures at the points of mediation between the author and the content consumer.

The urgency for the project emerged from publishing issues experienced at the British Council Australia in publishing an online educational service, and at the University of Sydney in building systems for regeneration of multilingual language resources (Patrick et al., 2000). The Montage project at the British Council Australia, the framework for the prototype testing of the model, is directed to content reuse in an explicitly web-based framework of publishing.

A key feature of this model is the disaggregation of content to achieve an Economic Content Unit (ECU). Separation of content from its presentation and structure has long been the

goal of descriptive markup. The author proposes that there are advantages for content reuse in the disaggregation of content into semantically meaningful fragments for HTML, XML and multimedia objects of other types, irrespective of their encoding or markup. These advantages are in the capability this delivers in content reuse in different formats and content regeneration across generational boundaries.

The disaggregation of content, and its reuse in different content documents, raises the issue of how in the long term the usage and identification of this content is tracked. To resolve the problem of the ephemeral nature of digital content publishing in its final form, content databases need to address issues of unique content identification in a manner that facilitates content exchange between various content stores for purposes of the downstream tracking of content usage. Content editors may also need to be able to understand how content has been used in different sites, and to track changes in relationships between content in its published form.

The cost of meta-enhancement and disaggregation of content can vary for different content formats. Textual content is obviously the most amenable to content disaggregation and meta-enhancement, including some automatic discovery of structural components. Video presents yet greater difficulties: the greater complexity and cost in automated content discovery, the difficulty of preserving meta-content with the video content itself, the challenges of storage, and the variants generated for purposes of content streaming in different media formats. The economics of this disaggregation process are a factor in the manner of the reuse, hence the *economic* aspect of the content unit.

There is also a domain-specific element to the economics of disaggregation. A small commercial website of less than a dozen pages may not benefit from disaggregation of content at a lower level than the whole web page. The entire web page is the ECU of smaller Website editing CMS products. Corporations attempting to address presentation of content across multilingual boundaries (for example a product description presented in several different

languages of essentially the same product) and consistently in different locations of their website can benefit from a lower level of content disaggregation. The linguist and researcher may be interested at the sentence, word or part-of-word level of disaggregation (represented by qualitative data analysis and linguistic content management tools).

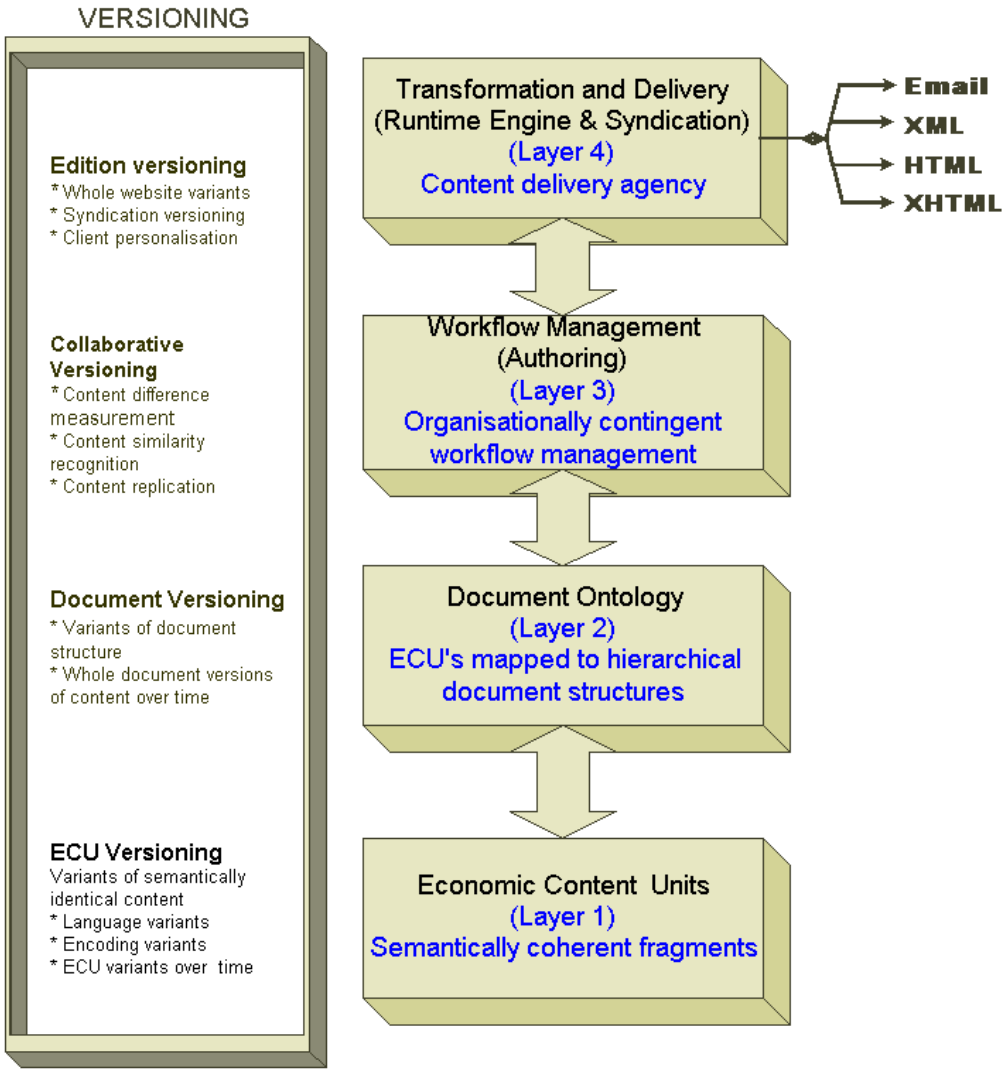


Figure 4 1: CMR: A model for Content Management, Regeneration and Reuse

4.2 The Content Model for Reuse (CMR)

The following section presents a formal model for content reuse. The model is presented as a four-layer architecture for structuring content for systematic reuse (Figure 4.1). The lowest layers (one and two) address the content primitives – the Economic Content Unit and the ontological model for document organisation. The third and fourth layers address the public interfaces of the model to the content author and the content consumer.

The first layer addresses the issue of content fragmentation and organisation. It applies encoding and metadata standards to separation of content fragmented in a semantically meaningful and economically useful way. The Economic Content Unit is the content primitive for the model. It describes a content resource (complete or a fragment) managed as a reuse object at the degree of fragmentation that is both semantically meaningful and economically effective for reuse. Integral to the ECU are the content itself and encoding and semantic information pertaining to the content.

The second layer addresses the ontological organisation of these fragments into meaningful document structures. The editing, management and reuse of content is not meaningful except in a document structure. The objective of this model is to present options for a flexible document ontology through which a variety of Document Object Models (DOM) can be realised. The function of the document ontology at this layer is to facilitate the generalised organisation of the ECU in a manner that facilitates the creation of semantically meaningful document structures supporting transformation to a particular DOM. The document ontology described below assumes an acyclic ECU content tree through which content can be represented in a DOM, with associated editorial and usage metadata, rights information and other content management data enhancement of a particular content unit object (also represented in ECUs).

The third layer is the point of mediation with the content author, in the organisationally contingent workflow management of the authoring process. The sharing of content in a

distributed authoring environment characteristically entails some form of workflow management of the authoring process. This layer describes the organisationally contingent workflow aspects of content editing, editorial approval and publishing, and the management aspects for content composition, manipulation and tracking. The workflow management process is a point of mediation between the content author and the reuse objects.

The fourth and topmost layer addresses the delivery of content in generated form to various runtime engines in the delivery of content to the targeted content consumers. The object of content authoring is, at some point, its use in some published form. This layer presents the technological framework for delivering the content in its final expression. Content may be reformulated in formats that include email, web-based, CD-ROM-based or XML syndication of the content, or regenerated as a new edition of content. The Content Delivery layer is the point of contact between the reuse objects and the targeted content consumers mediated by the organisationally contingent transformation and delivery model.

Content versioning is an issue that spans all layers of the model. Versioning, an intrinsic element of reuse, appears at all layers of the model. These layer reflect the principles for content regeneration and reuse discussed in the literature review (Chapter Two) pertaining to content fragmentation, document organisation, the process of content authoring and the transformational process for content generation and regeneration.

Principles relating to content fragmentation are:

- Content reuse requires the organisation of content in a manner that allows the separation of content from its final presentational form.
- Content discovery, and semantic information that facilitates this discovery, is an essential element of content reuse.
- Fragmentation of content is essential to achieve content reuse, although the granularity of this fragmentation is contingent on the reuse objectives.

Principles relating to document organisation are:

- Content syndication and reuse rely on agreed ontologies for the description of semantic content. Industry implementations remain domain specific, but there is a convergence toward RDF. The ability to address multiple semantic ontologies remains important.
- A consistent document exchange ontology with strong community support is an essential prerequisite to content exchange. The CMS should accommodate the flexible mapping of content and metadata to accepted document ontologies.
- Stability of the document ontology over time will improve content discovery and reuse.
- Content discovery is facilitated by:
- Semantic descriptors (using controlled vocabularies) supplementing the original content.
- Support for complex query structures.
- The ability to persist search content queries and their resulting document collections.
- Multimedia content reuse characteristically requires flexibility in the support of a heterogeneity of document models for authoring, publishing and syndication.

Principles relating to content authoring are:

- Facilities for content discovery and structured capture of semantic information to facilitate content discovery are essential to content reuse.
- Content discovery is facilitated by Semantic descriptors (using controlled vocabularies) supplementing the original content, support for complex query structures and the ability to persist search content queries and their resulting document collections.
- The end user should be masked as far as possible from the technical aspects of the publishing process to encourage currency of content.
- The issue of content reuse goes beyond specification for a document exchange ontology and requires a systems approach that is currently characterised by the Content Management System. The core functions of content management that support content reuse are: modelling, semantic metadata management, authoring workflow management, publishing workflow management, temporal content management and versioning.

Principles relating to content publishing are:

- The delivery of digital content in its final presentational form will involve some form of transformational process.
- Content syndication relies on strong ontologies, which achieve industry acceptance.
- The transformation engine should be support a flexible, possibly multi-stage, sequence of transformation processing - that is, a flexible generative process. This can be regarded as the "publishing workflow".
- Content reuse over long periods requires a framework that facilitates transformation of content into new media formats.

4.3 Web Services as a instance of the CMR

The CMR is a model for an information system. Content reuse requires more than a comprehensive semantic description framework (such as RDF or SCORM). The following sections provide high-level definitions of the model in terms of Web Services and a document ontology expressed in XML for each of the layers of the model.

Web Services are a key initiative of the W3C driven by very active industry-driven development of standards. While the standard is still in transition, the opportunities it offers have seen very rapid emergence of client-side and server-side development tools and a body of example implementations. The term "Web Services" has a range of meanings from the very general to the functionally specific. Lu, Dong, and Fotouhi (2002) describe Web Services as "web sites that do not merely provide static information but allow one to effect some action or change in the world". However, by "Web Services", this thesis is addressing that set of protocols (called "Web Services") that enable the discovery and integration of business functions for use by applications and accessible through the internet (Chung, Lin, & Mathieu, 2003; Geng, Gopal, Ramesh, & Whinston, 2003).

A key element of the Web Services framework is the Web Services Description Language (WSDL), itself a four-layer architecture providing the framework for software delivery which:

- Modularises and encapsulates well-defined standard interfaces.
- Can be managed and run locally or remotely.
- Can be ported over the Internet or intranet using standard protocols above TCP/IP (and through other delivery channels).
- Can be discovered from central registries in a manner that fully describes their service.
- Exhibits "plug-and-play" characteristics at the client level. (Geng et al., 2003)

In many respects, Web Services present a good example of the types of reuse and discoverability that should be typified by content reuse systems. The approach therefore provides a good framework for documenting the reuse model presented in this Chapter. Web services interfaces will be described below for each layer of the CMR definition.

Web Services provide an exemplary discovery and software reuse infrastructure for defining application logic accessible over Internet standards. As such they are a useful means of expressing a high level an instance of the CMR model. These Web Services functions are accompanied by document definitions expressed in XML for each of the layers of the model. This Web Services interface is implemented as part of the prototype described in Chapter Five. Other interfaces are also presented for purposes of authoring integration and workflow management, particularly at the third layer of the model. A detailed Web Services definition is provided in Appendix Two. An implementation of this Web Services interface is an important element of the "Inter-Publish" prototype derived from the CMR mode.

4.4 Layer 1 -The Economic Content Unit

The purpose of this layer of the model is to create a content primitive which is most economically effective for content reuse. Chapter Two presented the semiotic distinction between:

> *The nature of the content.* Hjelmslev distinguishes between form and substance of that which is signified. In the context of digital content this can be usefully recognised as the encoding of the content (its *form*) and the meaning of the content (its *substance*).
>
> *The nature of the expression.* The published result is a process of transformation that renders a digital multimedia document for use in a runtime engine. This final expression can be understood both in its substance (the meaning communicated and the relevant encoding of metadata) and its form (the presentational format suitable for the runtime engine).

This distinction is useful in reflecting on the utility and consequences of disaggregating content. This disaggregation would have only archival value were it directed simply to the primary generation of content. The driver for this fragmentation (Boll & Klas, 2001), or "chunking" (Boiko, 2002), is content reuse. This content reuse at the simplest level may be expressed in terms of the reuse of navigational and stylistic constructs in the generation of websites for purposes of consistency of presentation. Content reuse may be directed to the regeneration of content at points of generational change of websites, or syndication of content to other agencies in an acceptable form. Norrie & Singer (2003) discuss the granular dissection of content toward producing consistent content creation in multimedia forms (such as paper and CDROM delivery) and the need for a level of granularity in the content to allow cross-integration between media forms – such as from digitally augmented paper back to digital media resources such as CDROM or web. The fragmentation of content necessarily entails the management of digital encoding information

and metadata concerning the content and its placement relative to other content fragments. The metadata description of content addresses both the description of the content format and the encapsulation of the content with sufficient additional information to understand the substance of the content in its context with other content units.

The separation of content from its presentational style has been the explicit goal of descriptive text markup technologies such as SGML and XML. This is particularly directed to the nature of the final expression. Published digital content may comprise a heterogeneous mixture of content encoding and markup strategies applied to the generation of content in a final form. Fragmentation of content across different multimedia formats offers some of the benefits of content separation from style that has so richly enhanced the management of text. In some cases this might imply the separate management of fragments of XML (for example a product description across language boundaries deployed in different XML objects). Similarly a digitally encoded image might be used in several different resolutions across a single website (for example a company logo) but with a required level of consistency in presentation.

The objective of this model for content management is to retain importance of the intentionality expressed by the publishing of content in its final form while disaggregating the content in a manner that enhances its deployment in the context of other published content. Content capture is a non-trivial process in a heterogeneous media-authoring environment. The consolidation of content into a single database framework will invariably require the interaction of a range of content formats, authoring tools and content markup systems. With content reuse as an objective, the first prerequisite for effective content reuse is the decomposition of content into elemental Economic Content Units (ECUs) that can be reassembled in different styles and presentational forms. The lower the level of disaggregation, the more complex the modelling required to re-aggregate this content into another forms. This essentially *economic* consideration leads to the *economic content unit.* The disaggregated ECU's are not orphans. They belong to one or more document structures. The information pertaining to ontological structure of binding

Economic Content Units to form semantically meaningful documents is the subject of the second layer of the model.

4.4.1 Definition

The *economic content unit (ECU)* is the primitive of the Content Management System.

The economic content unit layer (layer 1 of the content model) describes a collection of multimedia ECU entities. Whether this collection is retained in a database or on a file system (or in a mixture of both), the ECU comprises:

- A unique identification (for each ECU)
- Encoding metadata associated with the ECU (including, with text resources, the character encoding and language characteristics, contextual information such as the DTD reference for XML fragments)
- Semantic metadata (to facilitate content classification, discovery and reuse)
- Rights metadata (for purposes of rights management)
- Usage metadata (to track usage of a particular ECU over time and over place of deployment)

An ECU might represent: a fragment of HTML, a branch of an XML tree, or a particular multimedia fragment (for example a GIF-encoded image, a video segment, or an on-demand audio fragment). Figure 4.2 provides a functional definition of the ECU, and Figure 4.3 presents an XML schema describing the ECU.

The ECU collection **C** is:

f_i = a data set containing a single encoded content fragment

$f_i = \{c_i, e_i, m_i, u_i, h_i\}$

c_i = encoded content item

e_i = encoding metadata (for example the MIME-type, dimensions, codec)

m_i = semantic metadata for this encoded fragment (for example using RDF)

u_i = rights metadata for this encoded fragment

h_i = usage metadata this encoded fragment

ecu_i = a non-null set of fragments $\{f_1..f_n\}$ describing semantically identical fragments (for example language variants, codec variants)

$ecu_i = \{f_1,\ldots,f_n, d_i,\ldots d_n\}$

d_i = semantic metadata descriptors describing this ecu

Figure 4.2: Definition of the ECU

Content editing and publishing functions may, at the lowest level, address a particular ECU encoded fragment f_i. This could include a single image, a video clip at a particular encoding level, a single cell in a column of a relational table (for example an email address), or a text fragment in a particular language encoded form.

4.4.2 Interfaces

This layer of the content model focuses on the integrative aspects of managing a heterogeneous collection of Economic Content Units. While public interfaces would not be expected, Web Services functions for the low level exchange between content services are desirable. Such content exchange requires the encapsulation of:

1. The digitally encoded content itself, or the referencing of content which permanently resides in external content sources
2. The metadata description of the content in terms of its encoding
3. The metadata description of the content in terms of its context in the wider document ontology (see layer 2)
4. Any rights, ownership, usage and descriptive metadata relevant to this content.

4.4.2.1 Web Services Functions

The principal application of a Web Services interface at this layer of the model is to provide a framework for information exchange at a low level with other CMS products. A detailed WSDL description of this interface is presented in Appendix Three. The following Web Services functions provide a low-level primitive for content exchange between CMS products. The *cmr_Selectionrule* (see Figure 4.4 below) provides an XML schema definition for select on content using the relevant web services selection functions (see Table 4.1 below) at the ECU layer (for instance, for low-level content replication between CMS products).

Function	Request	Response	Purpose
GetECU	*cmr_Selectionrule*	*cmr_Eculist*	Fetch (optionally with lock) an ECU
UpdateECU	*cmr_Eculist*	*cmr_Eculist*	Create/update changes to an ECU (releasing locks)
ReleaseECU	*cmr_Selectionrule*	*cmr_Eculist*	Release a list of ECU entries

Table 4.1: Web Services functions at layer 1

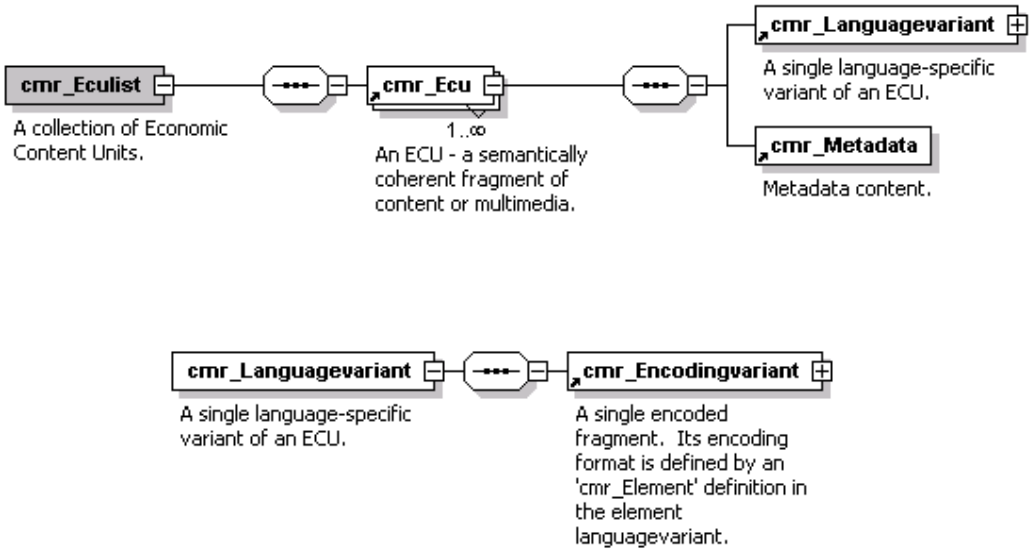


Figure 4.3: ECU schema

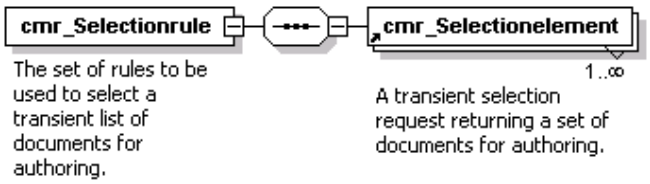


Figure 4.4: ECU selection rule

4.4.2.2 Summary XML structure of document interfaces

The *cmr_Eculist* describes a document model for the Economic Content Unit, comprising variants in language and/or encoding. A summary schema appears in Figure 4.3 and 4.4 above,

and a full XML Schema is presented in Appendix Two. The XML document *cmr_Selectionrule* provides a transient selection rule for a set of *cmr_Eculist* documents.

4.4.3 Discussion

There are a many of approaches to the encoding of multimedia objects, from proprietary formats such as Macromedia Flash to text encoding standards such as Unicode. There is a similarly rich range of standards for encoding of semantic metadata in the digital environment. Bandwidth limitations remain a significant factor in video delivery of content on the Internet. Text itself can be presented in a multitude of encoded forms, from the international double-byte Unicode character representation in a device-independent manner to highly proprietary graphical presentation through image or vector graphical representation. For instance, a text fragment may be richly encoded through using Macromedia Flash with a result that achieves a high level of fidelity to the vision of the author at the cost of ease in subsequent editing and update. A document model may, therefore, need to accommodate a range of content encoding representations even for text resources.

4.4.3.1 Text encoding

Unicode has emerged as the pre-eminent choice for encoding of original text for multi-lingual support. There are several text markup approaches that have reached wide adoption by industry heavyweights. At least two have emerged as leading proprietary industry formats for text description: Microsoft *Word* and Adobe *Portable Data Format* (PDF). Both these formats place a premium on the final presentation of the content. *Word*, coming from the tradition of descriptive markup in Word Processing systems of the 1980's and 1990's is a de-facto standard for personal and corporate correspondence and document capture. The more portable *Rich Text Format (RTF)* has emerged as a popular method for printed document distribution on the Internet, with its capabilities for strict preservation of the presentational styles and limitations on editing of content in its final form. While SGML has been overshadowed by XML for content markup, it also has a

track record on facilitating better text management. The early adoption of SGML for the markup of literary texts (the Text Encoding Initiative or TEI) has resulted in a large body of text content available in a consistent markup form. The utility of the document ontology of XML has seen its ready update in many arenas for content interchange, and increasingly also for archival content markup for permanent retention of documents – primarily due to the widespread availability of programming resources to traverse XML content structures. HTML, in common usage often an errant implementation of SGML, is the ubiquitous markup methodology for content presentation of web-based information. Some encoding methods lend themselves to disaggregation into smaller ECU's. The hierarchical nature of XML and SGML make such a disaggregation possible in some circumstances, in which case an associated DTD fragment may be required for coherent authoring of the content.

Multilingual computing has commonly been an afterthought of software design, and the World Wide Web is no exception. The inadequacies of HTML in representing multiple languages in a singe-page context have led to many add-on approaches to language representation, including particularly MHTML (Maeda et al, 1998). The emergence of Unicode as a content standard for text character encoding has been driven by the enormous challenges of multilingual content representation in Digital Libraries, Document Management systems and in the effective interchange of content in a manner that preserves and presents its encoding with adequate fidelity. This is a considerable advance on the over-loading of formatting and printer instructions characteristic of Operating System language codes. Some encoding ontologies have their own lists of country codes (such as the *USMARC Code list for Countries),* while in the website context, the international standard ISO 639-2 provides codes for languages and groups of languages. The ISO standard is probably the most advanced, and has the capability of representing major language groups and sub-groups – unlike the earlier ISO 3166 with 2 and 3 digit country names. Even ISO 639 is not exempt from criticism - "…even *cmc* and *but* are not comparable because

cmc refers to a whole language family but *but* only refers to members of the Bantu family" (Constable & Simons, 2000, p.9).

Digital expression of language has seen major improvements in universality of support with the growing adoption of Unicode at the national level and among Operating System vendors and software developers. The semantic management of text content has also seen strides with the widespread adoption of XML as a method of structural and semantic markup of text content. Nevertheless, the reality represented by the plurality of possible formats in which text itself can be represented makes it likely that many, if not most, environments directing their efforts toward content reuse will need to accommodate a plurality of text encoded document fragments.

4.4.3.2 Multimedia encoding

The plurality of formats for representing text is mirrored in other multimedia formats. Audio, video, music, dance and other forms of encoding show only emergent signs of standardisation with complexities arising from the need for efficient transmission encoding techniques over limited transmission networks. Semantic encoding of non-text objects shows an equally diverse range of emergent approaches. The object of the CMS is not necessarily to understand the encoding of the multimedia primitives, but certainly to describe them and facilitate their reuse.

The concern of the ECU is to capture in a plurality of encoded formats:

- Semantic information pertaining to the ECU
- Explicit or implicit rules pertaining to the storage, encoding/decoding of the ECU
- Language characteristics of the ECU
- MIME types/runtime engine handling rules for delivery of the ECU

4.4.3.3 Versioning

The ECU may be versioned across language, encoding and editorial variants. A language version describes a semantically identical content item encoded and designed for a particular

language context. Language variants can apply to images and other multimedia objects equally as well as to text. The ECU may also have versioning across content items encoded to different levels of fidelity or resolution. Finally, editorial versioning expresses the tracking of content changes over time to a semantically identical content unit. The encapsulation of all multimedia fragments in a framework that facilitates versioning has considerable value at all subsequent layers of content reuse.

4.5 Layer 2 - Document Ontology

An ECU taken in isolation is an orphan until its content is understood in a document ontology that defines its semantic and presentational context**.** A final published multimedia document may comprise a heterogeneous set of text, images, audio and video selected for a particular level of language or media encoding. Each component of this document (the text, images, audio and video) may itself be a collection of Economic Content Units linked together in a semantically meaningful way. The final transformation that generates a multimedia document may well draw on individual ECU's for existing static document structures, which form part of the final published multimedia presentation. The acyclic network of linked ECU's comprising a particular document or partial document is the subject of the second layer of the CMR model. A content editor working on a multimedia document is selecting semantically related resources. The acyclic nature of the document structure is important to prevent recursive relations between content that prevent effective transformation of the content for presentation (Boll & Klas, 2001).

A Document Definition Language defines the structure of content items in the multimedia context. The flexibility implicit in SGML supporting Renear's overlapping hierarchies (Renear et al., 1999) and Boll's content fragment trees (Boll & Klas, 2001) provide examples for the design of complex models for representation of heterogeneous content resources. The Document Definition Language provides the framework for describing the content structures that are allowed within the system. Such definitions may be implicit (as with authoring tools like Dreamweaver)

or explicit (such as in the more generalised content model described by Boll and Klas (Boll & Klas, 2001) and the model presented in this thesis. This definition may be expressed in a syntactical form (such as DTD) or through a modelling interface to facilitate the definition of content items, controlled vocabularies and metadata associated with particular content components. The explicit realisation of a flexible document definition capability is essential to this layer of the model, and builds on the ECU layer to provide a methodology for content reuse.

In this model the document ontology of the CMS describes the framework for meaningful relationships between ECU's. Different multimedia types have different Document Object Models (DOM), and the purpose of a flexible document ontology at this layer of the model is to allow flexibility in brining ECU content together in different DOM structures.

The encoded object represented by an *ECU* may well reside in an operating system directory rather than a database object (such as a relational table).

4.5.1 Definition

The reuse relationship layer brings together semantically related ECU's for management and transformation – for example as a "news" item or a "chapter" of a collaboratively produced book. Figure 4.5 below presents a specification for a Reuse Document Type (RDT), which maps the structure of a generalised document object comprising ECU relating to metadata, design specifications and editorial content. The Reuse Document Type brings together both content and metadata, and is necessarily an acyclic tree of ECU elements and relationships - that is, while a RDT may itself participate in other RDT's, this relationship cannot be recursive or cause a link in the document structure back to a higher point in the document tree. This specification is realised as an XML schema described in Figure 4.6. The purpose of the specification and schema is to provide a generalised (but acyclic) method of describing different document hierarchies organising ECU resources.

rdt - *Reuse Document Type* = $\{e_1...e_n\}$ – acyclic hierarchical document ontology model for a document structure

e_i = ontological element describing encoding and metadata structures for a given ECU

rdo - *Reuse Document Object* = $\{col_i, rdt_i, w_i, h_i, m_i\}$ where

$Col_i = \{ecu_1 \ldots ecu_n\}$ comprising an ECU *collection* represented by acyclic network of ECU's which describe this document object model. This document network may be statically prescribed or dynamically selected with rule-based selection (for example XPATH/XSL).

rdt_i –the particular document template for this document, representing the framework for interpreting this particular document model

w_i – *Workflow* metadata pertaining to usage rights for this rdo

h_i – *Usage* history for ths rdo

m_i – *Metadata* information (for example using RDF) describing this reuse document

Figure 4.5: Document Ontology

4.5.2 Interfaces

4.5.2.1 Web Services Functions

Expressed as a web service, the interfaces relevant at this layer are those that support management of the document ontology. In the example below a "document type" describes a Reuse Document Object (RDO) in terms of a set of ECU "elements". An ECU element might itself be a description of an RDO structure, giving a recursive hierarchical capability to support complex document forms. The flexibility of the management of the ontology is a key aspect of content reuse. It is the enabler for subsequent content management and reuse. The Web Services example in Table 4.2 below defines a recursive document definition structure the elements of which at any given layer of the hierarchy may be an ECU definition, a link to another document structure or the edge of a database table connection.

Function	Request	Response	Purpose
UpdateDocumentDefinition	*cmr_Documenttypelist*	*cmr_Documenttypelist*	Create/Update a Document Definition
GetDocumentDefinition	*cmr_Selectionrule*	*cmr_Documenttypelist*	Fetch document definition(s)
GetElementDefinition	*cmr_ Elementlist*	*cmr_Elementlist*	Create/update an Element definition
UpdateElementDefinition	*cmr_Selectionrule*	*cmr_Elementlist*	Save Element definition(s)

Table 4.2: Web Services functions at layer 2

4.5.2.2 Summary XML schema for document interfaces

The *cmr_Documenttypelist* provides the encoding definitions for elements in a document framework defined by a *cmr_Documenttypelist* (Figure 4.6). The interfaces at this layer are principally for the management of the ontology. The *cmr_Document* brings together a collection of Economic Content Units under the ontology described by the *cmr_Documenttype* (see Figure 4.7). The XML schema is described in more detail in Appendix Two and an instance of its realisation is described in Chapter Six.

4.5.3 Discussion

4.5.3.1 Document Ontology

The document ontology for definition of the document template will have a significant bearing on the degree of flexibility in supporting a wide range of media objects in the CMS implementation. Many existing CMS products implement a reuse document template based on the HTML Document Object Model (DOM). An abstracted approach to the document template is designed to support multiple document object models.

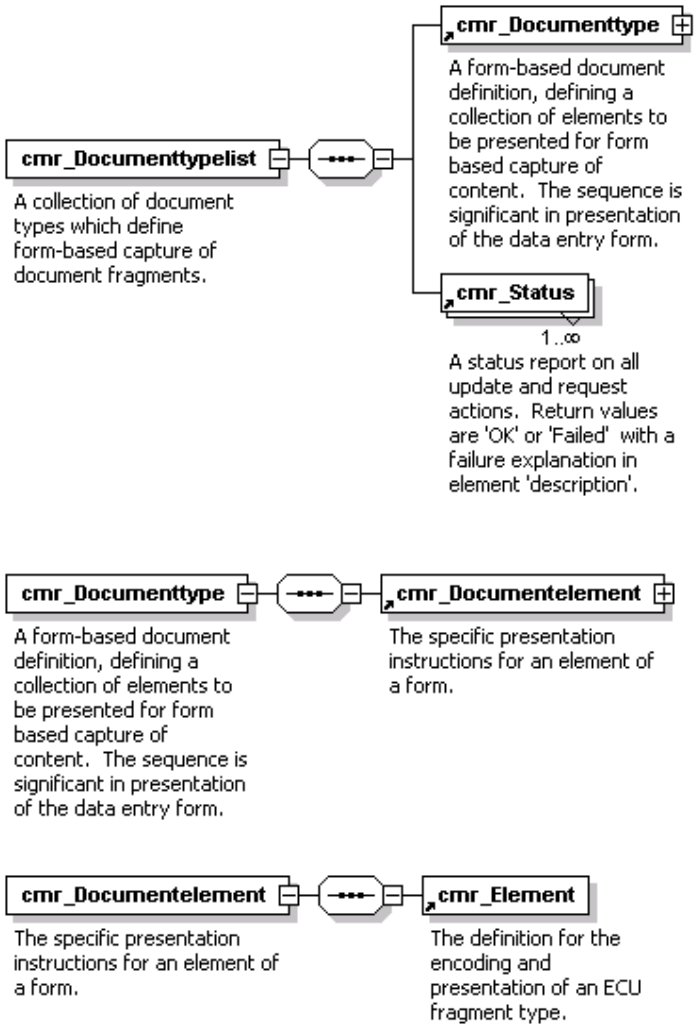


Figure 4.6: Schema defining the generalised document ontology specification

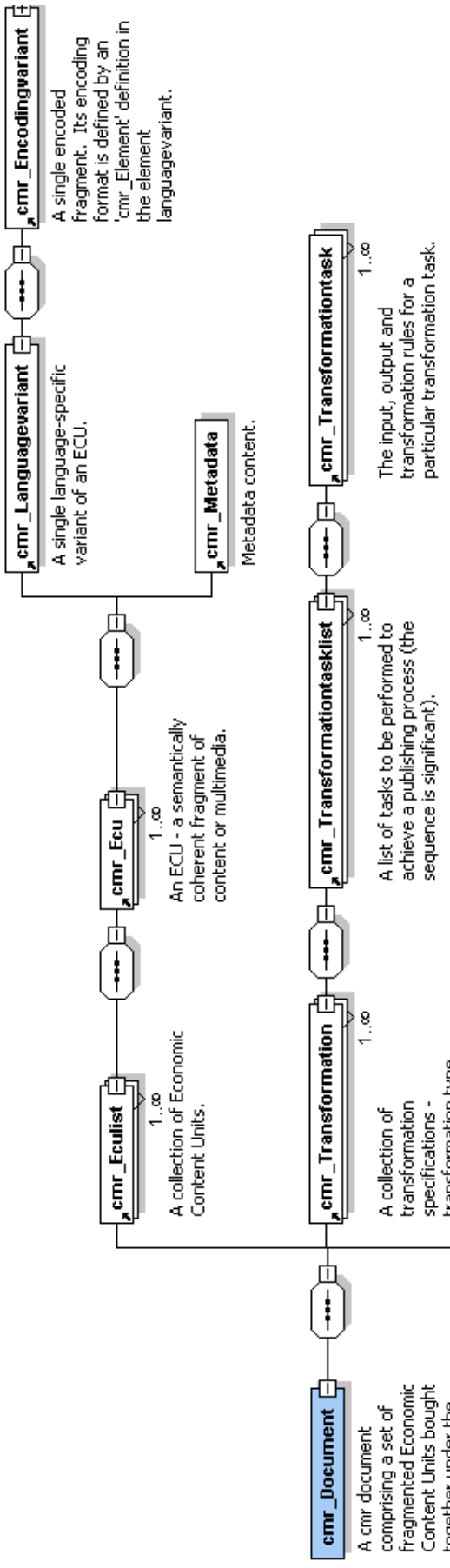


Figure 4.7: Document Schema

The document ontology must not only provide a flexible framework for supporting complex multimedia document objects, it must also provide a means to simplify the process of content reuse. It is in this context that the following document ontology is presented - an ontology not aimed at a finely granular implementation of a particular DOM but at the management of multiple DOM frameworks implemented to a sufficient level to support the ECU.

Public interfaces are not relevant to this layer of the model. However the DOM object must include the ownership and access rights pertaining to the use of the document within the workflow environment. The description of the *RDO* will require an effective, probably database-oriented, implementation of the document object model, which describes the relationship between content units for the purposes of content workflow management and control.

4.5.3.2 Versioning

Just as Economic Content Units can have variants across language, encoding and editorial dimension, so also the document ontology adds a layer of versioning complexity. The Reuse Document Type (RDT) may itself be versioned, with the obvious implication that changes to the document model may itself affect the management of the ECU collections that are members of an instance of a document model. The collections comprising a reuse document object (RDO) may also be expressed in editorial versions over time.

4.6 Layer 3 -Workflow Management and Authoring

A *workflow management layer* (CMR layer 3) describes the publishing and versioning workflow required to generate a particular presentational form. The information system design at this layer of the model will be situated in the contingent factors of editorial workflow. This layer of the model presents the functional characteristics that are commonly understood to describe the Content Management System.

4.6.1 Definition

The editor will access content in the framework of the document reuse objects defined at the second layer of the model, constrained by temporally defined access rights, based on a selection framework used to discover the content. A transformational process may be required to map encoded content to the particular editing framework used by the content editor, yielding a particular content view. This view is described in Figure 4.7 and described as an XML schema in Figure 4.8 below. The XML Schema for this layer of the model has the added complexity of static collection management (Figure 4.9), transient content selection (Figure 4.10) and workflow management (Figure 4.11).

view – a particular expression of an RDO document mediated by transformational rules to allow management of the content in a particular authoring language and document environment as governed by workflow rules.

$view_i = f\{rdo_i, w_i, tran_i, auth_i\}$

Where

rdo_i – represents the resource document being managed

w_i = the workflow rules governing the management of the document, including *a priori* actions and *a postoriori* metadata gathering relating. For instance, approval of a document may change its content form but not result in any final expression. It might trigger a generative workflow sequence as part of a chain of actions pertaining to a broader publishing workflow. Alternatively, final editorial approval might trigger the creation of a final expression document.

$tran_i$ – the transformation encoding and decoding to exchange content with the content author

$auth_i$ – the content author.

Figure 4.7: Content workflow framework

4.6.2 Interfaces

Rich public interfaces are required in order to support the many elements of workflow and authoring. While many elements of workflow are contingent to the organizational context, there are established practices in workflow modelling which improve the flexibility of the workflow process. Included in this definition is a simple ontology for linear workflow management with data-driven over-rides.

4.6.2.1 WebDAV

WebDAV is a tool commonly used for collaborative editing of content resources. Its design illustrates some of the underlying requirements of a workflow process. WebDAV began as a protocol founded on the in-situ editing of content with the web server acting as both content repository and runtime engine. Leveraging HTTP (HyperText Transfer Protocol), its first generation function was simply to provide a means of authenticated access for authoring of website content (Whitehead & Wiggins, 1998). The RFC has moved rapidly to a proposed standard published in February 1999 (RFC 2518) and has achieved widespread adoption in authoring tools, particularly Macromedia's Dreamweaver and Microsoft's FrontPage. In its simplest form, WebDAV allows the client to request the content of a published resource, edit this content and publish it back to the resource. The protocol level implementation is not greatly different from GET and POST in HTTP, with extended calls supporting property discovery and resource discovery, locking and unlocking - PROPFIND, LOCK and UNLOCK. (Whitehead & Wiggins, 1998)

The protocol uses well formed but un-validated XML transactions as extensions to the standard HTTP architecture (Hunt & Reuter, 2001). The exigencies of collaborative content development required the introduction of protocol extensions to incorporate versioning and version control. The DeltaV protocol for versioning first prototyped by Hunt (Hunt & Reuter, 2001) was approved in October 2001 as a proposed standard.

WebDAV could be regarded in this way as an early precursor to SOAP and Web Services (see below). The protocol opens up the more strictly confining HTTP protocol to a more generalised exchange of information through XML that fully implements the XML namespace. That it is implemented in an HTTP protocol environment makes the development of a server architecture interfacing through to a content store a more straightforward process.

WebDAV leverages the base protocol of HTTP for the receipt and delivery of content, adding to it the protocols for editorial access and version control. WebDAV is an illustration of one type of public interface to this layer of the model, and is amenable to presentation of a virtual view of the content resources, although still bound to a "file system" view of content structures. Its advantage over the newly emergent Web Services is the established desktop support for the WebDAV protocol in many Operating Systems and applications.

4.6.2.2 Browser-based client server interface

The browser is the ubiquitous client to many information systems. Its lightweight client-side runtime framework makes it ideal as an authoring interface to this layer of the model. The browser-based resources access paradigm is well accepted by users, as are the limitations of its graphical user interface.

4.6.2.3 Web Services interfaces

The Web Services interfaces for this layer of the model must support a rich layer of authoring interaction, from modelling of document ontologies to content interchange. The functions described in Table 4.3 provide the framework for workflow-managed interchange of content with authoring clients.

4.6.2.4 Summary XML structure of document interfaces

Figure 4.7 illustrates the XML schema for a single document. The third layer of the model is the point of mediation with authors and must manage the transition of one or more documents during the authoring process through potentially complex workflow stages. The *cmr_Documentlist* represents a collection of documents and is the principle content exchange schema.

Function	Request	Response	Purpose
List Documents	*cmr_Selectionrule*	*cmr_Documentlist*	Provide a document catalogue of items using a structured selection rule
Publish Documents	*Cmr_Selectionrule*	*cmr_Documentlist*	Publish one or more documents specified in *cmr_Selectionrule*, returning a status in *cmr_Documentlist*
SaveDocuments	*cmr_Documentlist*	*cmr_Documentlist*	Create or update one or more documents specified in *cmr_Documentlist*, returning a status in *cmr_Documentlist* (and releasing any locks)
Delete Documents	*cmr_Selectionrule*	*cmr_Documentlist*	Delete one or more documents specified in *cmr_Documentlist*, returning a status in *cmr_Documentlist*
GetCollection Documents	*cmr_Selectionrule*	*cmr_Documentlist* *cmr_Collection criteria*	Get return a set of documents matching a given collection
GetCollection Criteria	*cmr_Selectionrule*	*cmr_Collection criteria*	Return the collection criteria for a collection
UpdateCollection	*cmr_Collectionlist*		Create / update a collection
ListCollections	*cmr_Selectionrule*	*cmr_Collectionlist*	List collections based on criteria in *cmr_Selectionrule*
LockDocuments	*cmr_Selectionrule*	*cmr_Documentlist*	Lock and edit a document
ReleaseDocuments	*cmr_Documentlist*	*cmr_Documentlist*	Release a document (no save)

Table 4.3: Web Services functions at layer 3

Persistent collections of documents can be defined through the *cmr_Collectionlist*, for use in the workflow process and content discovery. An element of these collections is a rule-based approach to selection of content across the documents, and returning one or more documents in the form of a *cmr_Documentlist* (see Figure 4.8 and Figure 4.9). A transient collection of documents may be selected by a *cmr_Selectionrule*, being a flexible collection of selection values to manage a given collection of documents (see Figure 4.10 below).

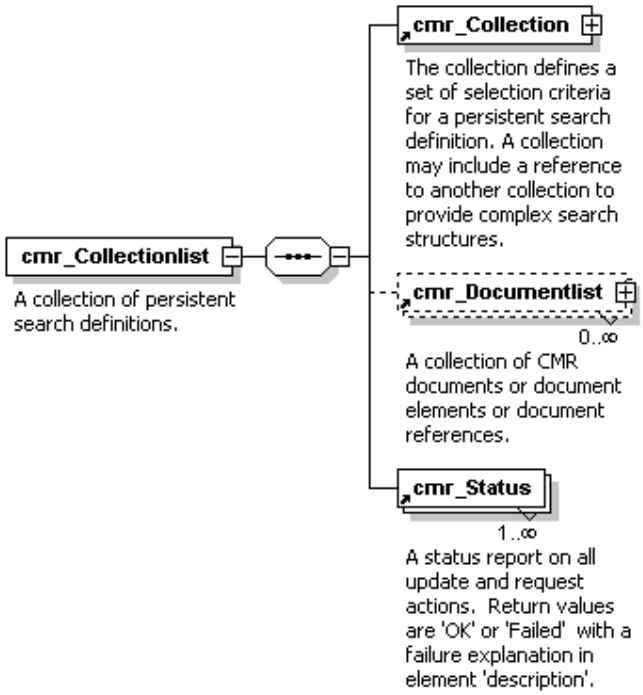


Figure 4.8: Schema for content retrieval and static collection management

The coordination of the authoring process requires a set of workflow conditions that can be tailored for the contingent requirements of an organisation. Figure 4.11 describes an XML schema for definition of such a workflow. This schema describes workflow events driven by the transition of documents from one stage to another (a linear workflow) and workflow events triggered by data-driven events.

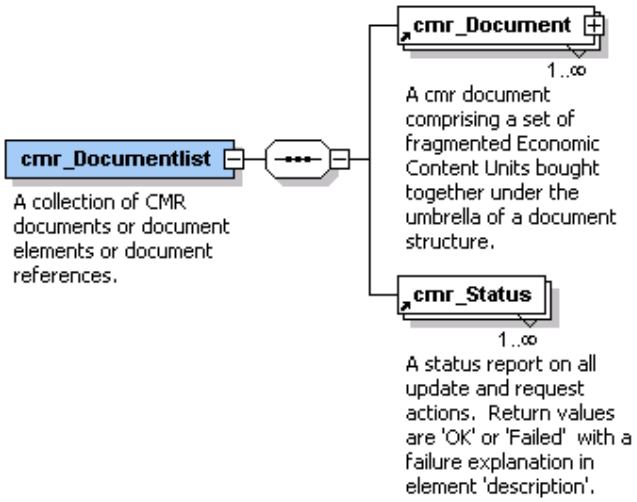


Figure 4.9: The document collection

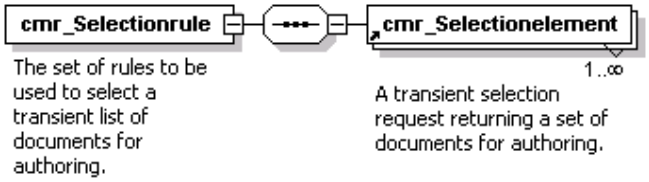


Figure 4.10: Document selection rule (for Web Services retrieval)

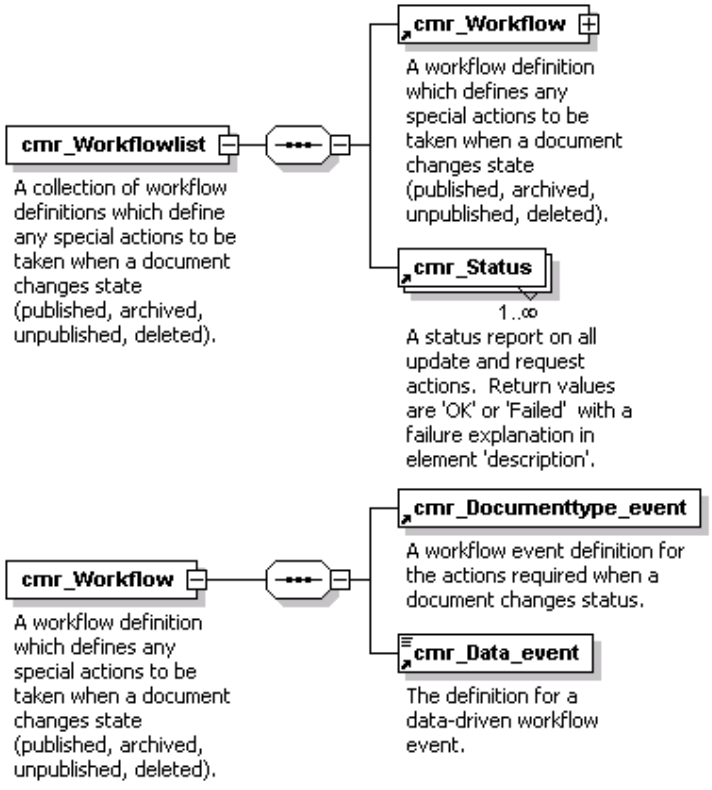


Figure 4.11: Schema relating to workflow management

4.6.3 Discussion

The Workflow layer opens the content to authoring and editing tools, placing *economic content units (ECUs)* in their document object context in a language and encoding specific manner that meets the needs of specific content authors. These public interfaces are mediated by the workflow usage and temporally defined user access rights. Characteristically, end users expect to interact with content through interfaces supplied by their existing editing tools, either browser based or through standard content delivery protocols such as SMTP (Simple Mail Transport Protocol, or email), HTTP, FTP, WebDAV and system proprietary file referencing conventions (such as the Microsoft UNC).

4.6.3.1 Content identification and discovery

Website content is characteristically already situated in an information-rich website hierarchy. Many CMS products simply address the task of publishing a particular website resource. Content discovery of website content is therefore facilitated by presentation of the existing logical website structure and known location information.

The simplest approach to content management is to treat the web server as both the content repository and the runtime delivery. A document may comprise a simple static web page with the presentational form and layout bound together as a single document. While the document remains, its URL may act as a permanent identifier. As discussed in the literature review, unfortunately the URL is not a reliable content identifier over time. Furthermore, many web pages serve an ephemeral role as navigational aids to reach target content: resource indexes, and search pages. Content reuse can complicate the discovery process, where published content may itself be the result of a transformational process gathering together multiple ECU fragments.

The workflow layer of a Content Management System must therefore facilitate content discovery at several layers of the model. One goal of the information retrieval engine is the

effective discovery of content that may be distributed across different content stores, in different languages and storage architectures. Another is the discovery of original ECU fragments where only the final presentational form is known (for instance to apply changes). Finally the information retrieval engine may serve as the vehicle for persistence of access to original content for editors and possibly for content consumers.

None of these goals is easily realised. There are ongoing debates in the Internet community regarding the methods for addressing non-ASCII resource locations, and on approaches to crafting links to physical rather than digital resources (W3C URI Planning Interest Group, September 2001). There are more profound problems in the persistence over time of resources published on the web (Lawrence et al., 2001). A potential benefit of a strongly reuse-oriented model for content management could be facilitation of long-term management of these resources by providing a framework that aids the identification and discovery of content.

4.6.3.2 Ontological framework and Controlled Vocabularies

The ephemeral nature of published multimedia content in its final form places greater demands on the Information Systems that were used to generate this content. As a result, the architecture of the Content Management System must incorporate strong mechanisms for unique content identification of multimedia content components. The organisation of the content must lend itself to the effective replication of content for archival purposes and the efficient reuse of content. This organisation may take many forms, including:

- Traditional database schemas, including relational databases.
- File system story enhanced by database-managed metadata.
- Search-engine enhanced file system storage of objects.
- XML database systems.
- Hybrid combinations of the above.

The literature review in Chapter Two explored the heterogeneous content models in multimedia authoring. Equally diverse are the tools used by the content creator (Brett & Nash,

1999; Michalas, 2000). The process of authoring can involve the complex interaction of different content authoring systems. Work by Cheung and Chanson (1999) shows the added complexity of reaching and identifying multimedia resources over heterogeneous networks. They propose three different models: configuration, user control and presentation. Multiple methods of publication extend the concern of publishers into the area of software and product support as well as the production and distribution of their content (Reynolds & Kaur, 2000). The once-off publication of multimedia content can itself be a demanding task. However, the later regeneration of multimedia content in new editions can present a challenge in every stage of the publication process (Patrick et al., 2000). Whatever the target storage and retrieval framework for the content, the Content Management Systems must therefore facilitate the effective management of the heterogeneous components of complex multimedia documents, some stored locally and some sourced externally. The multimedia content database supporting a Content Management System must therefore support the demands of content organisation in a manner that maximises content identification and dissemination of its resources, as well as effective consolidation of these resources for purposes of management and support. It must also manage of connectivity information for external content items and content repositories and the elemental organisation of content, content structure and meta-information relating to content in a manner that enhances content reuse.

Fundamental to the Content Management System (as distinct from an HTML editing tool) is a mechanism for schematically describing document structures – a meta-definitional method for describing the structure of content objects that are to be managed. This implies not only a method for schematic description of content components at the relevant level of reuse utility, it also implies a means of defining the relevant controlled vocabularies that apply to the ontological framework and transformational approaches for content capture that allow content to be edited and otherwise captured in a manner consistent with the ontological framework.

The meta-enhancement of content still requires considerable intellectual input. Automated semantic content detection has advanced considerably for text, particularly in the area of

knowledge management and semantic exploration, but still remains elusive for complex multimedia objects such as image, animation and video (Dorai & Venkatesh, 2001). With both automatic semantic analysis and human analytical allocation of semantic categories, a controlled vocabulary has immeasurable advantages over simple keyword searching but remains an expensive process that continues to command the attention of efforts in automated semantic discovery. A Content Management System must therefore provide not only the ability to meta-enhance with semantic annotations; it must also provide the facilities to manage the controlled vocabulary to be used. This is both a function of the ontological framework and the systems for management and enforcement of controlled vocabularies.

4.6.3.3 Intellectual Property Metadata

The encoding of intellectual property metadata may be as important in some contexts as the encoding of semantic metadata. Usage of multimedia content may be bounded by time (for instance a limited license to broadcast for a specific time interval only), usage (limitations to the number of times an item may be broadcast), format (limitations to usage in specific presentational forms only) and reach (limitations to the audience that may view the content).

The integration of rights management is an emergent area of CMS integration. Multimedia broadcasters must address rights management issues for many different content types. In many cases, content types may have restrictions on delivery in particular runtime forms or media environments. These rights may exist for specific, limited time periods, or for particular regions. The tracking of such rights becomes an important meta-informational element of content items. The inheritance of such rights information of content components from the original parent item, and the inheritance through to derivative content items is an important challenge.

4.6.3.4 Content Validation

Content components of all sorts contain a variety of internal coding and markup rules. They may also contain links of a variety of sorts to associated resources (for example, image

resources, hyperlinks, and so on). Content validation expresses the aspects of content management that include:

- Input validation of content captured online
- Parsing rules for import of external content
- Revalidation of associated resources to ensure validity and persistence of these resources (link checking, etc)
- Maintenance of integrity with current controlled vocabularies.

4.6.3.5 Workflow

Integral to the production process in the Content Management System is the workflow control of the editorial process. This can involve the separation of the roles of the designer, the content composer and the editorial supervisor. Such editorial management enhances the distribution of content capture throughout an organisation while maintaining the content and design standards essential for publication.

Traditional website publishing workflow can involve relatively simple linear authorisation of content through one or more stages before final generation of the content. The workflow model described in this chapter is sufficient to allow either a data-driven or linear workflow process, but is not sufficient for more complex groupware tasks such as document routing to multiple workflow channels. This process flow has been tested in the context of a Content Management System used for website publishing and in a project for regeneration of a multimedia language thesaurus.

Perez & Rojas (2000) present evaluation indicators for measuring the flexibility and capability of workflow implementations. Table 4.4 paraphrases their workflow indicators. Chapter Seven extends these indicators in the context of content reuse and the workflow design.

TECHNOLOGY INDICATORS	CHARACTERISED BY
Availability of graphic tools	Visual modelling capability in order to design the workflow process.
Routing capability	The ability for the end user to direct the content workflow to particular path(s)
Establishment of groups	The extent to which users can participate individually or in groups
Performance metrics	The tracking of the status of items at particular workflow stages and metrics on processing performance at each stage
Queue management	The ability to review and track documents and to prioritise action on documents in a queue
Management of events	The control of conditions in which workflow events will happen
Workload	The channelling of documents based on workload of document editors
Integration capability	The capability of the workflow to integrate across organisational systems
Monitoring	The tracking and alerting based on load conditions
Simulation	The ability to simulate under load conditions the workflow model
Handling of forms	The integration of models for form-based actions
Webflow	The integration of web-based tools (including email) in the workflow process
Task Management	The degree of complexity allowed in task management
Process Management	The degree of complexity allowed in overall process management
Documentation	The completeness of documentation in management and use of workflows
Hardware and software considerations	The platform requirements and flexibility of the system

Table 4.4: Workflow evaluation measures (Perez & Rojas, 2000)

This workflow characterisation provides a valuable checklist for the review of particular workflow implementations in the CMS context. Most workflow management in the web-publishing context focuses on *task management.*

4.6.3.6 Language issues

Internationalisation of CMS solutions must address not only issues of multilingual content capture but also multi-lingual User Interface issues and problems with current runtime engines.

4.6.3.7 Version control

Versioning of the content also becomes important in a collaborative work environment, as the need to view comparative changes over time becomes important. Version control of content may be required at both the content-component level and for organised collections of content.

The purpose of version control at the component level of content is the support for collaborative authoring and the ability to review changes over time. The purpose of version control over broader collections of content is to support the controlled regeneration of larger collections of content in a manner that can be consistently repeated.

4.6.3.8 Scheduling and Access limitations

Content authors, particularly those preparing news announcements and other time-critical items, will generally want a means to schedule the release and longevity of the content. The Content Management System must therefore capture the metadata related to scheduling and access restrictions, and provide these as temporal and user-based access restrictions associated with the content. Educational packages such as WebCT go further in allowing the content release to specific Workstation IP address ranges. Online database providers have similar network-identification based access restrictions in conjunction with user-based logon restrictions to limit the community of users having access to the runtime engine.

4.7 Layer 4 - Transformation and Delivery

The *content delivery* layer is the vehicle for final expression of the content. It defines the rules for interaction with the high-level presentational information system (client & server), which yields the final content version (see v_i in Figure 4.12 below*)* to the target audience. The runtime delivery layer is quintessentially the point of mediation between the end user of the content and the runtime delivery engine(s). Content expressions may have dependencies over time in relationship to other content expressions. For instance, the content editor may expect the content to be delivered only at a particular time, or only to particular users. The content delivery system must, in this context, negotiate the issues of transient network failure, and to that degree ensure the integrity of content delivery in the form of the minium level of coherent expression. Equally important is the role of the runtime engine in addressing the behavioural context of the content requirement of the information seeker. Overby (1999) criticises the simplistic reuse of text out of

its original context. The effective reuse of text in an information systems context introduces the understanding of the relationship between the information system and the intent of the original author.

Many content management implementations, particularly in a dynamic content delivery environment, make little differentiation between the information systems of CMR layers 3 and 4. A dynamic content engine may be interrogating a database dynamically and yielding a result to an end user. The point of differentiation between a dynamic, data-driven website and a CMS is precisely the point of focus on content management in the context of content reuse. The economic worth of content management lies in its ability to manage content in a manner that yields benefits in its effective reuse in a syndicated sense or in a regenerative sense. A dynamic data-driven website, like any other information system, need not necessarily exhibit either of those characteristics.

The merit of separating levels 3 and 4 of the information system lies in the systematic engineering for effective content reuse rather than the satisfaction of the demands of implementation of a particular information system, and for the task-oriented management of the transformation and delivery process for complex regeneration systems. A runtime engine may be an engine for static delivery of fully composed content (such as a simple static web server) or an information system that draws dynamically off the Layer 3 systems. Some content management products (for example Vignette and Obtree) have hybrid runtime engines that integrate with existing Web Servers. Others (such as the prototype discussed in this paper) are at arms length from the actual runtime engine.

Online Learning Systems continue to be an active area of experimentation in content reuse. Just as personalisation within a commercial web delivery framework has been a focus of website design, so we are also likely to see greater efforts at the personalisation of higher educational delivery mechanisms. Universities with a focus on building campus-wide models for effective

capture of educational objects will be the best placed to begin the delivery of effective, personalised online educational delivery.

Web-based runtime engines supported by CMS software are therefore typically either by Serving content which is statically published through a CMS generative process, or through dynamic content generation assembled for the content consumer more-or-less on demand based on the context of their information request (and which may be personalised for the particular content consumer).

A traditional website serving static pages or having minimal dynamic generation of content still represents the most speed-efficient runtime engine (Mendes & Almeida, 1998). The web server, acting essentially as a content file server, needs little interpretive intervention to deliver the content. Some web content delivery platforms, such as Vignette, have made a virtue of proprietary caching and generative mechanisms that deliver content as quickly as possible within the constraints of dynamic regeneration of the content in its final form. Dynamic online generation of content has considerable advantages for currency of content and personalisation, but requires itself a new layer of architectural support to meet the higher server load demand (Anderson, 2001). Whether dealing with a static or dynamic model of content delivery, the underlying content management issues are similar – the difference between dynamic and static content delivery is essentially one of the timing for content recomposition. The final expression of the content may be highly proprietary in form and transient in its expected longevity. The Content Management System must support both generation of content in its original form and translation to new runtime environments. An extension of this regenerative process is the long-term persistence of content in current delivery formats to address the issue of technological obsolescence.

4.7.1 Definition

The publishing process is a multidimensional transformation in one or more stages. This transformation is a process of mediation with a runtime engine, the presentation format for assembling the content and the personalisation requirements of the end user. This mediation process could be managed by a single runtime engine serving static content (and so minimal personalisation) or through a dynamic runtime engine with on-the-fly creation of pages based on the end users profile and the content resources available. The complexity of this process may be determined by the degree of personalisation required for user presentation and the point at which this personalisation occurs. A specification for this relationship is described in Figure 4.12 below and realised as an XML schema for content syndication (Figure 4.13) and content transformation (Figure 4.14). In some cases, the personalisation of content is managed through specific runtime engines, with increasingly complex mediation between the user and the runtime engine – such as through the medium of digital augmented paper (Norrie & Signer, 2003).

The final content version v_i expresses the delivery of fully encoded content in its published form to the end-users runtime engine based on one or more transformations that yield the final expression of content based on the contingent states of the reuse objects, the transformational template, the content usage rules, the distributive rules and finally the relevant profile of the target community.

$v_i = f(rdo_i, tran_i\ temp_i, dist_i, u_i)$

Where

rdc_i = $\{rdo_1 \ldots rdo_n\}$ that represents one or more reuse documents (rdo_i).

$tran_i$ = the transformational template mediating the delivery of the rdc_i document collection to the user community u_i.

$temp_i$ = the temporal rules pertaining to content usage (time/s at which the content can be considered valid)

$dist_i$ = the distributive rules by which content reaches the relevant runtime engine, including relevant security constraints

u_i = the target user community

Figure 4.12: Content transformation and delivery

4.7.2 Interfaces

4.7.2.1 Web Services Functions

The topmost layer of the model is directed to the fruition of the previous layers of content management. There is a multiplicity of runtime engines used to deliver content, the most popular being the Web Server and email. The format of content also varies, but most popularly HTML and XML. Table 4.5 describes a set of simple Web Services functions that can achieve a high-level fully encoded interchange of content at full arms length – for example in a syndication process.

Function	Request	Response	Purpose
GetDocument	*cmr_Urnlist*	*cmr_ Encodeddocument*	Deliver through Web Services a fully encoded document for presentation through another service
GetDocument Preview	*cmr_Urnlist*	*cmr_ Encodeddocument*	Delivery through Web Services a preview version of an encoded document based on the security context of the user
GetSyndication	*cmr_Syndication requestlist*	*cmr_Syndicationdocument*	Deliver through Web Services an XML encoded version of the document

Table 4.5: Web Services functions at layer 4

4.7.2.2 Summary XML structure of document interfaces

The transformation layer of the web service provides the means for transformation and exchange of documents presented according to defined ontologies. The transformation rules for presentation of this content are described in Figure 4.13. An encoded and translated document may be selected through a *cmr_Urnlist* oriented to fully formed document exchange or XML syndicated exchange according to a known document ontology (see Figure 4.14 below). The Schema is presented in full in Appendix Two.

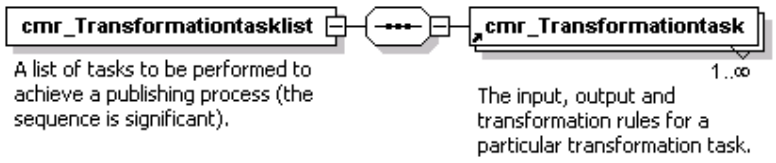


Figure 4.13: Content transformation rule-sets

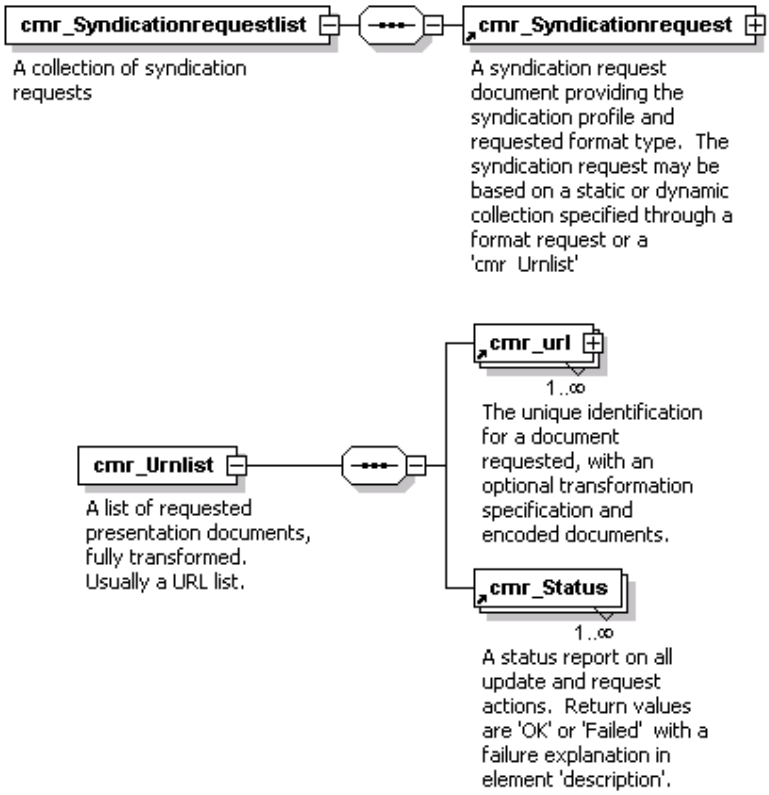


Figure 4.14: Content distribution and syndication schema

4.7.3 Discussion

4.7.3.1 Dynamic versus Static content serving

Some CMS solutions directed at web-based content delivery integrate their own runtime engines. A particular focus of such systems is the optimisation of the dynamic generation of database driven content in an attempt to achieve the sort of response times that static content servers can yield. The opportunity presented by the integration of the runtime engine is the capability to serve the runtime engine content directly from the database, in a dynamic manner

personalized to the particular client context. CMS vendors that typify this approach include Vignette and ObTree 3D.

Other systems publish to one or more different established runtime platforms for content delivery. The CMS is system integrative in the sense that the Content Management System sits astride the authoring processing and the runtime engine for content delivery. This does not obviate the construction of personalisation aspects in the runtime engine, but it does generally imply the delivery of content to different runtime engine repositories. While this may imply the duplication of the published content, such duplication is a reality of delivery to many runtime delivery contexts (for example email delivery rather than web delivery) in which the runtime engine exists on the client rather than server system environment.

4.7.3.2 Content Exchange: Moderation and Syndication

The very nature of the Internet encourages the dynamic interchange of information. Many multimedia resources encourage and facilitate the submission and exchange of information and content items, including text, image, video and other content forms. Content Moderation addresses the process of automated or user-interventional vetting of inbound content. Content Syndication is the systematic reuse of content on a commercial basis to client users.

While distributed models for document interchange exists, they have yet to address complex issues of multilingual content management in the context of the heterogeneity of multimedia forms. Information retrieval architectures such as Z39.50 provide a framework for the distributed interrogation of information resources, and this standard has recently been extended to incorporate the "digital library object". While standards such as Z39.50 have yielded good results in the content dissemination and information retrieval of library collections they highly domain specific. In the management of multimedia resources, the integral relationship between the technological platform for the information delivery and the digital content present considerable problems. XML is the encoding standard of choice for text content exchange, and already has

wide acceptance, with the availability of complex ontological frameworks for semantic markup toward syndication and reuse (such as SCORM, PRISM, NewsML and others as discussed above). The exchange of information will obviously be enhanced through the presence of a methodology for definition of an ontological framework for content exchange. Microsoft's *BizTalk*, for instance, situates itself as a syndication and moderation intermediary to facilitate the exchange of content. XML has obviously emerged as a popular vehicle for text interchange between systems, commonly using URL-style naming conventions to transfer associated multimedia content.

4.7.3.3 Language issues

The web content delivery infrastructure (including Domain Name Servers, Internet Servers and client Browsers) continues to struggle with the challenges of non-Latin scripts. While there is emergent support for some alternative encoding (such as Chinese Big 5 encoding of domain names), Domain Name Support remains largely limited to roman alphanumeric character sets. Ubiquitous support of double-byte language-encoded location identifiers has some way to go (Hoffman, 2000).

4.7.3.4 Versioning

The content management layer introduces the management of version issues at a broader level – for instance, the maintenance of consistent version information across all ECUs for a particular version of the published content. An example of the application version information at this level might include the archiving or rollforward/rollbackward of all content in a website to a particular moment of time. Conceptually this equates to the full edition versioning of the published product. Complexities at this level of versioning include the integration with other temporal database systems in a dynamic environment, which imply a lack of specific "version in time" content. However, many websites are amenable to "editioning" at a particular point of time.

The support of such edition versioning is an important element of humanities web publishing (Eaves, 1997).

4.8 Alternatives to the Model

4.8.1 XML is not sufficient for Reuse

XML is becoming the pre-eminent form for markup approach for managing unstructured textual data, and as the vehicle for exchange between information systems of both structured and unstructured data. However, most CMS products will have to deal with a variety of content formats, many of which are encoded in their own, sometimes proprietary, media format. While XML is one of the most important innovations in relation to content management in the last decade, that importance does not reach the point of making XML a necessary condition for Content Management Systems or for content reuse. Nor can XML be considered sufficient to achieve content reuse, but rather relies on several other aspects of content storage, management, transformation and delivery as discussed in the literature review. In many cases XML schema can also be represented as relational database tables either through translation of the document model to a fully normalised set of table structures, or else through the partial translation of the schema combined with the use of a Binary Large Object (BLOB) for encoding of “chunks” of the XML document. The reality of most CMS products (as becomes apparent in the industry review presented in Chapter Two) is that they are implemented in either relational database systems, or in hybrid relational database systems that use some form of fragmentation of complex document forms such as XML. As the literature review in Chapter Two indicated, content encoding of multimedia documents must necessarily encompass many different encoding, markup and metadata markup standards.

XML has gained momentum borne of its many practical successes. The ease of building XML parsing systems has resulted in a rich array of XML management technologies. The utility of XML's implicit self-documentation extends its explanatory value as a medium for information

interchange between systems. The generalised, user customisable templates offered by XSL as a style sheet for XML offers rich opportunities in the runtime delivery of XML-based information. For these reasons, XML will be an integral element of many Content Management solutions, and indeed most current syndication approaches use XML document definitions of one form or another. However, it cannot be said that XML is essential to the database management of content for reuse. Furthermore, XML remains a *markup* system and not an *information system*. The processes associated with the workflow model described in layer three of the CMR model will require information systems components, mediation with other systems and interfaces with authors. These processes will be managed through software components that may or may not have an element of XML.

When leveraging content reuse across all content types, content markup cannot be constrained to XML, although content reuse will often be realised using XML. While the advantages of SGML and XML in unstructured text documents and business document interchange are well established, the utility of XML for other media and content types is still an open issue. For instance, many "standards" for XML encoding of music have already emerged. The markup of a musical score loses many of the underlying strengths of XML in its inherent self-documenting nature. When XML becomes remotely distant from the inherent explanatory value of the tags used, the question of its utility over other more efficient markup approaches arises. Furthermore, a musical score may have many overlapping themes not necessarily easily described in XML. While it is likely that strenuous efforts will continue for an XML standard for musical score encoding, the choice of XML would seem to be associated more with its current favour and ubiquitousness rather than any essential merit of XML in this context.

For multimedia objects such as images, embedded watermarks may be XML encoded, or encoded as other encrypted objects. Many existing multimedia image standards have their own approaches for embedding metadata information although such approaches may well be enhanced through use of XML metadata.

Finally, the issue of content reuse centres not on a particular ontology, however comprehensive, but on an approach that integrates an information systems view of the problem.

4.8.2 Model-driven content management

Fraternali and others have explored concepts of information modelling as they apply to database intensive website generation (Fraternali & Paolo, 2000). They have, in this process, explored website modelling systems similar to UML oriented to website generation. Such modelling distances the site designer, at least in the first instance, from the specifics of web page naming, runtime engine structures and particular physical components of the website. It could be argued that the model presented in this chapter adds little value to database intensive content models such as those discussed by Fraternali.

Fraternali (1999) also explores the benefits of modelling as a means for the consistent definition of website structures. This research direction has culminated in a modelling ontology for website description called WebML (Ceri et al., 2000). Such modelling architectures for the visual presentation of content structures are a means of enhancing the utility of content design and management of data-intensive and well-defined hierarchies of website content. As with similar approaches to software design characterised by CASE (Computer Aided Software Engineering) tools, the suitability of imposing a modelling approach is highly contingent on the nature of the website itself. Ceri (2000) defines a language which, like CASE tools, is not only used for site modelling and conceptualisation of the interrelationship of broader structures in a website design, but realises generatively the runtime version of the website. This places a considerable burden on the efficacy on the modelling language in describing all aspects of the runtime implementation. It is likely that the utility value of modelling will be linked to specific contexts and applications of content management such as the data-intensive and highly structured sites Fraternali (1999). This level of modelling is an element of the workflow processing and the definition of the structural

relationships between collections of documents. Typically, CMS products reflect the physical website structure in the visualisation of the content structure.

WebML and UML-based orientation to website design can provide a valuable framework for ongoing management of web-based resources, particularly where such resources are patterned on reusable structures over the whole website. Just as UML has considerable utility in modelling the design of software, WebML has great value potential value in describing the formal structure of a website. This value decreases in the context of more complex, heterogeneous runtime resources that may only partially be managed by the Content Management framework. The complexity of many rich content resources, particularly media websites, makes such a universal modelling approach difficult to implement.

Using WebML, the site designer might execute all site structural changes through the modelling interface, while edition changes and site generational changes would be addressed through the explicit regeneration mediated by the modelling interface. The WebML model focuses on a unified view of the website particularly in the context of data-intensive web-based information services. Such a model could take a role in the workflow and publishing layers of the Content Model for Reuse. WebML has not achieved the sort of industry adoption that could be hoped in either the open source or proprietary systems favoured in current industry usage (see Chapter Two), partly due to the heterogenous system context of many runtime environments. At this layer, complex information systems (such as personalisation engines) may also play a role outside the delivery of a standardised website presentation. Such complexities are difficult to encapsulate in a single unified XML model of a website. However, such a modelling approach enhances a concept of structural reuse of the site design, and WebML model could be realised as an element of the top layer of the model, while also providing a framework for describing the structure of the website in the workflow layer of the model., enhancing the reuse of the structural and design elements of the website. A similar approach to this is taken in the creation of a "white site" for the Montage project described in Chapter Six. The Information Systems model presented

here, with its focus on content reuse, is not necessarily holistic in its claim on the territory of the runtime engine that delivers content to the user, and in the field implementation (see Chapter Six) forms only a small part of the wider publishing activities of the target organisation.

The CMR model is not therefore at dissonance with WebML approach to website design - although its focus is somewhat different. Many data driven website services can benefit from the structure imposed by a good modelling approach. Such websites could also benefit from consideration of the issues that this thesis presents in content reuse. Scrutiny of an information system using the model presented in this thesis allows the content management to examine the components of the online information system from a new angle: specifically their effectiveness for long-term regeneration and reuse.

Ephemeral as the content may seem in an online database-driven system, such systems can gain from scrutiny against the CMR approach. For example, a review of an e-commerce site against each layer of the model may raise the following design issues:

Layer 1: to what degree would the underlying database support extension of the online service to a multilingual context? To what degree can product descriptions be managed consistently with versioning across languages and over time? Are the same product descriptions used in online catalogues, web site resources and email mailouts?

Layer 2: The relational model of the orders file represents a document model of a sort. In many cases XML document models and relational database schemas are now interchangeable. Relational databases are not alone in being relatively rigid in their ontology – complex XML schemas such as used for B2B interchange and Learning Object exchange necessarily tend toward a more tightly defined DTD or Schema for use a particular context. The information systems framework supporting this layer of the model is the base for later content reuse. The online interrogation of the relational database can be considered a form of content reuse.

Layer 3: Many online websites are themselves information systems. E-commerce websites may well have a mixture of information content and ordering modules. There may well be tailored product maintenance facilities for management of the content specific to the information systems on the website. This may involve the distributed capture of information in the online relational database used by this application. Even in this context, the design of such information systems could benefit from the implementation of reusable content fragments that can be deployed not only from the database perspective but also through other publishing systems. Commonly, the failure to reuse these content elements can lead to online PDF-style product catalogues which do not synchronise with the online e-commerce systems. The workflow systems for managing a product-oriented e-commerce system are equally important, but often lack the interfaces to provide efficient authoring of information such as product catalogue resources. A similar problem arises in the consistent product description across different languages on the website.

Layer 4: The runtime layer warrants separate scrutiny in the management of the relationship between the particular consumer of website content and the information systems. XML profiles lend themselves to personalisation of the website experience as an adjunct to statically generated information resources. The continued publishing of some information in static form can yield greater scalability in the delivery model for the content. Dynamic generation of content can be laboured in the scalability of the generative process under load and the lack of flexibility in the end user personalisation possible. Syndication of product and pricing is a much under-utilised element in many websites. Many commercial enterprises syndicate only one content item: their share price (Stahl, 2003).

Careful consideration of the separate issues at layer 2 and layer 3 may open the expansibility of the online ordering system beyond a particular runtime context. For example, if at layer 2 the interchange architecture between the runtime engine and the relational database has been crafted in XML using XSL for the presentation of the results to the end user, then the same

XML may be amenable to delivery to other runtime architectures – such as email-based delivery of enquiry results or user customisation of report results. At layer 3, distributed content management issues could pertain to both the scripts themselves and the management of content stored in a relational table. How, for instance, will long-term archival information on order history be retained for an e-commerce system? A simple backup of the relational database may suffice for purposes of business continuity, but may not be satisfactory in terms of long-term data mining of the information resource.

4.9 Summary

The literature review introduced the difficulties facing current approaches to content regeneration and reuse: the heterogeneous nature of digital encoded content; the need for ontological structures to coherently organise content for reuse, the need for workflow systems to facilitate content authoring, and the reality of continuous transformation that is demanded by regeneration and reuse of content. These issues can only be addressed through an information systems approach as distinct from a particular document encoding method or ontology.

This chapter presents a disciplined model for Content Management directed at systematic content regeneration and reuse. The layers of the model are discussed at a broad functional level and demonstrated in terms of a Web Services interface which address the Economic Content Unit (layer 1), the document ontology (layer 2), the workflow (layer 3) and the transformation and delivery engine (layer 4).

The next chapter describes the prototype design that emerged as a semantic realisation of the theoretical model proposed in this chapter. Chapter Six then describes the Montage field trial, in which the prototype was implemented in an demanding publishing environment. Chapter Seven presents a detailed qualitative and quantitative evaluation of the theoretical results of the field implementation.

Chapter 5 – Prototype Specification

A key element of this research is the development of a rich prototype exploring the functions at each layer of the Content Model for Reuse (CMR). This chapter presents a detailed specification and design of the main elements of the prototype, its implementation, and a description of its system architecture, data and functionality. The prototype is also informed by a survey of existing industry practice and the specific requirements of the Montage project described in Chapter Five. The prototype was branded under the name Inter-Publish for purposes of the field trial. The prototype was rigorously field tested in a multi-national and multi-lingual website implementation across over 10 countries in the Montage website implementation (described in Chapter Six). The design and implementation of the Inter-Publish prototype has been limited to only those aspects of the model that could not be easily adopted from the readily available software tools, such as editors, parsers and content converters. Inter-Publish was constructed to demonstrate principally the novel aspects of the model, namely the ECU approach to content, the flexible document ontology enabled by this model and the facility with which it enables content publishing and reuse. The author was solely responsible for the software design and development described in this thesis. The British Council supplied staff in Australia and internationally for the graphical design and provision of content for the websites described here.

5.1 Inter-Publish Background and Objectives

5.1.1 Prototype objectives

The purpose of this Software Development Research is to undertake a software design and implementation process to explore and test the Content Model for Reuse, in order to:

- Address issues of inconsistency in the presentation of the same semantic information across different media

- Ameliorate the economic costs of content recapture during generational changes of content
- Reduce the opportunity cost represented by the failure to deploy content in all relevant content forms (for example through syndication).

While a lab-based trial of such a prototype would be possible, the author had the opportunity to test the concepts underlying the model in a much more exhaustive field test. The British Council Australia had a requirement to expand a simple online educational website into a resource able to support content sharing and syndication across several countries and languages with a minimum of data repetition and reusing much of the content from the original site. This requirement presented an ideal framework for a complex trial of the conceptual model in a large field test.

The prototype presented here was field tested for over three years and the versioning elements of the model (implemented very early in the prototype) yielded a rich and effective resource for evaluating content reuse over the duration of the project.

5.1.2 Prototype chronology

While key elements of the initial design of the prototype were designed against the principles arising from the literature review and development of the theoretical model, many aspects also emerged through the evolutionary prototyping over several generations of software development. While software changes were released on a roughly monthly basis to meet the exigencies of an operational server used by the clients for ongoing web publishing, major functional changes were released at intervals of between three and six months.

The following are the major generational stages of the software. Perhaps the most important stage of the project was the December 2000 release, with the realisation of the first two layers of the model. This early implementation of a framework for version history proved beneficial in the later analysis of content use and reuse over time. Much of the focus thereafter

centred on workflow, publishing and content distribution support, and the implementation of Web Services interfaces. The chronology of changes is:

September 1999 – Initial prototype implementation at the British Council including basic functionality for website publishing workflow, content management and reuse.

December 2000 – Implementation of an underlying database model reflecting Layer 1 (encoding) and Layer 2 (reuse) aspects of the content model. This architecture proved to be a stable and effective framework for the remainder of the project. The content versioning implemented at this stage was particularly useful for the subsequent detailed analysis of editorial usage and publishing over the next two years.

April 2001 – Re-architecting the user interface independent of Microsoft IIS for purposes of extension of standard interfaces to support Web Distributed Authoring and Versioning (WebDAV or Web Folders) and File Transfer Protocol (FTP) access and extension of Layer 3 workflow model. Extension of publishing capabilities at the Layer 4 (runtime engine) capabilities of the system, and to support XML content interchange.

April 2002 – Further enhancements to the site replication and modelling environment

October 2002 – Layer 3 enhancements - generalised workflow management & extended XML support

January 2003 – Layer 3 enhancements - virtual site browsing capability and extended WebDAV capability. Beta trial of DotNet web services framework.

March 2003 – Layer 4 enhancements relating to Portable Data Format (PDF) publishing.

May 2003 – Implementation of the DotNet framework for delivery of the Web Services layer.

In early 2003 the server on which the Montage content was hosted was relocated (at the request of the British Council Head Office) to the UK. The infrastructure for both the CMS and the public web server were relocated to London, while staffing responsibility for the project was also relocated initially to London and subsequently to Manchester. The new UK support team itself was relocated at this time from London to Manchester, resulting in additional transition delays. The internal security requirements of the UK operation also presented difficulties for those accessing the CMS outside the UK offices. These changes brought a nearly three month hiatus in content authoring. The initial relocation to the UK also revealed weaknesses in the international date support of the prototype.

5.2 CMR Prototype – Architecture

5.2.1 Overview

The model presented in this thesis defines four layers for content reuse. The principal objective of a prototype was to test the degree to which the CMR model could be implemented in a practical sense, and was efficient for content reuse. To achieve this, the author developed a substantial Content Management System applying the model in a Web Publishing environment.

The field trial was an initially small website which expanded rapidly with reuse being a key aspect of this expansion. This section describes design and implementation of the Inter-Publish prototype in detail and explores the outcomes in testing the CMR model in the Montage project. The prototype was branded for client purposes initially as “E-Publish”, later changed to “Inter-Publish” after formal trade marking of the software application in 2002. Issues of scalability and future development are discussed at the end of this Chapter.

As an expression (or semantic interpretation) of the theoretical model situated in a particular context, the “prototype” has particular limitations and exigencies of design determined by the platform, environment and user requirements for which it is targeted. For example, the target

runtime environment in the prototype was directed to the environments that were to be used in the field test of the prototype and familiar to the author as the developer – that is, Microsoft Windows servers. While there are limitations in the prototype, it represents a realistic exposition of the four layers of the CMR

The prototype explored specifically the content management and publishing of the following content types:

- The fragmentation of content into ECUs.
- The development of a document ontology to manage the ECUs.
- The development of a flexible workflow methodology to manage content authoring.
- The development of a variety of interfaces to support authoring and publishing, including the development of a Web server, WebDAV server, Web Services framework and FTP server to facilitate content interchange at the ECU level and higher.
- The final publishing of XML, html and text, with delivery to runtime servers using FTP and email clients developed by the author.

While the Montage project began as a small online educational portal, its scope rapidly expanded, and a secondary but important consideration in the software design was addressing the issues of security, scalability and stability. While scalability was not a principal objective in this prototyping process, the issues of scale that emerged during the project itself demanded attention as the document collection grew rapidly.

5.2.2 Architecture

5.2.2.1 Layered Architecture

While the prototype saw several generations of changes in design, the culmination of the experiment was a full four-tier prototype implementation, and the key architectural elements of the design were engineered quite early in the prototyping process. This suited both the theoretical

demands of the research and the exigencies of a field test client had immediate need for the potential offered by the software.

The approach chosen for the software development was a server-based web-enabled application using an Application Server model. The service component was developed to a Windows architectural base. A minimal amount of client-side scripting was used to enhance aesthetic and client-side functional elements of the application. At the client-side the focus was on using existing platforms and authoring tools to maximise the integrative elements of the implementation. Most key functional elements of the prototype are performed at the server-end. In a prototyping environment this has considerable advantages for the rapid development and deployment of changes in an environment that encourages quick feedback from the participants. The layered architecture is presented both through four tiers of Web Services, and through more standard methods of interfacing for purposes of client authoring at the third and fourth layers of the CMR model. The Web Services interface to the prototype provides an elaboration of the CMR model that allowed content exchange and transfer for purposes both of syndication and content exchange with other CMS.

Figure 5.1 (below) describes the software architecture of the Inter-Publish prototype in relation to the four tier CMR model. The topmost layer, being the point of mediation with public clients, has principally outbound interfaces for publishing, and inbound interfaces for external delivery of XML content - with external functional interchange through Web Services. The third layer has the richest level of public interfacing, with a rich browser-based interface supporting complex workflow interaction. The workflow business rules are governed by the primitive states of ‘draft’, ‘published’, ‘deleted’ and ‘archived’, which could be extended by the state logic specified in the ECU document type of ‘workflow’ or through the cmr_Workflow CMR schema. The third layer is also presented through WebDAV and FTP services view for interaction with existing client-based multimedia authoring tools. An FTP interface is common to many multimedia-authoring tools, and WebDAV gives a SOAP-like interface for document authoring.

WebDAV has gained considerably popularity by virtue of its implementation of "Web Folders" in the Microsoft Windows platforms. This third layer also supports a complex Web Services functional interface for interchange with other CMS and content applications. The second layer of the model is implemented within relational tables that describe the document ontology and through the associated XML schema for describing the document ontology - providing the management of the document hierarchies and the delivery of document structures through the Web Services interface.

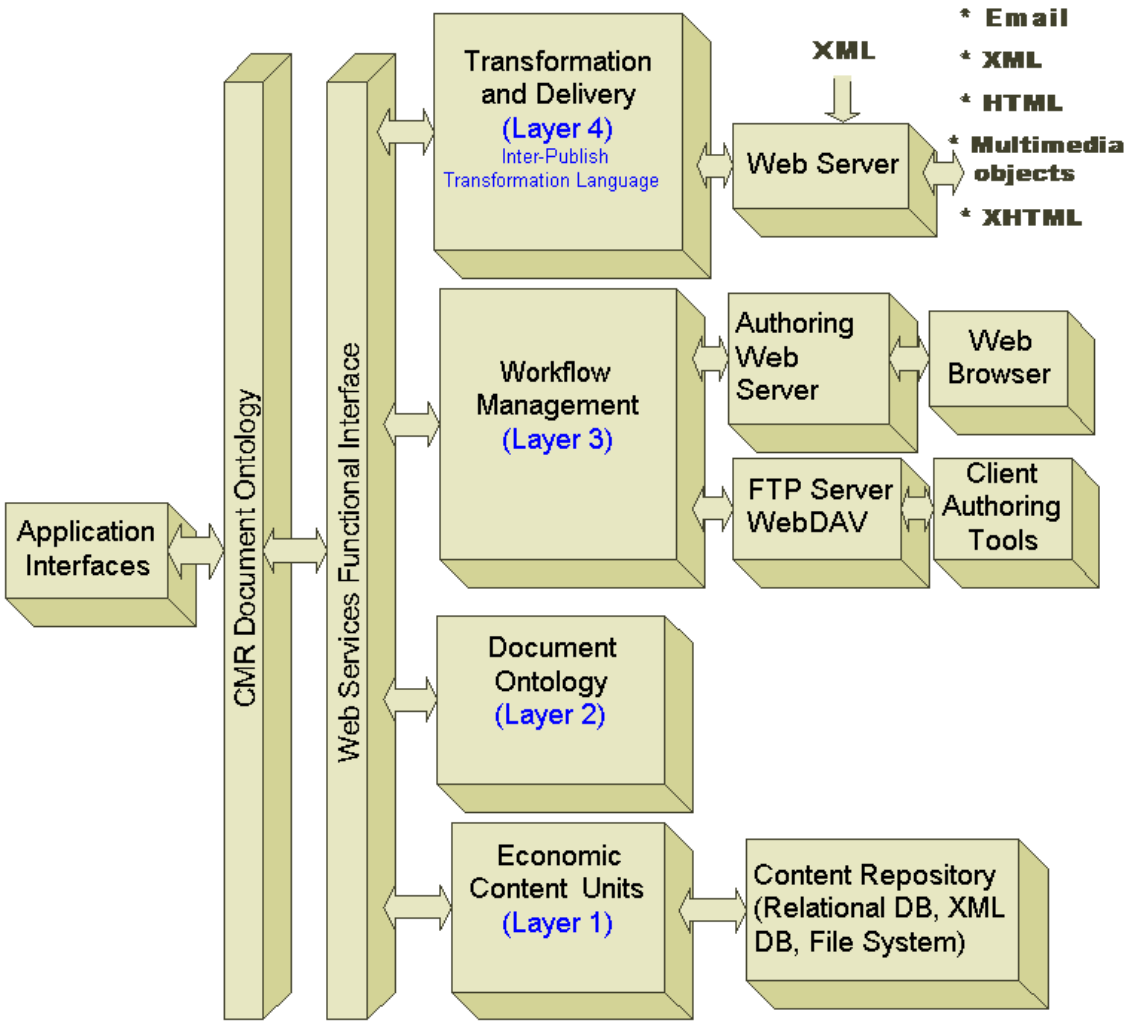


Figure 5.1: Inter-Publish Architecture

To facilitate the quick adoption of the CMS for new sites, a CMS function was provided to pre-populate the CMS document ontology with the document structures most commonly used in the Montage projects. The lowest layer, the ECU management at layer 1, is managed by the browser-based HTTP server and also by the FTP, WebDAV and Web Services interfaces mediated by the workflow security considerations imposed (by the software) at the layer 3 level of the model.

For speed of prototyping, and to capitalise on the existing skills of the author with Microsoft Visual Basic, the HTTP server was designed in VB6, with Web Services extensions implemented through VB DotNet.

The final published content was hosted on the British Council Montage service using Windows NT Advanced Server systems with Microsoft Internet Information Services, and making use of Microsoft Indexing Services. The author supplied a Windows NT server on which the Inter-Publish software was hosted, and the British Council kindly hosted (and continues to host) this server on its premises for the period of the trial.

A key element of realising the Web Services framework is realisation of the document ontology for each layer of the model describe in Chapter Four. An XML Schema describing the Web Services and the underlying document model for the CMR is presented in Appendix Two. The document model follows the schema structure shown in Figure 5.2 below.

5.2.2.2 System Architecture

The Inter-Publish server was the central integrative component of the prototype. The server was hosted on a Windows 2000, but can operate on Windows 98 or later. The server application comprises the Inter-Publish CMS Web Server (HTTP, WebDAV and Web Services) written in Microsoft Visual Basic with Visual Basic DotNET to support Web Services. The software was tested on Microsoft Access and SQL Server 2000 (MSDE)

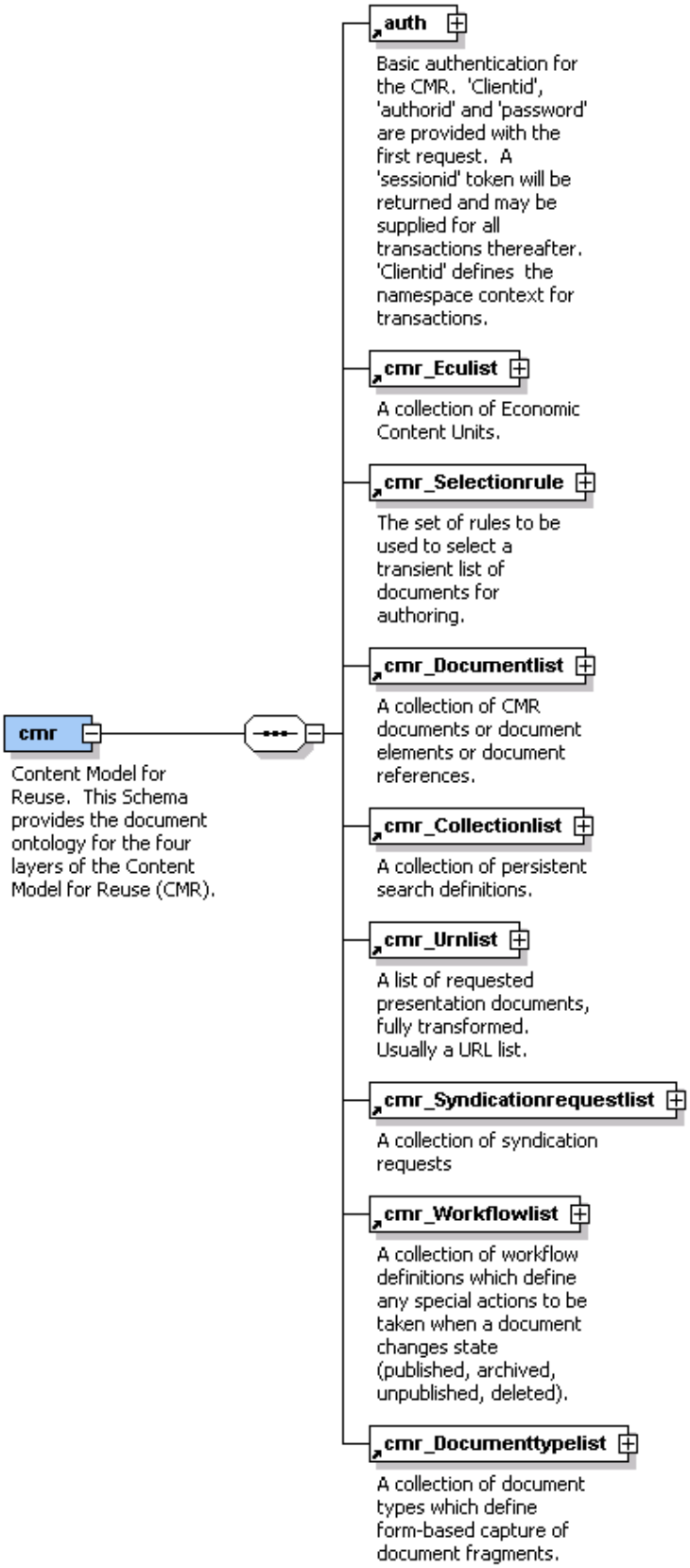


Figure 5.2: CMR Web Services XML schema

The Inter-Publish web server was written to support HTTP, HTTPS, FTP and SOAP style content interfaces. Developing a ground-up web server gave considerable flexibility in presenting a mediated view of the document and content ontology through standard content authoring interfaces. Users could log on to a traditional browser-based application to undertake text and HTML authoring, or could use FTP or WebDAV to view content in a structured manner.

5.2.2.3 Inter-Publish Services

A key element of the architecture of the prototype is the Inter-Publish web server. For most simple editing, a web based scripting solution managed on one of the existing server platforms might have been satisfactory. One of the key challenges discussed in the literature review was the management of authoring interfaces to support heterogenous media types. In order to explore the presentation of ECU-fragmented content in a virtual file system manner for existing clients interfaces such as FTP and WebDAV the author chose to develop a specialised publishing server.

Natively, both FTP and WebDAV (Web Folders) expect a native file system view of content. As part of the extension of content management to the ECU level, the author sought to provide a view of the content in its fragmented form and through the document ontology. This motivated the author's development of the Inter-Publish ftp and web server. The elegant simplicity of the HTTP protocol and the existence of relatively simple low-level TCP/IP components to work with Visual Basic made this task achievable and opened prospects for integration of emergent technology for Web Services without the constrains of the IIS framework – and in particular in a way that allowed virtual browsing of the content.

In developing the initial web server the author made reference principally to RFC2616 for the HTTP and HTTPS protocols. Particularly helpful were Gettys et al (1999) and Franks et al. (1999). Content interchange of files through the browser interface required implementation of RFC 1865 (Nebel & Masinter, 1995), and the HTTPS interface required extensions discussed in RFC 2617 with authentication (Franks et al., 1999).

Having implemented a stable web server, the extension of this framework to support SOAP-like services such as WEBDAV and SOAP-based handling of Web Services was possible. The WebDAV interface was developed with reference to RFC2518, particularly using Goland, Whitehead, Faizi, Carter, & Jensen (1999). The FTP interface was developed against RFC 0542 (Neigus, 1973).

The "Content Type" of RFC 0822 (Crocker, 1982) and its successor RFC1341 (Borenstein & Freed, 1992) is implemented at the ECU level of Inter-Publish, as each ECU can potentially be a different content type. The "Content Type" of Inter-Publish is a higher-level organisational unit bringing documents of a common purpose together.

The external client websites published using Inter-Publish were hosted on a relatively standard Microsoft Windows 2000 web server supporting Microsoft Indexing Service and O'Reilly's Web Board conferencing software. The server was tested on most currently supported Windows platforms (Windows 95, NT4, ME, 2000, XP).

Where relevant components are readily accessible and suitable for the architectural base, these have been preferred over bespoke development both to accelerate the development of an advanced implementation model and to test the integrative elements of the software at different layers of the model. Low-level Internet interfacing was facilitated by the IP*Works TCP/IP components published by *nsoftware* (www.nsoftware.com). The following packet-level controls were used from the IP*Works toolkit:

- *IPDaemon* - which his a generalised IP listening service which can be used to interface with relevant IP services (such as HTTP)
- *IPPort* - which was used for lower-level IP communication at the port level, as an interface to the Windows “winsock” networking layer.

PDF and image thumbnailing were achieved through the excellent component provided by WebSuperGoo.

Future extensions could include support for architectures such as native interpretation of RDF feeds (Lassila & Swick, 1999), but were not required for purposes of the British Council experiment. A syndication framework supporting RDF-style syndication is presented through the Layer 4 Web Services interface.

Four application servers were implemented for the prototype:

1. An HTTP server to support content authoring and workflow. A browser-based front end is used for modelling of website architectures and content structures. The content editing interface presents a site-hierarchical view of content components and provides a generalised interface to editing content models. It is also a component-oriented publishing interface for the content. A Microsoft ActiveX component is used for HTML, XHTML and XML editing of content presented at this layer. The HTTP server provides a structured user interface for content management and workflow editorial control (layer 3) and a publishing interface servicing the needs of layer 4 (content distribution).
2. A Web Services provider. The DotNet version of Visual Basic (VB.Net) provides extensive support for SOAP-based delivery via Web Services. While services are exposed to all layers of the model, only layer 3 and layer 4 interfaces are directed to public authoring and access.
3. An FTP server. Layer three content management interfaces are delivered through FTP for client integration with the many multimedia tools that use FTP as their principal interfacing vehicle.
4. A WebDAV server. Layer three content management interfaces supporting WEBDAV virtual folder browsing of the content, providing another form of client integration with heterogeneous editing tools (Platform independent).

5.2.3 Architectural Limitations

While the client-side interaction was generalised as far as possible for a reasonable level of platform independence, some site modelling components are Internet Explorer specific. The integrated HTML editor requires a DHTML-based ActiveX control – and is, therefore, in most respects Windows and Internet Explorer specific.

In other respects, the content integration through the HTTP, FTP and WebDAV interfaces was platform independent. Indeed, the focus of the content-editing elements of the prototype has been on integrating with heterogeneous multimedia tools. The client environment in which the prototype was tested comprised Microsoft Windows OS client PC's running Internet Explorer 4.0 or later and Windows-based authoring tools such as Macromedia Dreamweaver, Macromedia Homesite and various image editing tools.

5.2.4 Focus of the Field Implementation

The British Council Montage project provided a rich environment to explore content regeneration and reuse. Their plans for the Montage website included management of some multilingual sites as well as the reuse and replication of content rebranded for different countries. In particular, the Montage project offered the opportunity to test aspects of content regeneration and reuse at the different layers of the model. This evaluation is discussed further in Chapter Seven. Initially, the Montage field implementation offered at least the following opportunities to explore different aspects of the model:

Layer 4: Transformation and Delivery

- Test content reuse in a challenging Internet publishing environment – specifically the primary generation of content and its later reuse through regeneration in web, PDF, and other web-based content forms, and through syndication activities.
- Test content reuse in alternative runtime delivery environment: including to a discussion list (O'Reilly's *Webboard)* and through email delivery of content.
- Test content delivery through XSL based on XML source data and through direct XML delivery of content.

Layer 3: Workflow Management

- Test integrative elements of the model through search features and content structural presentation.
- Provide a model for persistent (rule based) document search profiles to facilitate content reuse through indexing and transformation of content.

- Design of a flexible, extensible, workflow environment.
- Demonstrate component-oriented integration concepts with other authoring tools and content sources.

Layer 2: Document Ontology

- Demonstrate the utility of a flexible document ontology.
- Demonstrate content integration with content resources managed integrally within the CMS and with externally housed multimedia resources.
- Demonstrate the architectural utility for interfacing to other relational database objects (that is, in a clearinghouse model of content access).

Layer 1: Economic Content Unit

- Provide an integrated environment of HTML, XML, text and multimedia content capture, fragmenting the content at a level of utility for reuse suiting the target Montage environment.
- Demonstrate the management of content across language boundaries and for across different encoding formats.

5.3 Functional Decomposition for Reuse

The industry survey in Chapter Two examined representative CMS products out of more than two hundred industry solutions currently available. Many of these solutions remain directed principally to primary generation of website content, and many recent entrants continue to use their own proprietary template-based publishing. Systems with full content life-cycle functionality come at no small cost, with a full implementation costing in excess of $AUS500,000. The strengths of the larger systems lie especially in their enterprise workflow and document management capability and the availability of add-ons to achieve many functional elements of Content Management. While CMS products are also emerging in the Open Source arena, many have functionality directed to their particular "client" communities.

The initial deconstruction of functions relating to the CMS products reviewed in Chapter Two resulted in 190 functions broadly categorised in Appendix A under the headings "Content

Management", "Administration", "Versioning" and "Authoring Integration". These detailed functional elements were subsequently analysed in the context of the CMR model. The original functional decomposition is presented in Appendix A. This section presents a high-level functional decomposition focussing on those that pertain to content reuse, and characterised against the CMR model. This functional decomposition was an element in the functional design of each of the layers of Inter-Publish.

5.3.1 Layer 1 Functional Decomposition – Economic Content Units

The principles for content reuse discussed in Chapter Two have functional implications at each layer of the model. The indicators presented in Table 5.1 are examples of those functions that might demonstrate realisation of these content reuse principles.

An ECU is a single fragment of content that might have encoding or language variants but which is semantically identical across these variants. That most CMS suppliers at the top end had internationalised their interfaces and could encode content in different language forms does not necessarily extend to effective cross-language management of the content. It was clear from the industry survey of CMS products (see the literature review - Chapter Two) that Internationalisation of CMS products was well advanced in the commercial arena. Some Open Source solutions were limited in their Unicode support where they relied on the MySQL relational database.

The boundary between smaller web editing tools (such as Dreamweaver) and CMS products is the database treatment of multimedia objects. Dreamweaver, and other web editing tools, continue to treat the web server as the principal content repository and runtime engine. This allows the deployment of complementary authoring workflow technologies such as WebDAV. However, all systems reviewed had integrated text and image management in their content repository. A few of the CMS examined in the literature review continue to place video and audio resources in the runtime delivery system with metadata references to the resource. While there are

advantages of simplicity in the dual role for the web server both as a runtime content distribution engine and as a content repository, there are also difficulties. Version control and distributed authoring are particularly difficult in such a context. Furthermore, where the document repository is also the runtime engine, the extent of content reuse may well be constrained by the technological architecture of the runtime engine.

Function	Indicator	Industry status
Content reuse requires the fragmentation of content in a manner that allows the separation of content from its final presentational form	A flexible document ontology that allows separation of content elements.	Content separation from structure and style is a "sine qua non" of the CMS industry. All products at least claimed this capability, although the flexibility of the ontology varied.
Support for content in many languages	Unicode support, Double byte character support.	Several CMS products were lacking in Unicode support.
Internationalisation of user interface	International product use outside the originating country	All category 1 products had strong internationalisation, characteristic of their strong supplier base. Mid-range products (such as RedDot) and some open source (especially Tiki) show good internationalisation.
Multi-lingual content variants	Translation support or presentation of semantically identical content on a single page with multiple language variants	Multi-lingual support varied dramatically across products. Some CMS products even in category 2 supplied basic translation services integrated in their product.
Temporal Versioning	The ability to version content over time, and recover/retrieve/review variant versions	Most CMS products featured at least whole-document version recovery.
Encoding variants	The ability to encode some objects at different levels of content granularity.	This aspect appears to be limited in current CMS products.
Content discovery, and semantic information that facilitates this discovery, is an essential element of content reuse.	Ability to define metadata content with controlled vocabularies and supporting different ontologies	The flexibility of metadata management appears limited in current CMS products, particularly in automatic derivation of metadata from content.

Table 5.1: Functional mapping to Layer 1 of the CMR

All of the Category 1 CMS described in Chapter Two maintained a content repository separate from the runtime engine, often in a relational database (some with an option of an XML repository).

Most efforts at content versioning were directed at text content management. Some systems, such as Vignette, appear to take a toolkit approach to versioning. The product Documentum showed strengths at all levels of versioning, reflecting its document management roots. Few supported dynamic image handling or versioning of images across different formats and resolutions.

Relational database engines were the common database platform across all products, some through storage of XML objects within the relational table, others through translation to an intermediate or final XML format for publishing. Insufficient technical information was available to determine the extent to which any of the products generalised this framework to support integration to any SQL schema, although two Category 3 products appear to be modelled around such an approach. Few products supported arbitrary mapping to different database schema and content sources. This is an area where Web services may provide an effective means for content interchange between CMS products as large organisations address issues of information interchange between large CMS solutions.

5.3.2 Layer 2 Functional Decomposition – Document Ontology

Table 5.2 presents a functional mapping at layer 2 of the CMR. All CMS packages discussed in the literature review exhibit some form of separation of content from presentation. This separation is fundamental to gaining a utility value from separating the content that has persistent value over time (the essence) and content relating to aesthetic presentation and navigation (the templates). Content separation from style and structure is achieved either through an application or media specific encoding and markup. Regeneration of content is achieved with either script-based approaches (such as XSL) or template-based transformation systems. The consistent capture of content for effective reuse also depends on the effectiveness of the ontological framework for specifying document structures and controlled vocabularies. The value of a template-based architecture is evident in its use ubiquitous use in commercial CMS.

Function	Indicator	Industry Status
Template-based or transformation structures to yield published content from content fragments	Forms-based or XML-based content separation. Fragmentation or "chunking" of content.	Most CMS products use a template-based approach, some with proprietary markup and scripting for their template-based otology. Some also support XSLT transformation.
A flexible ontology that supports the definition of document types amenable to simple change and expressing complex content elements (Economic Content Units)	The ability to create on-the fly document structures without database changes	The flexibility of the document ontology, or at least the complexity of creating new definitions (the rigidity of the ontology) varied considerably.
Ontology versioning	Where the document object model has changed over time, historical content can be viewed in the context of the original or current document ontology.	Limited in most systems reviewed.
Content can be used outside the particular web page context to which it is published	Content reuse in different forms – for example using Syndication	Available, often as a chargeable add-on to solutions – this was the case even in some Open Source solutions
The ability to create content collections for use by reporting, indexing and syndication engines for transformation and distribution of content	A report writer, transformation service and content generator that can support all media formats. Support for content definition based on existing XML/DTD definitions Ability to import XML content	Flexible report writers operating at a fine granularity appear limited across current CMS products.
Support commonly used document structures, and validation against those structures	DTD validation SMIL multimedia document generation	There is variable support of common document ontologies. Some offer additional plug-ins to support well-known ontologies such as RSS.
Document ontology can itself be documented or is self documenting	The ability to version content over time, and recover/retrieve/review variant versions	Limited in most systems reviewed.

Table 5.2: Functional mapping to Layer 2 of the CMR

Approaches to content storage have been discussed in some detail in Chapter Two. Due to the ubiquitous nature of relational database systems, most CMS solutions attempt to accommodate content storage within existing corporate relational databases. There are, however, emergent approaches using XML database systems for purposes of content storage. A hierarchical database model might be more suitable to the unstructured content resources mapped to hierarchical document structures. CMS vendors naturally direct their product development to architectures acceptable and familiar to their client community, commonly offering Oracle, Microsoft SQL Server as the relational database server with commercial products, and in open source community generally offering MySQL.

Function	Indicator	Industry Status
Content editors can be defined with different levels of content visibility	User- based, group-based or LDAP-style control on access to content resources.	Category 1 suppliers generally provide both LDAP and group-based user access. LDAP support irregular among smaller CMS vendors and open source solutions.
Editorial constraints on modification to templates	Content access can be constrained to particular types of content or content domains.	All CMS products differentiated authors from designers in some manner – some with group-based privileges.
Multi-lingual publishing	The ability to publish language variants of pages in a language-based template driven manner.	Multi-lingual support is uneven across the products examined.
Content can be discovered through transient and persistent collections.	Support for ad-hoc Boolean-style queries across the content, and the ability to persist these queries over time.	While many systems have some form of indexing engine or report writer, use of persistent collections as a consistent organisational method across workflow activities was not evident.
Temporal Versioning	The ability to version documents over time, and recover/retrieve/review variant versions.	Available in most industry offerings.
Availability of graphic tools	Visio-style modelling capability in order to design the workflow process.	Category 1 products generally boasted some modelling capability.
Routing capability and the establishment of workflow groups	Visio-style modelling capability in order to The ability for the end user to direct the content workflow to particular path(s). The extent to which users can participate individually or in groups.	Category 1 products generally boasted complex routing capability, although complexity in configuration of workflow routing was in many cases an issue. Email-based alerts featured in those CMS products with an established workflow engine.
Performance metrics and Queue Management	The tracking of the status of items at particular workflow stages and metrics on processing performance at each stage and the ability to review and track documents and adjust their priority.	Generally available only in the category 1 products, and even there limited.
Management of events	The control of conditions at which workflow events will happen.	Some category 1 products had complex event management capability (especially Documentum). Some Open Source products had extensions to integrate with other Open Source workflow solutions.
Authoring Workload and Monitoring	The channelling of documents based on workload of document authors and editors and the tracking and alerting based on load conditions.	Generally available only in the category 1 products, and even there limited.
Simulation	The ability to simulate under load conditions the workflow model.	Not common across the CMS product class.
Handling of forms and Webflow	The integration of models for form-based actions. The integration of web-based tools (including email) in the workflow process	Generally available in most CMS products, but great variability in flexibility and complexity of form definitions. Some CMS products had fairly rigid document structures after initial establishment. Webflow is the universal paradigm for forms management.
Documentation	The completeness of documentation in management and use of workflows	Generally strongest in the category 1 products, and weakest in category 3 systems

Table 5.3: Functional mapping to Layer 3 of the CMR

5.3.3 Layer 3 Functional Decomposition – Workflow management

The functional elements of Content Management Systems are at their most diverse at the point of interface with the content author. The Category 1 proprietary systems discussed in the literature review offered the richest workflow features, although many provided a collection of tools rather than a single solution, and required tailoring to achieve site-specific implementation. The functions described in Table 5.3 below include the elements of workflow that are either integral to effective authoring (and therefore axiomatic to content reuse) or which extend the workflow to enhance systematic content regeneration and reuse.

Function	Indicator	Industry Status
Primary generation of web pages	Can publish web content in HTML, XML and other formats, dynamically or on-demand	All suppliers can meet this requirement in one form or another. Template rather than XSL transformation, often through supplier-oriented template-based methods remains the most popular means of content transformation.
Syndication to varying document ontologies	Support for NewsML, RDF, RSS, PRISM, SCORM, SMIL	Major suppliers have developed plug-ins for specific syndication ontologies
Ability to support alternative transformation approaches	Support for XSLT or template-based transformation techniques and the ability to manage tasks for multi-stage transformation. Support for multi-stage transformation where elements of the transformation occur in other systems.	Server-based XSLT transformation was available for a few products, but did not appear to be the preferred transformational approach. Multi-stage transformation was not evident among the product sets reviewed.
Website versioning	Can create editions of a website, and review, roll forward, roll back these editions.	This is a function that challenges all CMS suppliers. Only two suppliers (at least nominally) supported the rolling forward and backward at the whole edition level.
Flexible delivery agency	A flexible delivery agent with capability for retry and a scheduling agency for time delivery	There were considerable differences in delivery agency among suppliers. Server-centric suppliers often had integrated, dynamic generation of content as part of their design. Major product suppliers could offer a spectrum of delivery agency.
Personalisation	A means of delivering content based on the known demographics and information requirements of the target audience.	This is a prized feature of the major CMS suppliers, and can be implemented as an independent layer of many Web Servers. A number of Open Source personalisation "plug-ins" have also emerged.

Table 5.4: Functional mapping to Layer 4 of the CMR

5.3.4 Layer 4 Functional Decomposition - Transformation and Delivery

Three approaches were apparent in the Industry Review: internal template-based approaches to content transformation, XSL transformation producing static content and "on-the-fly" XSL transformation. Syndication, one of the clearest demonstrable products of content reuse, remains a weakness across many of the CMS products in the market. Table 5.4 presents the indicators of content reuse.

5.3.5 Data Flow

Chapter Two explored the contingent nature of the workflow process in web publishing. As a result, there is no "generic" dataflow model for web-based publishing. Figure 5.3 illustrates the publishing workflow process. In this example workflow functions are mapped against the model in web publishing framework. The example illustrates a workflow in which image content is managed and delivered directly from the web server (which acts as both a content repository and web server) while text content is managed in a central CMS repository.

This workflow illustration shows two content sources: automatic external import of content from XML sources and authoring-based capture of content through the workflow system. Text content is retained in a central repository and linked to image resources retained only on the web server. Publishing involves the assembly and transformation of text resources for delivery to the web server runtime engine. This workflow model is characteristic of many CMS products that operate at arms-length from the web server, and as distinct from those which have an integrated web server and dynamically generate all content. Some CMS products have integrated a more complete life-cycle approach to content, integrating at least image content with text content.

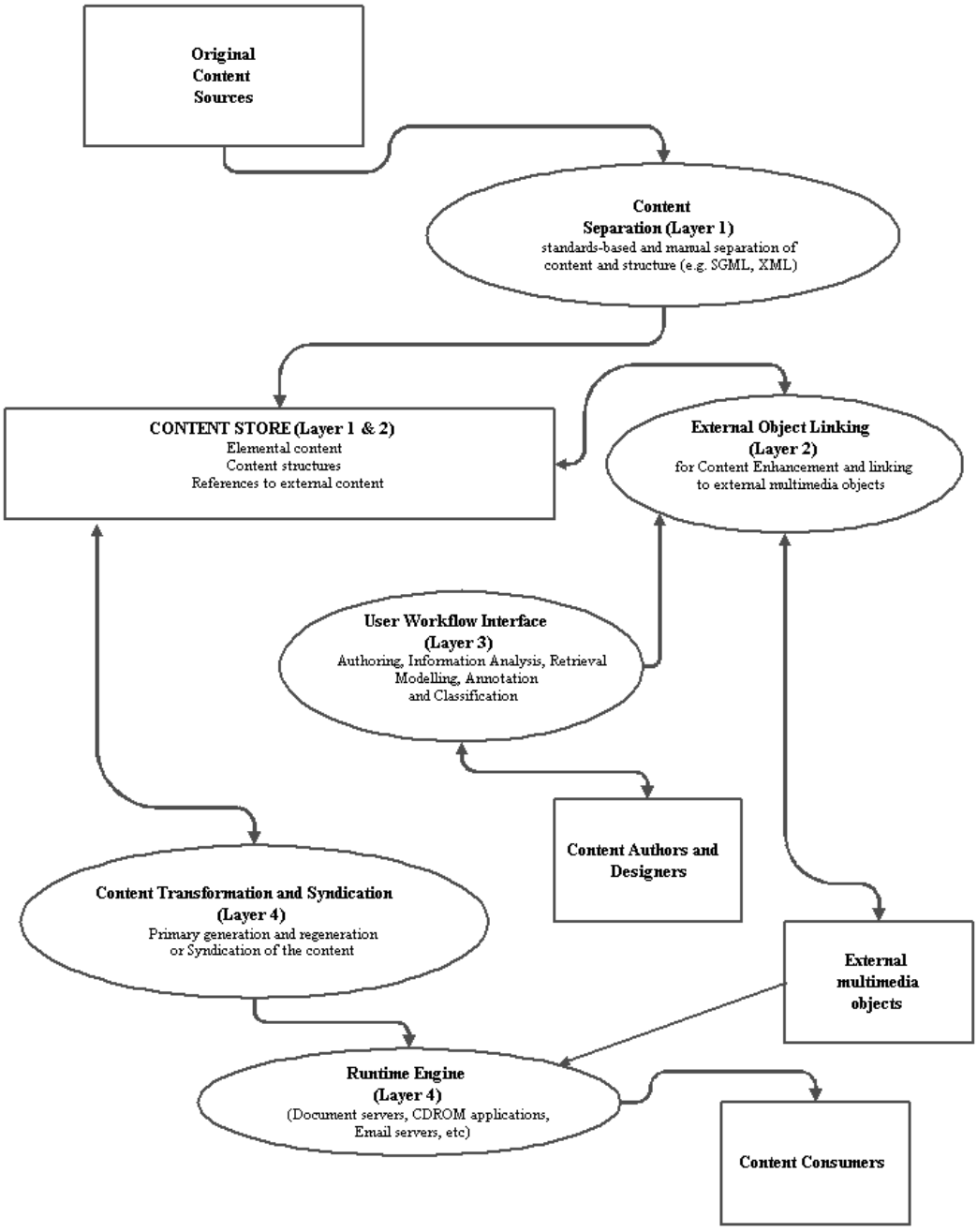


Figure 5.3: Illustrative data flow model for multimedia publishing

5.4 Inter-Publish Database design

5.4.1 Background to the design of the content repository

The first version of the prototype was exploratory in nature, experimenting with the user interface design for web publishing support and the workflow issues surrounding publishing, rather than issues of content reuse and the underlying ECU model (which was still in conceptual

formulation). An Active Server Pages (ASP) user interface to a simple relational database was designed (see Figure 5.4 below). A Microsoft Access database was used for content storage with a hierarchical model for content storage of simple HTML pages with associated images and links, which achieved basic separation of the HTML content from its presentational layouts. The schema was designed for web page management with a coarse granularity of the content units (a web page comprising a template surrounding one or two content elements). The software identified content editors by the Department and Section, organising the content into this simple content hierarchy. Content was stored in a single table "Documents", with a limited number of (HTML only) content elements, and a simple persistent indexing methodology for presentation of structured content indexes (saving persistent searches in the *CollectionDocuments* table). One table row in the *Documents* table represents the content and meta-information for one published web page. The *References* table captures ancillary information relating to an HTML document (attached images, references, etc). This design was suitable for a simple document architecture directed to very simple content separation and a coarse granularity of fragmentation (one text block per page) with its associated publishing Layout. The architecture supported a fixed set of meta-content elements predefined in the table architecture.

Many CMS products map their document model directly to a table schema and there are several advantages to a simple document schema. One advantage is the facility with which existing relational query tools can be deployed against the database architecture. Another advantage is the existence of an established component set for mapping XML to relational tables. Such approaches may limit the flexibility of the ontology as the document collection becomes large. In 1996 this author implemented such a CMS model in the Australian Broadcasting Corporation (ABC), in a framework that by 2002 had reached 836209 documents. Rather than expand the table indefinitely as new content elements were required, some of the table columns are now used in an "abstract" manner, with their meaning defined in a separate content definition table. A single table row in the ABC architecture continues to define all content elements of a

"news" item in the CMS. When a complex content definition requires an additional column, it is necessary to add another row to the main content table – the "Stories" table. Over time this has led to a very wide table structure more like a sparse matrix than a relational table. The principle difficulty with this design approach was the lack of ontological flexibility in describing different document structures.

A similar approach is to map the document ontology to a "class" which is realised as a table schema and specialised for particular purposes (Fraser, 2002). Such a mapping tends to make the document structure relatively inflexible after initial definition. Semi-structured document forms are not well suited to such an approach.

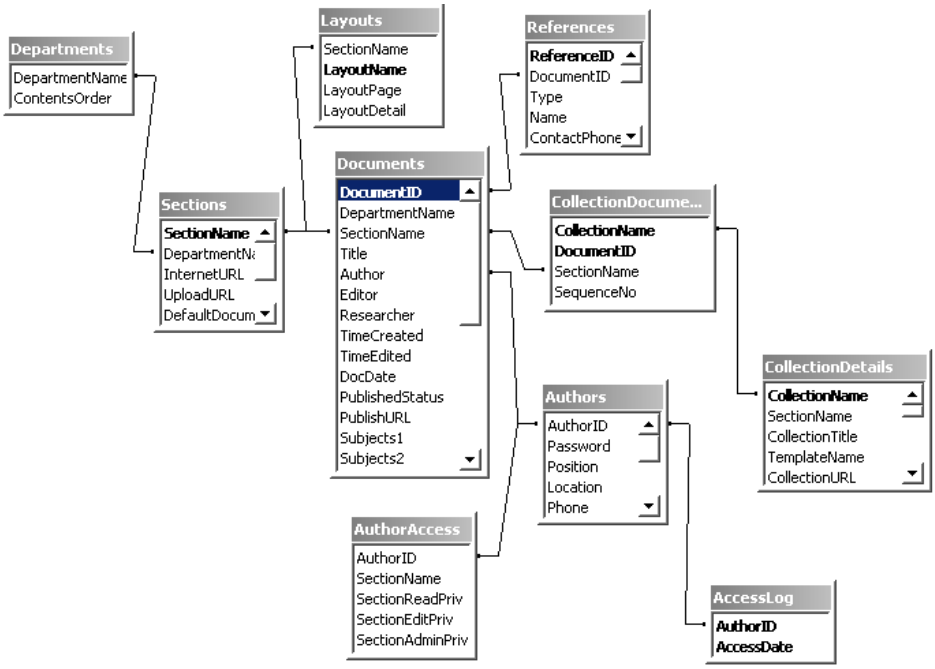


Figure 5.4: First prototype relational model

The first (throwaway) prototype was useful in eliciting the workflow requirements required for web publishing in the target environment. However, the author did not think that complex document hierarchies, representing many content types could be adequately represented in such a static hierarchical data model. Where relational schemas have been used, specialised web blades have been required to mitigate the problems presented by the relational structure in management

of complex multimedia documents (Boll & Klas, 2001). Even in a web-publishing environment the document elements could be considerably more complex than described by such an approach.

The conceptual objective of the database schema was to realise an instance of the 1st layer of the CMS model in a manner that realised the multi-lingual and multi-format content issue. Documents should support a range of language instances, multiple media types and multiple presentational instances.

The second relational database schema was designed to support capture of heterogeneous text and multimedia content types of unpredictable hierarchical structure and content elements. These requirements mandated a more complex relational model. The objective of the revised schema was specifically to support the layer one elements of the CMR. While XML databases are emergent in the industry in recent years, the target client infrastructure centred on a relational database infrastructure – either Microsoft Access or Microsoft SQL Server. The approach to modelling the content at this stage of the project was to realise this content in a hybrid relational model that would allow the realisation of the required flexible hierarchical complexity of content objects in a framework amenable to implementation in a relational structure. If the relational model is not ideal for a loosely defined hierarchical mapping of information, such weaknesses can be mitigated through their support the Binary Large Object (BLOB) of effectively unlimited size. This second-stage relational structure was designed to maximise the flexibility of the document object model while retaining the benefits and ease of implementation of relational database. The Economic Content Unit was stored in a BLOB with sufficient metadata to describe the encoded content along with a supporting document ontology to allow complex relationships between ECU objects.

The Inter-Publish prototype released in December 2000 contained the following relational tables:

Authors - a table with metadata describing the roles and capabilities of a given author

AuthorSections - section specific access limitations for a given author. Authors could be allocated a role of "read", "edit" and "design" access to a given section.

Clients - a namespace identifier for a given collection of content. A client namespace defines the entire universe of content and document ontology available for a particular publishing community.

Divisions - the highest level of content organisation for a "client" - for example, all content for a given website

Documents - a table describing the core rights and workflow status of a Document, and containing search metadata

DocumentContents - a table of Economic Content Units

Attributes - a table providing the attributes of elements of a given Document Type. An element could describe either a reference structure to another document structure or an ECU element, its authoring interface definitions and encoding.

DocumentTypes - a table describing acyclic hierarchical document definitions

In addition to these core content tables, the following tables were required for workflow and authoring support:

AuthorSessions - a table containing the current presentation preferences for a given author.

WorkflowSequence - a table managing the current state in a multi-screen user interaction or theme.

WordFrequency - a table containing stemmed values for each word (excluding stop words, markup and scripts) from documents, and used for automatic document similarity and semantic key phrase analysis.

5.4.2 Layer 1 Database implementation - the ECU

Chapter Four describes Web Services views of each of the layers of the model. The base layer of the model, the ECU, is central to the content reuse strategy described in this thesis. A strictly relational model for describing content in this manner was not satisfactory to achieve the

flexibility in document hierarchy or content structure. The ECU provides a universal view of all content objects accessible across the content space. The task of metadata description of multimedia objects of other sorts does not have quite the same clarity of direction. Literature in multimedia indexing, metadata description and storage experiments abound, but a clear direction on multimedia is at present defined more by market directions than theoretical conventions. Emergent XML databases may well provide a suitable means for flexible multimedia document management, but were still nascent at the time this thesis was commenced. The relational schema was developed to support a flexible ECU content management framework within a standard relational structure supporting multiple language and encoding versions (Figure 5.5 below). The ECU encapsulated language and encoding variants of content, but also allowed encapsulation of semantic metadata, rights metadata and workflow status metadata. The ECU was expressed in Inter-Publish as a row in a relational table. Language and encoding instances were realised in a single Binary Large Object field, while some metadata tracking information was realised as columns in the relational table for efficiency in retrieval. An ECU could participate in an unlimited number of documents but must (for management purposes) belong to at least one.

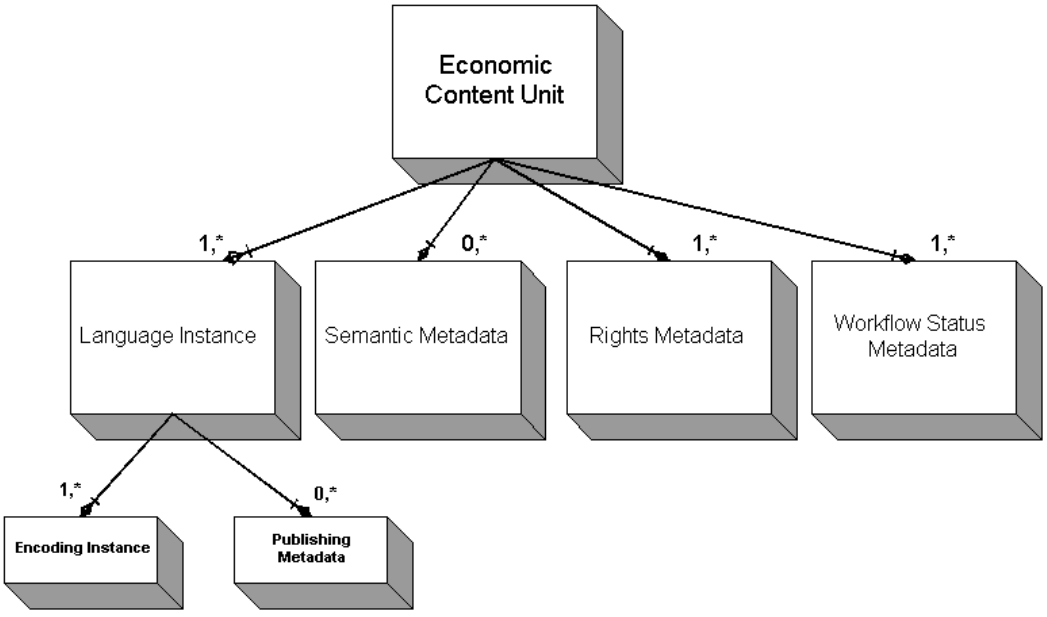


Figure 5.5: Structure of the ECU

The ECU was realised in the *DocumentContents* table of Inter-Publish. While most semantic metadata is contained with the ECU fragment, some metadata is realised as relational columns to facilitate workflow management of the documents as follows:

Namespace context of content resources: The column *ClientID* is the namespace for the overall content database. Different content collections could be bought together under different *ClientID* values. The column *ContentID* provides unique identification for the ECU.

Rights data: The ECU belonged at a minimum to a parent document. Access to this ECU could be defined by a matrix of SectionName and DocumentType. Important columns in this table are *DocumentID* (the parent document identifier for one or more ECU objects), *SectionName (*the principle workflow organising category in Inter-Publish) and *DocumentType (*being a workflow-management document definition at the CMR layer 2).

Semantic metadata: Each Metadata item associated with an ECU, including the publishing usage of the ECU, was itself saved as ECU content within the *DocumentContents* table bought together by the parent *DocumentID* and the unique content identifier *ContentID***.**

Workflow status: Without further modelling, a simple workflow process is implicit in Inter-Publish. A document has at all times a status of "Draft", "Published", "Archived" or "Deleted". This simple workflow model is extended to a more complex workflow system using document type transition as its key event paradigm. This workflow management system is described in more detail below, and is managed using the column *DocumentStatus*

Language instance: The ECU is encoded in the relational table in the column *DocumentText* – a content BLOB containing *either* the content itself (for a trivial ECU with no variants, *or* where variants exist the full XML description of the ECU:

```
<languagevariant language="en">
    <encodingvariant encodingdescription="html"
        element="ipelementname">
            <!#DATA >
        </encodingvariant>
</languagevariant>
```

For workflow management and parent identification, the following columns were mandatory for each ECU:

ClientID – the namespace context for the content.

DocumentID – bringing content together as a layer 2 document.

ContentID – the unique identifier for an ECU.

SectionName – the parent section name.

DocumentType – the unique descriptor for the structure of a particular document tree.

ContentType – an aggregation of Document Types into generic classes of document (for example Word, HTML, PDF might be aggregated to a Content Type of "Document")

The metadata pertaining to the encoding and use of a particular ECU element is described in the relational table *Elements* containing information such as:

ElementName – a binding element used to reference this element structure within document objects.

ElementDescription – a descriptive label for the element giving more details on its usage.

EncodingType – the default encoding method for the content – overridden by content variants of language or form. – for example XHTML, XML, HTML, and plain text.

FieldSelections – a multi-value for fixed controlled vocabularies (for example drop-down lists of fixed values). This is as distinct from a controlled vocabulary instantiated as another content form within Inter-Publish.

ElementArray – an XML description of other elements that belong to this element. This XML description is used to present an array structure of elements to the end users.

The database schema implemented to support this structure is described in Figure 5.6 below.

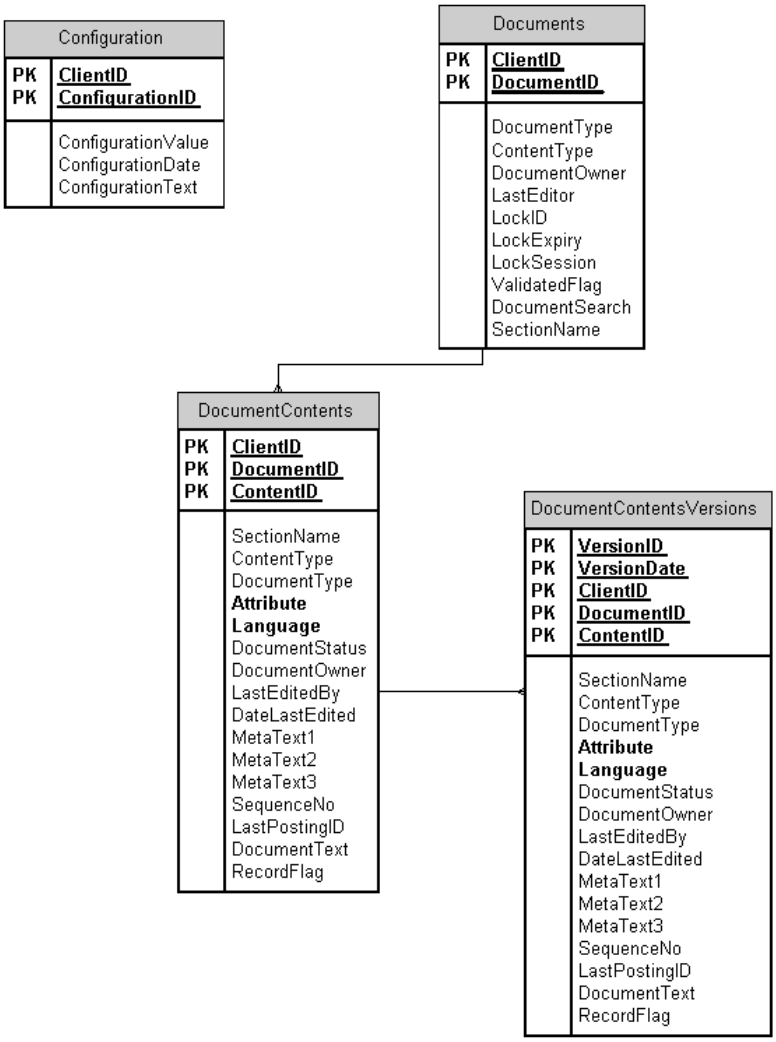


Figure 5.6: Economic Content Unit schema

Every ECU fragment could participate in one or more documents. It was essential, for orphan management, however, that an ECU belong to at least one parent document – as the point of visibility for purposes of workflow visibility and control. The parent section to which the content belonged also determined editorial access to the content.

The following are the significant principles associated with this layer that are realised in the prototype implemented as part of this thesis:

1. Each content unit has associated with it a definition which includes its mime-type handling, as well as other associate metadata regarding its content handling rules – in particular its presentational format, its delivery handling for purposes of multimedia editing, and its language – the *Element***.**
2. Each content unit can have attributed to it associated metadata descriptive information based on free text or controlled vocabulary

The base layer of the model, the ECU, is central to the content reuse strategy. The ECU provides a universal view of all content objects accessible across the content space. The last decade has seen considerable advances in some methods for content encoding. The task of encoding text content has been considerably advanced with the evolution of Unicode. The structural description of text content has also been advanced with the advent of XML. The literature review illustrated the multitude of document and metadata ontologies, and in this context, the ECU layer and the document ontology layer must be flexible to allow transformation of content to multiple document ontologies.

Content versioning was supported through use of triggers updating mirror tables (in Inter-Publish the versioning table *DocumentContentsVersions*), allowing individual ECU rollback.

5.4.3 Layer 2 Database Implementation - the Document Ontology

Central to the model being explored in this thesis is a flexible document ontology allowing complex unstructured document forms to be created (by the end user) in a manner that is readily accommodated by the structure and interfaces of the system. The ECU provides the primitive serialised layer of content. This content layer has no intrinsic meaning until it is situated in one or more document structures. These document structures might vary in web publishing from a

simple HTML document object model to a more complex aggregation of multimedia objects represented by SMIL document. In this prototype the *DocumentTypes* table is used to describe complex document hierarchies. The *DocumentTypes* table has a simple relational structure encoding an XML description of the document ontology. A ClientID namespace identifier qualifies the table. The *ContentType* is an aggregation of Document Types into generic classes of document (for example Word, HTML, PDF might be aggregated to a Content Type of "Document"). The *DocumentElements* column defines the document structure using an XML description in the form:

```
<ElementArray>
    <ElementName />
    <ElementDescription />
</ElementArray>
```

A document element could be one of three types:

Simple - this element describes a discrete ECU

Compound - this element brings together a group of ECU's in an "array" of semantically related ECU's. The compound element type was useful in describing well-structured content forms.

Linking - defining referential relationship to another document type - the application enforcing an acyclic linkage.

5.4.4 Layer 3 Database implementation - the Workflow process

The workflow layer represents the most complex functional element of the CMS, managing as it does the complex interface with content authors and their heterogeneity of authoring systems. The database model used for Inter-Publish was designed to meet at a minimum the workflow requirements of the British Council in their Montage project, but to be implemented in a manner that allows reasonable extensibility to more complex workflow

processes. This proved beneficial when special workflows were required for competitions and other multi-level approval processes eventuated.

The database diagram presented in Figure 5.7 below extends the basic content ontological model to provide contextual information about the authors and their access. The *Clients* table provides namespace relativity for content authors and ontological definitions. The *Authors* table, taken in conjunction with the *AuthorAccess* table, defines a section-based content segmentation. Group-based and LDAP-based author access privileges were not implemented but could be accommodated in this framework. The system was designed to eight levels of access, with the content 'Section' being the principal granularity of content access:

- Global administrators – having a content view across all name spaces.
- System administrators – having a content view across all content for a given C*lientID* namespace.
- Site administrators – having a full content view for nominated sections.
- Content designers – having a full content view of both content and design layouts for nominated sections.
- Content Editors – having a full content view of informational content but not design layouts for nominated sections.
- Limited view authors – having a view limited to their own documents only or to particular document types only.
- The *AuthorSessions* table defines the current state of authors, their preferences in the browser-based interface and state management information. The *WorkflowSequence* table is for transient stacking movement history for multi-paged editing sequences. The collections, collection criteria and collection documents tables provide the framework for the definition of static, saved, search definitions. These collections are accessible for use in all aspects of content authoring, global content management, searching, indexing and publishing.
- In addition to its role as a content and design layout repository, the *DocumentContents* table is used as a framework for multi-lingual user-interface design. The document type "Word" is used to provide user-configurable context-sensitive and internationalized browser presentation of content.

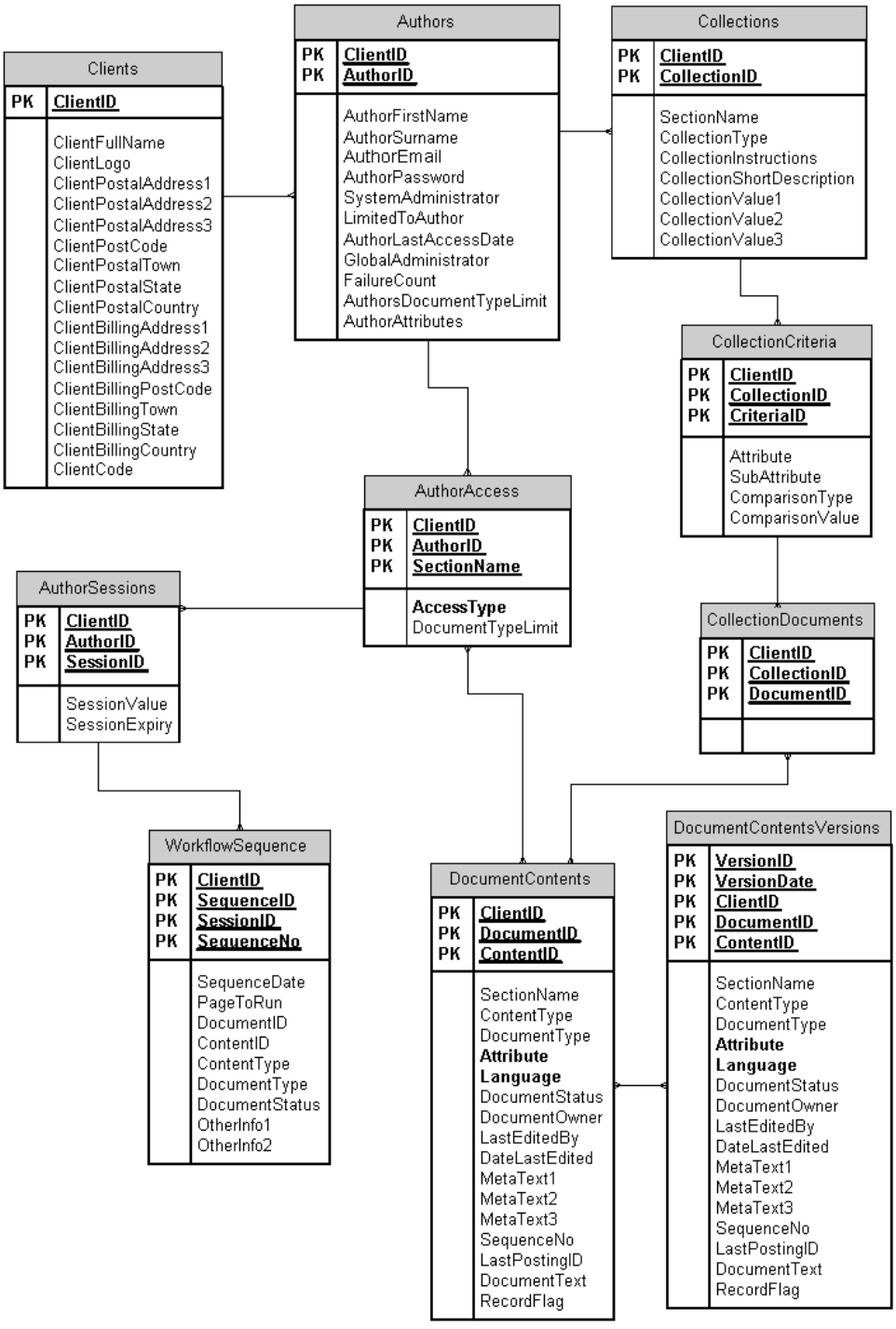


Figure 5.7: Document Ontology Schema

5.4.5 Layer 4 Database implementation – Transformation and Delivery

The database implementation of the top-most layer of the model inherits most aspects of the workflow layer. State-based information on the syndication delivery such as the *FtpQueue* table adds a management layer for failure management in the asynchronous publishing process. Most of the publishing metadata is itself represented as elements within the *DocumentContents* table, and is therefore not represented separately in the relational design. This allows the inclusion of publishing metadata in all aspects of content report writing and management. Figure 5.8 provides an illustrative schema for syndicated delivery of content transformed according to a set of transformation rules.

5.5 Inter-Publish User Interface Design

This section presents in detail the evolution of the content authoring User Interface (UI) over the life of the project. The immediate tensions in the user interface design related to the visualisation of the website content. The initial UI design emphasised the structural design of the fragmentation content while the user demand was consistently for WYSIWYG-style UI realisation of the content. The following sections present the evaluation of the workflow layer of the model and the ways in which the UI and workflow evolved to meet the competing tensions of content fragmentation and WYSIWYG visualisation.

5.5.1 Browser-based authoring interface version 1 (September 1999)

The first version of the prototype was developed largely in Microsoft Active Server pages in order to quickly deploy a publishing system that (a) fulfilled immediate web publishing needs and (b) elicited feedback on the Client interface elements required for a more complex application.

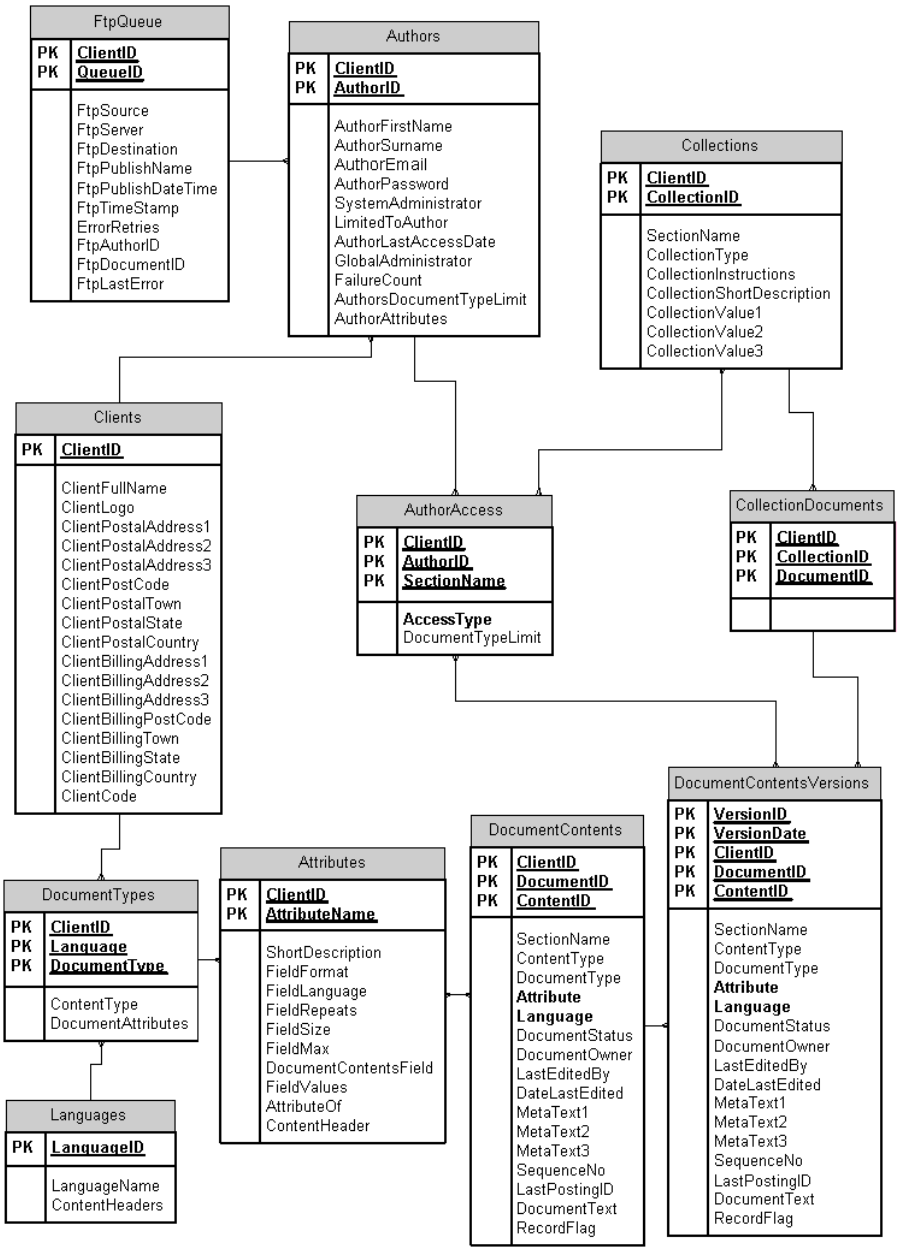


Figure 5.8: Syndication Schema

Session continuity with the server was achieved through ASP Session objects. These had the advantage of simplicity of implementation. During the development phases this approach revealed a number of difficulties:

- The session objects were lost on a Publishing Server reset, reboot or failure.
- The objects were bound to a particular instance of Microsoft Internet Information Services (IIS).

While satisfactory for a limited trial this architecture raised issues of scalability and system continuity. This issue was addressed in the second stage of the prototype. Figure 5.9 illustrates the 3-pane framed window UI design used to present the website structure and content.

In the design of the first stage of the prototype, it was apparent that the site structure of the available web sites would need to be revealed to the user. In the first stage the following model for information presentation was chosen:

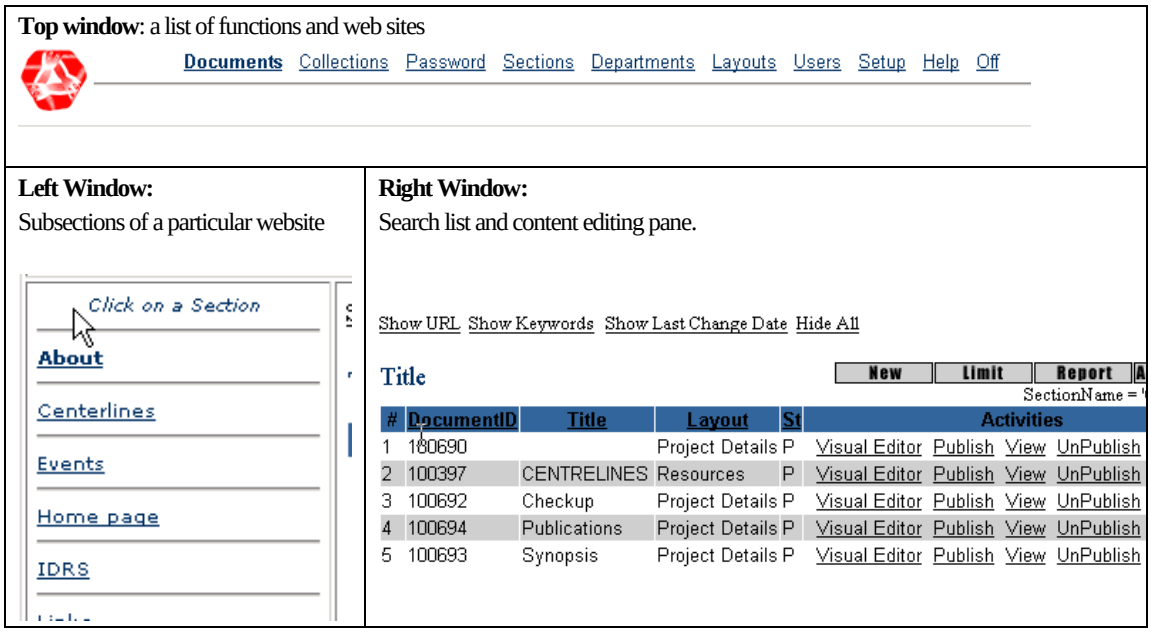


Figure 5.9: Three-pane content editing design

This first stage of the prototype was aimed primarily at exploring the technical issues of simple HTML content management. The content structures were limited to HTML content capture and the separation of content and presentation for HTML web page generation specifically. After a review of the initial prototype, users expressed concerns at the limited screen space available to present the list of documents and the content editing form.

The software framework that emerged in the first prototype provided a consistent structure in the further development of the software: a class-based approach to implementation of common functions – the document list, navigation functions and security/database access functions

particularly - and intelligent error reporting of software failure events to allow mitigation of problems as promptly as possible.

5.5.2 Browser-based authoring interface version 2 (December 2000)

The second version of the prototype extended the implementation of the model and addressed design issues arising from the first prototype workshop. The key elements implemented in this release included:

- The development of a flexible document ontology (layer 2 of the model).
- The transition to a scalable model in the server architecture.
- The introduction of multilingual elements with the language variants (layer 1 of the model)
- Generalisation of the model for content capture.
- A move to a more space-efficient navigational UI design.
- Session continuity on the server through the use of permanent Cookies (not IIS session-based cookies) with state management on the relational database server.

The feedback from this prototype was positive: it realised the need to capture a multiplicity of content types and provided an integrated database view of the content. However, as indicated above, several users expressed difficulty in the navigation. Integration of functions into drop-down lists in the top window had the advantage of maximising the data presentation area in the bottom window, but was at the expense of exposing the site structures in a way that was intuitive to user navigation.

In its user interface the design prototype consciously presented the logical dissection of content structures that is fundamental to the content reuse model. This was achieved through a two-window content presentation (see Figure 5.10 below). The UI expressed the content navigation in terms of the key content structures: sectional and content type navigation. The new UI design hid some of the site structure, privileging instead the logical structure of the content. Users expressed a preference for the first generation of design that visualised the site structure.

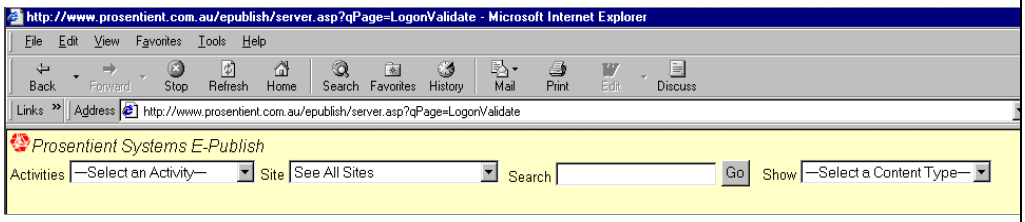
Prototype Version 2: Main Browser User Interface Top Window

The user interface was remodelled to give prominence to visualisation of the content model of information.

Information was presented in the following layout:

The drop-down lists dissected content based on:

- Functional Activities (editing, publishing, etc)
- Web sites registered on the system
- A free-text search field
- A content-based view of the data



Bottom Window

The catalogue of content item search results and document editing pane

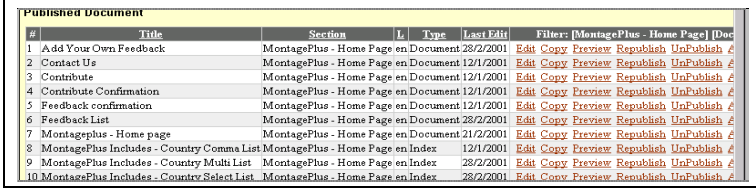


Figure 5.10: Two-pane content dissection

This stage of the schema development realised what (Fraternali & Paolo, 2000, p 2) terms HTML-SQL integration, providing a framework for large-database support to the publishing process.

5.5.3 Browser-based authoring interface version 3 (April 2001)

The third version of the prototype addressed remaining theoretical elements of the model – introducing modelling-based approaches to the site definition and implementation of a generalised framework for the multimedia document structure. It also explored several different visualisation approaches to the presentation of information in order to address difficulties users were experiencing in the discovery of content in a manner that matched their expectations. This phase of the prototype also focussed on integrative elements of the system and generalised models for workflow definition.

This version revisited the user presentational aspects to recover some of the advantages in site navigational presentation while still attempting a reasonable level of space efficiency in the interface within the limitations of Browser application delivery. The navigation model was now presented in the following style show in Figure 5.11, with a two-pane User Interface, the left window providing site navigation and structure and the right-pane acting as the authoring and activity window.

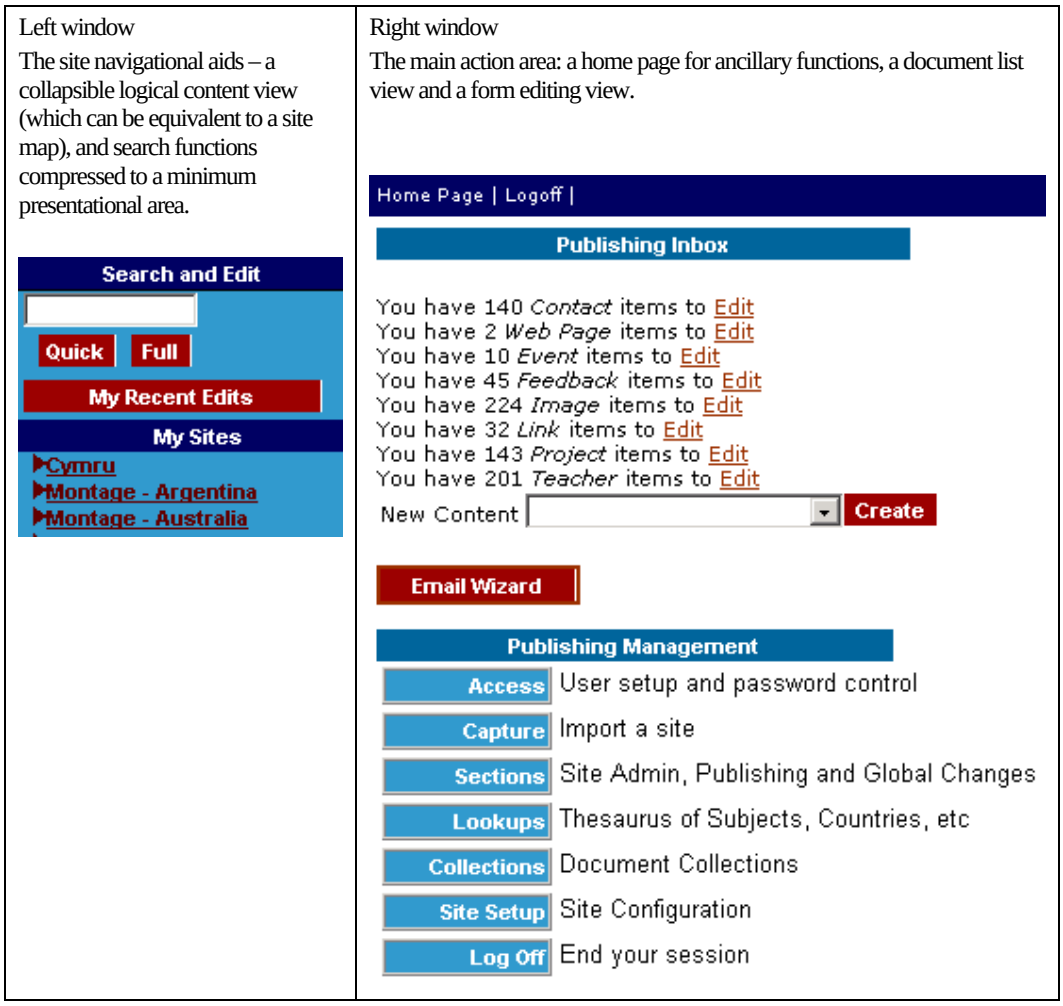


Figure 5.11: Two-window design – site navigation pane and content activity pane

Considerable effort went into the design elements of the third version of the prototype aimed at intuitive navigation, terminological consistency and the presentation of information in a space efficient manner. A persistent request from the users was for a more visually colourful and

appealing interface. A graphical makeover of the navigation, action buttons and format was undertaken in conjunction with user design reviews. Specific technical issues addressed in this version included further modularisation of content management functions for shared use with the FTP server, and better flow management of UI page sequencing rewritten to avoid the use of HTTP redirection on completion of actions (which caused problems through some firewalls). A table called *WorkflowSequence* was implemented to allow the flexible definition of sequences of page interactions.

While the ECU structures defined early in the database design were an enabler for multilingual content management, in this stage the ECU structure was also deployed to support the Internationalisation of Client User Interface itself. Where possible, all language elements in the User Interface design were referenced from an interface ECU definition which could be used to provide over-riding language translations for the User Interface Page, based on the language context of the user.

The key elements of the theoretical model realised at this stage were a model-based structural development of websites, the generalised Document Definition Language, generalised publishing, and scheduling mechanism.

The experience of applying the system in the context of the British Council Japan Montage implementation (called E-Link) in a multilingual context was salutary. Japanese has a multiplicity of character encoding approaches. The effective website implementation of the characters in one of the encoding schemes was complicated by:

- Server-specific approaches to language specification.
- Different encoding methods used by separate designers who were supplying the site design, and the need to rigorously enforce encoding methodologies through the application interfaces.

- Problem in the rendering of Japanese-encoded characters in visual editing tools – in particular a variety of bugs in the handling of Japanese-text HTML attribute tags such as ALT.
- Additional meta-information separate from the HTML language metadata directing the correct language for the search engine. While a number of search engines were tried for the support of the multilingual content, the results were only marginally adequate. As has been discussed, such difficulties in the User Interface for search design and the rendering of search results are themselves an area of extensive research, with consistent theoretical approaches only just emerging (G. Jones, Collier, Sakai, Sumita, & Hirakawa, 2001; Suzuki, Inoue, & Hasimoto, 2001). The approach by Jones et al (2001) is to integrate information retrieval and machine translational, an attractive direction that would warrant further exploration in the context of the ECU fragmentation of content (but is beyond the scope of this thesis).
- The need to experiment with various approaches to present the multi-lingual versions of an HTML document. While originally the design envisaged the co-location of Japanese and English text within the one document definition, users generally managed the content better when discrete documents, one for each language presentation, were created.

5.5.4 Abstract site modelling

Garzotto, Paolini, & Schwabe (1993) argue for a high-level UML-style modelling language for design of data driven websites. Their early work realises this model through extensions to an existing modelling approach HDL (hardware description language). Fraternali and Paolo (2000) extend this concept to propose WebML – a high-level modelling language for data-drive websites. Such an approach has great utility in the visualisation of content structures (Merialdo, Atzeni, & Mecca, 2003).

This prototype presents a hierarchical site structure for document organisation. While content reuse is applied at the component level, the imposition of a hierarchical order that organises documents and their content components serves several needs:

- It provides the intuitive hierarchical structure to express and navigate the document structures.
- It supports the security model for the editorial workflow.
- The indexing element of the software provides the transformational tool to dissect and organise content across the breadth of the content components. The component linkage mechanism also provides a means of expressing the content structures in a non-hierarchical manner. Examples of the application of this transformation are presented in Section 6.4.

5.5.5 Prototype version 4 (April 2002)

The fourth version of the prototype addressed further changes to content visualisation and modelling, and saw improvements in the software architecture.

The first (throw-away) prototype was written in Active Server Pages (version 1). Subsequently it was implemented as a VB WebClass (version 2 & 3) operating in the Microsoft IIS web. In version 4 a Web-DAV enabled HTTP web server was written by the author to complement the existing FTP server to give greater portability to the final implementation. The exercise in writing a simple HTTP web server was trivial in the early stages but quite time consuming in the bedding down of a stable architecture. Besides implementation of RFC1945 (HTTP 1.0) and RFC2616 (HTTP 1.1), the foibles of many browsers and proxy servers required considerable additional exception coding. However, vagaries of client and proxy behaviour required as much coding for exceptions as for the original coding of the standards-based web server. This was an unfortunate (but typical) additional labour required to achieve a stable web server for client access in the context of different browsers and proxy server intermediaries.

To allow effective multimedia file management, it was also necessary to implement RFC 1867 multi-part file encoding support. This also enabled better support for multi-lingual content encoding and visualisation (see Figure 5.12 below).

The outcomes of this transition, where, however, beneficial to the capabilities of the prototype to reflect aspects of the model. The result is an application that exhibits good design practice in Scalability, Modularisation and Flexibility. The architecture will support multiple load-balanced front-end servers. An advantage of the design is that these do not have to be high-end Microsoft servers – the design can operate at present on any Microsoft Windows platform from Windows 95 or later. There was modularisation of key functions internally: a comprehensive class library now exists supporting both the FTP, HTTP interfaces to the content, as well as a set of web services interfaces at all layers of the model. Only a little further development work (and a great deal further documentation) would open the design architecture up to allow other developers to access the content management modules of the software. A brief overview of the key web services functions is presented in the appendices. Flexibility in the security design of the architecture allows multi-level inheritable security for users.

5.5.6 Prototype version 5 (January 2003)

A key visualisation innovation in this stage of the prototype introduced a virtual site browsing concept in which the original single page "preview" of content was remodelled as a virtual site editing tool: the preview mode:

- Presents visually as far as possible a given page rendered as it would be seen in the target publishing server
- Adds editing icons to allow direct hyperlinking to content editing of a particular content item wherever in the content model it is situated
- Adjusts all hyperlinks to continue virtual site browsing of the target link rather than linking to the page itself.

The original logical view of the site (based on content type not physical site) was retained in the left-hand pane of the browser window. New users of the software invariably preferred the virtual site browsing and the search function as their main method for discovery of content.

Finally, full WebDAV integration was introduced after selective field-testing in December 2002. The integration of WebDAV into the prototype extended the capability of the user interface to support multiple heterogeneous client tools while sustaining an effective content view.



Figure 5.12: Context-sensitive visual editing in a multi-lingual ECU

5.5.7 Workflow Management Concepts

The first two layers of the model were implemented quite early in the design of the prototype. The workflow implementation of the prototype was directed particularly to web publishing and went through many evolutionary changes as the project progressed. Workflow primitives were also recorded as ECU's in the document model of the CMR. A document type of *Workflow* was defined, describing for the "Publish" and "Unpublish" actions of Inter-Publish the consequential affect on the document being publishing. The content administrator could define a new document type to which the content was mapped if it was published (or unpublished), with the consequent changes in authoring visibility and publishing actions mandated by that document type.

An example of the Inter-Publish workflow is the "Future Connections" competition developed by the British Council Australia. This was a public essay competition. Entries were

submitted on a web-based form, generating an XML document interfacing action to Inter-Publish. The resulting "Future Connection" application was moderated and approved (or discarded if the entry was spurious). Publishing the application resulted in an email transformation and delivery action sending an email to the applicant, and changing the document type to "awaiting". Once the applicant's essay arrived, a second publishing event changed the document type to "review" and caused a further email receipt acknowledgement to the applicant. Selection of essays for the shortlist similarly caused a document transition to "short listed". The workflow document types are illustrated in Figure 5.13 below.

#	Published Entries	L	Type	Last Edit	Status
1	Future Connections Application	en	Workflow	09/09/2003	Published
2	Future Connections Awaiting	en	Workflow	12/09/2003	Published
3	Future Connections Review	en	Workflow	14/09/2003	Published
4	Future Connections Shortlisted	en	Workflow	09/09/2003	Published

Figure 5.13: Workflow example

Documents could be routed with alert via email to targeted individuals or group email addresses. Complex routing to multiple target groups was not implemented and was not required by the client. Simple email groups are supported in the workflow ontology. The detailed tracking of content version information would probably allow detailed analysis of performance metrics, but no such tracking was implemented.

In the Inter-Publish workflow processing, there are capabilities for mass queue management but these are not explicitly directed to workflow management. Workflow status information is realised in the Inter-Publish prototype as ECU-based content and therefore is amenable to all global change features established in the user interface. The Content Management features allow reallocation of all aspects of document status, and en-masse management of changes to content, including by virtue of this the queue status. Documents could be routed with alert via email to targeted individuals or group email addresses.

Event management is limited to alerting for content publishing events, and other aspects of event management, including workflow management, simulation, monitoring of status and data-driven workflow state change have not been a feature demanded by the client community within the context of the Montage experiment. These would obviously be elements of a rich workflow implementation and are obvious elements of systems for which workflow design has been a focus of greater attention (such as Documentum).

While there are many areas of potential extension of the workflow capabilities of the software, the workflow capabilities of the system more than met the workflow requirements of the target field test environment described in Chapter Six. The workflow management interface is documented in Appendix Three.

5.5.8 Authentication

The prototype implements user and role-based security on the web server using HTTP basic authentication and through Web Services using a simple authentication dialogue.

An author registered through the prototype can be one of:

- A global administrator - can view content across multiple systems
- A system administrator - can perform all functions within one system/client
- A designer - who can perform major content changes and designs within nominated sections
- An editor - who can be cast for particular sections with “read only” or “edit” access to content.

Group-based rights and LDAP authentication were not implemented.

5.5.9 Controlled Vocabulary and Metadata

The fragmentation of content at the ECU layer of the model provides an effective framework for presentation of controlled vocabulary. At the simplest level, a controlled vocabulary can be represented as a structured set of ECU elements, which can be selected for

participation in a single-level controlled vocabulary. As a result, a workable generalised and simple hierarchical controlled vocabulary was implemented, as a set of reusable content objects. For ease of controlled vocabulary management, ECU elements selected to participate in a controlled vocabulary were made available for editing in a "Lookup" menu option. Similarly, ECU's can be used to build a controlled vocabulary from existing content (for example City Name, or Project Name) that is otherwise uncontrolled but relatively reliable.

A separate management page for definition of controlled vocabularies was incorporated in the browser-based content interface, providing a separate level of workflow control for definition of these ECU elements. For purposes of this evaluation, the controlled vocabulary was a simple single-level list. This architecture could easily be extended to a more complex hierarchical tree of controlled vocabulary items. More complex 2 and 3 level controlled vocabulary hierarchies could be implemented by defining a Document structure for the hierarchy and making the lowest layer of this hierarchy available for selection.

The architecture for metadata and controlled vocabulary management served as a solid base for the prototype and required very little modification during the life of the project. The primary area of extension in the management of metadata information was the progressive extension of the metadata associated with the content units – that is, the metadata pertaining to the method of encoding, MIME type, presentational format for content units, and language handling metadata.

5.5.10 Confusion with fragmented content

The objective of the client interface was to realise theoretical aspects of the presentation modelling layers of the prototype in a manner that tested the suitability of the conceptual model in a web publishing environment. In this context, the Browser-based client interface was intended specifically to provide an integrated view of the heterogeneous content elements expressing the requirements of Layer 3 of the Content Model for Reuse. The principle function of this layer of

the model was the organisationally contingent workflow management of the authoring and publishing process, including the content discovery and content visualisation process.

The fragmentation of content presents an essential difficulty in visualisation at the authoring level of the interface. This section explores the authoring interface as it changed to meet the authoring needs in the context of the Montage project. The workflow model that evolved was generalised as far as was possible in the scope of the project. The author explored several user visualisation approaches with the target community, from interfaces that privileged the logical hierarchy of the content (with which users struggled) to those that privileged the reassembly of the content in its published form.

The early prototype visualisations of the content suffered two key difficulties:

- Many end users of the CMS did not necessarily understand or think in terms of the content as separate units of information.
- End users discussed content in terms of the published web page rather than the underlying content.

The extensive use of content reuse concepts not only for syndication of and reuse of information content but also in the reuse of design concepts for the site (for example navigation bars) resulted in considerable fragmentation of content within the CMS. For example, a standard published web page might be "assembled" from a layout which itself draws on content units representing the standard top navigation, side navigation and footers for the site (pre-publishing "includes") and standard metadata, style sheets and dynamic scripting definitions. This page would also be built from one or more content units containing the content to be "wrapped" in the layout, this information dissected at the defined level of content granularity

The principal difficulty in the first two prototype CMS user interfaces was the effective rediscovery of content as it pertained to the published site. This issue strongly reinforces the

importance of understanding the semiotics of Content Management, particularly in the context of the separation of substance from particular expression. Indeed, the frustrations with the second-stage prototype exemplify the issues surrounding the nature of the content and expression discussed in the literature review. It would be interesting to explore the extent to which the reification of separation of content and structure combined with a dogmatic critique of WYSIWYG itself ignores a fundamental aspect of the semiotics of the content communication process. The user may inevitably focus on the final expression, rather than on the componentisation of the content units required to represent the content in meaningful units.

The obvious benefit of WYSIWYG lies in the very expression of the content in a semantically complete context of final expression. The challenge for the CMS workflow layer is to find a middle way between WYSIWYG and good content separation, as well as the elicitation of structured metadata describing the content.

The advantages and disadvantages of the browser as a client interface were apparent during this development process. The advantages of server-based application delivery in a browser-based client included the ability to update the user application rapidly and seamlessly. This had considerable advantages to the flexible support of an application prototype that was developing at a rapid pace. The architectural design of the software was, by the third prototype, robust enough to sustain user continuity through restarts of server processors or indeed cold restarts of the server itself. The disadvantages, common to many server-based applications, was the speed of page transition and the limited "screen territory" available for Client Interface design. During the evolution of the prototype a number of approaches were explored for the presentation of the site structure. The following is a brief overview of the prototyping stages that led to the current Content Visualisation approaches. Content designers generally preferred to work as much as possible with their own content editing tools, interfacing to the CMS through FTP and WebDAV. New users introduced to the system, content creators and editors, tended to prefer working through the browser interface, even after the introduction of an FTP server.

5.5.11 Text mining of the ECU for authoring support

The above history of the UI evolution indicates how dramatically the content visualisation approach changed over the period of the project. A further challenge of the field test was the difficulty of eliciting metadata from the end user. While this metadata information remains important in the discovery of content through search engines, it remains a challenge to elicit this information from the content author. The ECU fragmentation, being situated in a well-structured content framework, offered the opportunity to explore some simple text-mining techniques to ameliorate this difficulty.

Facilitating semantic analysis of content as it is captured depends of efficient analysis of the content. Approaches to text mining have been receiving increasing attention, both in the context of search-engine ranking systems and email content analysis (Di Iorio, Diligenti, Gori, Maggini, & Pucci, 2003; Nahm & Mooney, 2002).

The CMS presents an opportunity to deploy multi-methodological approaches to the text mining process drawing on the rich contextual resources available to the Content Management System. This contextual information can be deployed to achieve high levels of document similarity matching by combining text similarity algorithms with sematic information identifiable from controlled vocabulary information and the explicit navigational context in which the author has chosen to situate the content during the authoring process. The objective of this analysis is to assist the author in selection of suitable descriptive terms for the authored content, excluding stop words, and to assist the author in enhancing the interlinking of the content with other similar documents.

Web-based ranking tends to favour link and anchor information over content-based analysis. However, in the context of the CMS, with content already separated from style, content value for similarity matching is improved.

In our prototype CMS, the author is presented with content styles to select from when creating new content. Content was presented to the author in fully styled framework, with hyperlinking to the underlying content forms using simple editing tokens.

Simple text mining techniques were applied to the content supplied by the author to facilitate the presentation of metadata and inter-linking suggestions grounded on an understanding of the context of the newly created content after initial submission by the author of content. Content submitted by the author was initially parsed to formulate a bag of words (Figure 5.14).

$$B_c = \{(w_1, f_1),...,(w_n, f_n)\}$$

where w_n is the word f_n is the frequency of the word within the document (Haveliwala, Gionis, Klein, & Indyk, 2002).

Figure 5.14: Bag of words

The bag of words was retained in a relational table representing the content space, after some limited word stemming and stop-word removal. An initial cluster of documents is extracted through a simple ranking measure (Figure 5.15).

$$R(d_x, d_y) = \max\left\{\sum_{i=1}^{n} \left(w_i / f_i\right)\right\}$$

where *n* is the bag of words in d_x and documents are chosen based on their maximal score based on the frequency of the d_x terms in d_y as a ratio f the frequency of the term across the full collection.

Figure 5.15: Similarity measure

This ranking measure lent itself to a fast relational select across the bag of words collection and served as an initial clustering collection for further analysis.

Final ranking of document similarity on this pool is achieved through calculation of a Jacquard coefficient (Broder, 1997; Haveliwala et al., 2002) to provide a more accurate similarity measure in the in the initial cluster (Figure 5.16).

$$J(d_i, d_j) = \frac{|d_i \cap d_j|}{|d_i \cup d_j|}$$

Figure 5.16: Jacquard Coefficient

This can be used to produce a rough initial clustering very rapidly from the document base. The final set of documents is presented to the author categorised by logical website section (if available) and ranked.

The bag of words B_c is also used to calculate words for semantic metadata enhancement by selecting terms from the bag of words with a very high or very low ratio, with additional weighting on terms used in the document title. The set of terms derived through this process can be processed through a simple co-word analysis to derive phrases by:

- Extraction of likely proper names.
- Identification of phrases through selective co-word analysis favouring words with a high level of uniqueness within the document itself.
- Identification of phrases through selective co-word analysis based on words that have a high frequency in union with the high-ranked pool of documents selected through the Jacquard coefficient.

Finally, the content author validated the keyword selection.

Other text mining and phrase analysis techniques warrant further analysis, and this represents one area of possible further research to determine the degree to which the ECU-structuring of content and the rich metadata information available enhances their efficacy. The simple techniques described above were at least sufficient to improve the threshold keywords. The user community resisted mandatory input of keywords, and voluntarily attributed keywords for content never went above 5% of documents. The combination of improved keyword capture (with 90% of now having some keyword descriptors) and guided website linking provided a visibility to current content search engines.

5.5.12 WebDAV authoring interface

The attraction of implementing a WebDAV protocol server in the prototype was twofold: firstly, and importantly, the integration with commonly used authoring platforms; and secondly, the possibility it offers for a flexible content architecture. While WebDAV is commonly implemented as an authoring workflow interface to a Web Server where the web server is both the content repository and the content runtime engine, this prototype implements WebDAV as an interface through to the content store. Such an implementation was facilitated by the near-complete approach to integrating XML in the interface approach. Frustrating irritations in the development of the web server centred on the uneven application of standards among clients and in the specification itself - for instance, while Appendix Two of the WebDAV RFC2518 orients to ISO8601 for *creationdate* the *getlastmodified* element uses an HTTP 1123-style date. Nevertheless, WebDAV provides a framework very amenable to implementation as a content database front-end, and is seeing steady adoption now that it is integrated as the "Web Folders" feature of the Microsoft operating system offerings.

The server implementation in this prototype is based on RFC2518 for WebDAV (Goland et al., 1999) and RFC 2617 for web-based authentication (Franks et al., 1999). For purposes of the prototype, only Basic authentication was implemented.

The first stages of the prototype were implemented as web classes operating in a Microsoft Internet Information Server (IIS) framework. The planning for WebDAV began in early 2000, as it required a major technical reorientation of the prototype. In particular, to use WebDAV in a manner that retains the content view of the data rather than a web-server file system view, it was necessary to develop an integrated Web Server and Web-DAV server separate from IIS.

The first stage of the Web-DAV implementation, therefore, was the completion of a functional HTTP web server framework for Layer Three of the model. Implementation of WebDAV was thereafter an extension of this implementation to support the WebDAV protocol

RFC. The Delta-V extensions to WebDAV supporting document versioning would be a natural extension to the work completed to date.

While the implementation of WebDAV support was an attractive extension to the capability for integration with client-side authoring tools, it cannot be said that the protocol has quite the reach achieved by FTP at the client-side level. FTP as a protocol for content access is ubiquitous. The adoption of WebDAV for Microsoft's "Web Folders" for content sharing will see the protocol more widely adopted. While a characteristically cavalier Microsoft implementation of the relevant protocols, it provides a useful interface for testing the protocol. The WebDAV implementation was also tested through Dreamweaver. The seamless integration with client-side authoring tools using Microsoft's "Web Folders" is limited by the vagaries of their implementation – particularly their determination to give precedence to the (potentially arbitrary) file suffix over the content type supplied (a long standing issue with Internet Explorer). Changes to Microsoft's WebDAV interface between their Internet Explorer 5 and Internet Explorer 6 required considerable special handling at the server end.

5.5.13 FTP Authoring Interface

An FTP server interface to layer 3 extended the availability of the content to a wide range of content authoring environments. FTP clients are ubiquitous. The approach taken in the prototype is to implement an FTP server to allow the user to navigate the standard content hierarchy and to support FTP GET and PUT functions in normal and PASV mode to exchange content located in the database or fetch-able through the publishing system interfaces. Extension of this facility also allows publishing through this interface. Implementation of the FTP server was non-trivial, given the vagaries of protocol implementation and timing on different FTP clients.

The server implementation in this prototype is based on RFC 0542 (Neigus, 1973) and has been tested with a range of clients, including Internet Explorer 4, 5 & 6, the Mozilla browser, FTPPro95 (WsFTP), Macromedia Dreamweaver, Macromedia Homesite, and command-line

FTP clients such as available in Linux and Microsoft Windows platforms. While the interface appears relatively stable, only the above clients have been explicitly tested, and some firewalls have caused timing difficulties with the server. Support for some FTP commands is deliberately limited (such as new sub-site creation using MKDIR) because this interface is primarily a content editing view not a site creation/modelling interface. There are difficulties with MACBinary support in the upload of files from Apple MAC systems probably relating to differences in binary encoding of information in the FTP interface.

The implementation reflects the use of FTP as a front-end to the content databases. As such, the file names reflect the content hierarchy - rather than file system handles. Actions taken on GET, PUT and NLIST commands are acting on the content database rather than the file system. FTP is therefore used as a functional interface to the relational database store. This presents interesting opportunities to automate trigger actions on updates that come in based on the CMS content workflow rules. The content hierarchy presented can reflect the level of access relevant to the user and can be based on the same workflow rules and access security. The FTP server component was tested against several FTP clients commonly used at the British Council: Microsoft FTP client, WS-FTP Pro and Dreamweaver.

This approach makes it easier to interface with other authoring tools, such as Dreamweaver or Word. Commercial products have invested in specialised plug-ins to popular authoring tools such as Word. Obtaining clean content that is well structured for content reuse is not, however, a simple matter when dealing with convoluted formats such as Word. The author developed a parsing function (and observed similar functions in other CMS products) to facilitate some of the more problematic Word markup constructs.

5.5.14 Web Services authoring interface

The Web Services interface is at its most rich at the workflow layer of the model, being the principal authoring interface with the content. Web Services provided a very effective approach

to cross-application integration, both in outward sharing of content, inward import of content and integration with other workflow and scheduling agents.

The challenge of large content services such as the Montage project is the demand for management of complex interchange of content. For instance, Montage authors would source a list of existing schools from an Access, Outlook or other contact relationship management (CRM) system. Web Services provided a useful public framework for inter-database exchange of data. The most interesting application of this layer of the interface was the exchange of content with the CMS product ObTree. At this layer of the interface, content was managed in Inter-Publish but authored and published to non-Montage databases through the ObTree CMS. This was a good example of the flexibility of the Web Services approach and of the content model to content interchange.

5.6 Content Transformation and Delivery

5.6.1 Content delivery in the Field Implementation

Layer Four of the CMR model pertains to the public interface to the content. Content is assembled and transformed to yield the final form available to the end user. This initial transformation by Inter-Publish was itself just a prelude to additional selection and transformation - for example by intelligent agents and personalisation engines on external web servers.

Figure 5.17 below illustrates the relationship between ECU content and final expression in one of the Montage E-Zine design templates. This format was sent by email to participating schools.

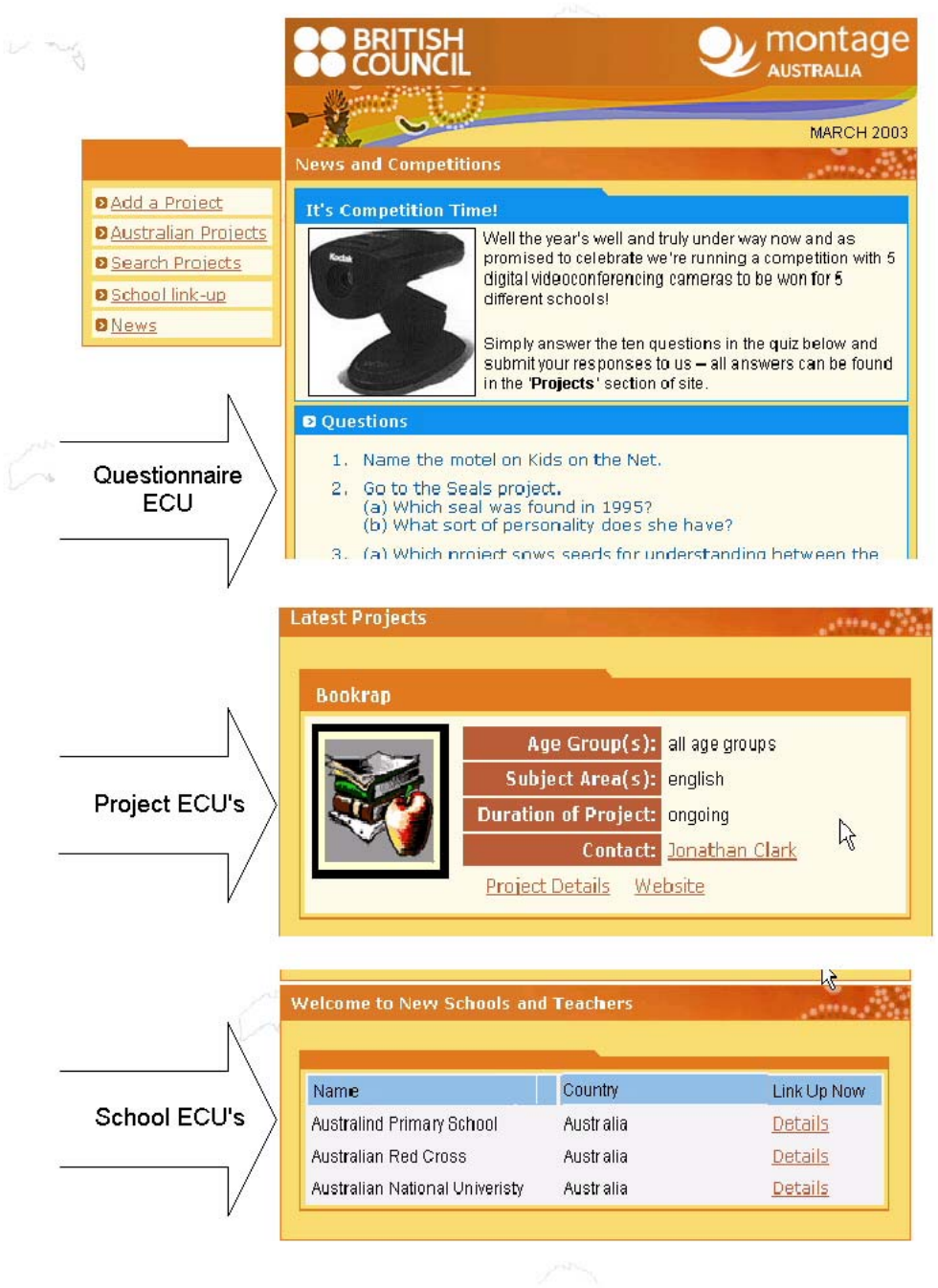


Figure 5.17: Content reuse in E-Zines

As discussed previously, CMR expresses an information system rather than a single ontology. The transformation process at Layer 4 was supported through two approaches. The Inter-Publish Template Language - an embedded template-oriented markup and transformation language designed to transform ECU and document content into other forms. Publishing tasks also allowed the initiation of post-generative XSLT transformations and the initiation of other,

external, scripts, although this aspect received very approach attention in the Montage trial. While these could be considered interchangeable at this layer of the model, there was efficiency in the integration of a specialised transformation syntax for Inter-Publish. The Inter-Publish Transformation Language is a lightweight transformation engine for performance of high-volume and complex transformations. While lacking the plurality of XSLT, its strengths lie in the efficiency of transformation that it provides. The strengths of XSLT lie in its system-independent XML transformation syntax and its rich transformational capability, and (notoriously) not in its performance across large document sets, nor in the simplicity of scripting.

Inter-Publish was used to provide some personalisation facilities for the Montage website. This personalisation could be characterised as a final transformation of the content mediated by the web server prior to delivery of content to the targeted end users. This mediation is an intersection between the clients' known interests and the content available. The Inter-Publish database maintained document entries not only of the content related to the projects, but also detailed information pertaining to the registered schools This allowed targeted delivery of content on the website (through simple personalisation functions) and through email updates and information mail outs.

Crucial to this layer of the project was the development of a rich template-based indexing/report writer for regeneration of content into its target runtime media form. This indexing engine, operating on a template basis with an integrated scripting language, provided the framework for both primary generation of content to be used in runtime engines, and for secondary generation of content to be interpreted by final-stage content transformation engines. For example, the CMS was used to generate XML content for dynamic transformation in the website context using XSL.

The following transport agents were implemented to support this layer:

- FTP delivery of content to external web servers with generalised definition of FTP parameters separating logical and physical publishing characteristics. Inter-Publish was designed to operate at arms length from the target web server, delivering static content to existing Web Servers.
- SMTP (Email) delivery of content on a single or mass-mailout basis with generalised selection of content and template-based delivery of emails in either plain text or HTML form.
- HTML (Browser-based) content delivery, particularly for printed content.
- XML content delivery for syndication over an HTTP transport layer using RSS.
- Web Services delivery for high-level content mediation (again largely syndication).

In most cases, final content delivery to the Montage Clients was by either email or the British Council Web Server. Controlled vocabulary lists were used to facilitate content searching across published content, aided by Microsoft Indexing Service. The Web Server also served as the platform for distribution of XML-based syndication content.

Content was also published to a Conferencing Service (O'Reilly's WebBoard), although the ongoing moderation of the Conference discussion proved too time demanding for British Council staff and was discontinued.

Content was published in email form in E-Zine style newsletters which deployed current content from the Montage database in order to keep subscribing customers informed on developments.

A virtual browser interface to content provided a means of published content browsing, without an authoring interface. Such an approach was valuable in providing a QA method for non-authoring users to review content prior to publication. While this feature is commonly achieved through a "staging server", it was useful to provide a fully generative view directly from the content. Obviously such on-the-fly website generation is quite demanding, and this interface

was not used for purposes of normal public access to the content. Vendors of some CMS products incorporate such facilities with highly tuned caching strategies (especially Vignette).

Finally, Web Services were used for integration at the authoring and content distribution layers of the model. The Web Services interface provides a flexible standardised content interface to the model, allowing content interchange with other CMS products, and in particular with ObTree, the other CMS used in the British Council. It was used by the British Council to build content bridges with other products containing useful information resources – such as Outlook.

5.6.2 Browser-based navigation and publishing interface

The prototype is implemented as several servers. The web server itself provides both a browser-based interface and exchanges content using XML with other services: It can also be used asynchronously to manage content through defined URL calls using HTTP GET and POST. This has the advantage of universal accessibility through firewalls, and allows simple automation of processes through any scheduling system. Many other applications can now make URL-based queries.

Defined XML interfaces with email acted as the principal method of interchange with external content sources and web-based. The email-based XML interface is particularly useful in asynchronous collection and moderation of data from web servers that are data sources for the publishing system. Once again, email clients are generally readily available on web servers and information transfer is not usually impeded by firewalls.

5.6.3 Inter-Publish Transformation Language

An important innovation in object-oriented programming was the reorienting of programming style from a procedural orientation to that of “design patterns”. This concept can be realised in publishing forms also, and is exemplified in the Montage websites. The Inter-Publish

Transformation language is an important element that enables this design reuse. The transformation syntax (described in detail in Appendix Three) provides a lightweight framework for hierarchical embedding of templates and documents - some of which themselves may have complex content selection and transformation rules.

In its simplest form, the language provides the definitional structure to transform ECU content into its published form, a model for content transformation adopted by many CMS products (see Figure 5.18). The layer 3 browser-based forms interface provided the user view of the underlying ontological structure, providing content separation. The publishing process re-aggregated content to produce the final published content in text, HTML, XML or PDF.

Each Montage site follows a similar pattern, and the HTML template models use such conceptual design patterns. Subsidiary templates can be included by simple meta-tagging constructs - for example "<<include 'subsidiary template or document name'>>".

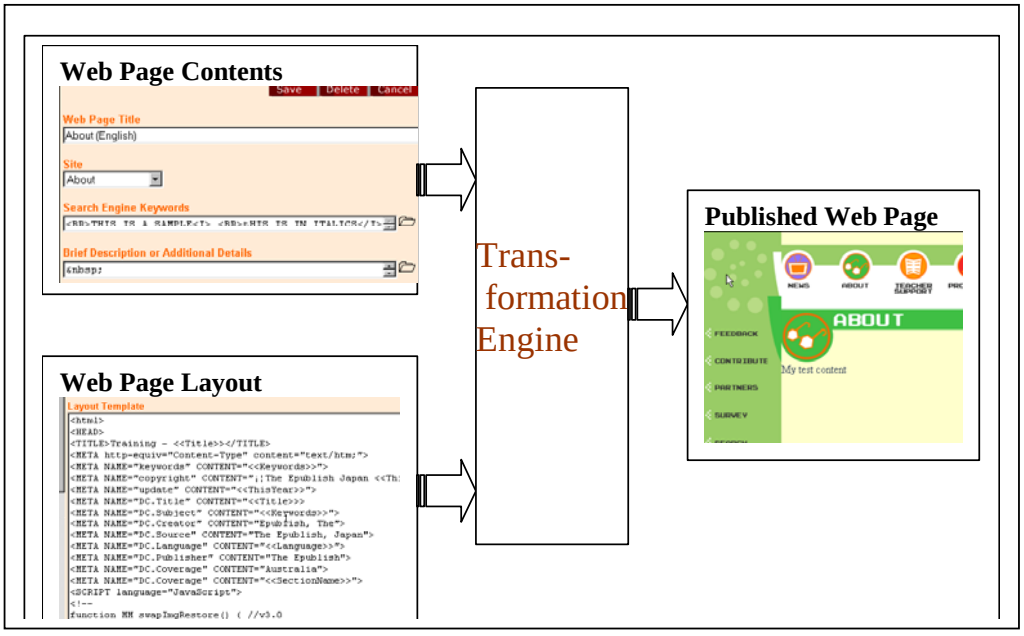


Figure 5.18: Simple transformation example

5.6.4 Syndication

The principle method for syndication in the Montage field test was through RSS syndication (see Chapter Two) in the format specified by RSS 0.9. The two elements of this syndication are:

- A static collection selection rule selecting Montage Projects that have been added or changed in the last week.
- A transformation template to map the ECU contents to the RSS 0.9 standard.

5.6.5 XML profiles for Personalisation

XML is the natural vehicle for many forms of information exchange between systems. In the case of the Montage field test, XML was used for personalisation of school content on the target web server. Inter-Publish was used to create and update XML definitions for registered participating schools on the target web server. Further personalisation was then performed using ASP scripts using these XML profiles.

5.7 Testing, Quality Assurance and Documentation

The prototype was used for the active publishing of website content. Clearly a satisfactory level of Quality Assurance was required while still achieving a reasonably rapid update of new functionality. *Beta* testing of the software was undertaken on a development server that mirrored the data contained on the current software version. *Alpha* testing of major releases was performed by staff at the British Council on a development server prior to implementation.

Implementation was accompanied by email documentation updates, with major version updates to the documentation at stages of major functional implementations.

With an internationally distributed community of users in the Montage project, email was an essential method for supporting the software development process. Email was also useful in

assuring the stability of the prototype, with email alerts providing an effective means of detecting early any error conditions in the software. This intelligent alerting for software faults proved invaluable as the complexity of the software itself grew.

The prototype software was implemented with integrated alerting for failure at two levels:

1. Error trapping in all cases generated email alerts on all errors giving detailed information on the nature of the error. This form of automated error feedback was very valuable in providing quick reaction to application errors. Figure 5.19 illustrates such an alert.

2. Status monitoring through external agents of the server status and availability.

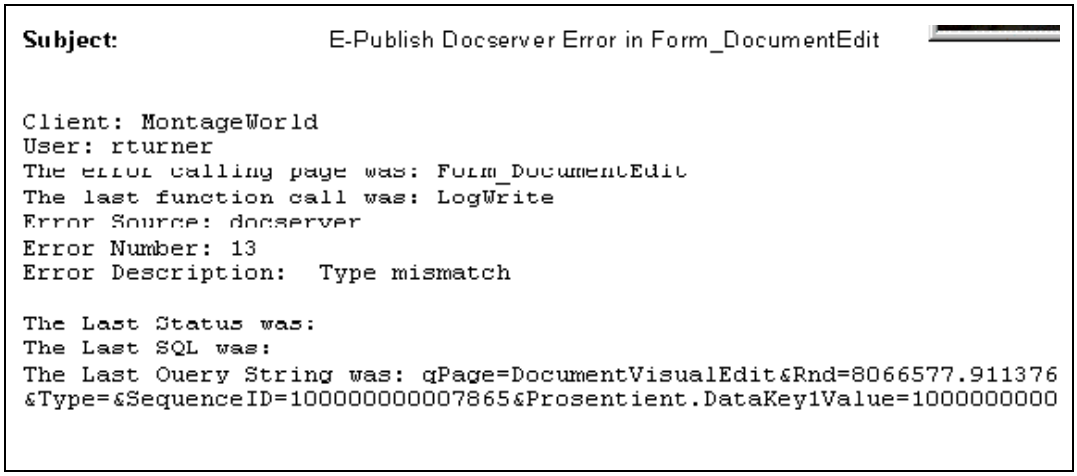


Figure 5.19: Internal alerts

Detailed documentation for the Inter-Publish Template Language is available in Appendix Three, with links to a User Manual and System Administration Manual.

5.8 Conclusion

This chapter presents the design of the Inter-Publish prototype. The prototype began with the principles elicited in the literature review and the CMR model described in Chapter Four, and evolved to meet a number of User Interface and publishing challenges during the life of the project. Principle among the UI design challenges was the tension between content fragmentation

for content reuse and the visualisation of content in its published form for the user. Additional technical challenges in server design arose from the demanding nature of the Montage field implementation and the exploration of different issues in multimedia and multilingual content encoding and management. The next chapter describes the Montage project and leads to a detailed evaluation of the Software Development Research presented in Chapter Seven.

Chapter 6 – The Montage Service

Chapter Four introduced the Content Model for Reuse (CMR) and Chapter Five presented the design of the prototype developed to test the model in a comprehensive field test. A key element of this Software Development Research was the realisation of the model in a rich prototyping process explored in a complex field trial testing the reuse and regeneration concepts across a large document collection. In 1999 the author discussed with the British Council the prospect of implementing a Content Management System (CMS) to extend the content reuse of a nascent but interesting online educational website. The resulting Montage service was the testing ground for the Inter-Publish prototype. Beginning from a base of simple website publishing progressively more complex functional aspects of the model were implemented. This Chapter introduces the project and presents its practical outcomes. Chapter Seven then presents a detailed evaluation of the prototype and its theoretical outcomes in content reuse.

6.1 The Montage Project

The Australian office of the British Council has been responsible for the innovation of a number of online educational resources targeted at primary, secondary and tertiary students. The organisation had broad web publishing ambitions and an imminent need for systems better to manage its resources. It was willing to participate in a field trial to implement an advanced concept for Content Management and Content Reuse.

The online education resource that is Montage was originally piloted in 1998 as a means of linking schools in Australia and the UK in undertaking curriculum projects in areas of mutual interest. The early challenge of the project was to establish an effective means to manage the international growth in participants, facilitated in a manner that maximised resource sharing while sustaining a high level of quality control and content moderation. The project was constrained by limited staff resources.

The initial pilot was so successful that British Council sought to extend the Montage concept to act as an online portal for teachers. The Montage pilot demonstrated the strong need of teachers to be able to integrate the use of the Internet into the classroom situation. There is a great demand for pedagogically sound curriculum material online which can be shared and which is safe for students.

The author first worked on Content Management strategies for the Australian Broadcasting Corporation in 1996. This experience inspired the desire to formalise strategies for content reuse. The author's involvement with the ABC led to joint work with the ABC and British Council on early Montage website support resources, prior to commencement of this thesis. The British Council Australia was an enthusiastic participant in the proposed prototype for content reuse. At the commencement of the field trial, the Sydney office of the British Council operated two web sites:

- *www.montage.edu.au* - their Australian corporate gateway to tertiary education resources
- *www.bc.org.au* - their Australian corporate gateway.

Their suitability as a base for a field test of the prototype was enhanced by the increasing demands for website content management support, and particularly for their need to manage multiple different websites in conjunction with other Non-Government Organisations, their use of multimedia elements such as audio, video and flash and their requirements for regeneration of content in HTML and email form, and for printed mail-outs, and reuse of content (particularly schools information) in a number of different contexts.

The first publishing prototype was implemented in their office in mid-2000. By December 2000 the underlying content layers were in place and formed a stable framework for the next two years. During the period of the field trial, their demands for additional websites resources grew rapidly, supported by the existence of the publishing product to facilitate their publishing

ambitions. In this respect particularly the Action Research aspects of the prototyping field trial were very evident: the goals and objectives of the Australian office were expanded as their publishing capabilities and content reuse facilities increased.

Finally, their international base offered an attractive environment for testing the content management concepts in a multilingual environment.

The Montage project has been built on collaborative partnerships between organisations in the arts education and science and technology. Each partner has brought to the programme something that the other does not have. This is not necessarily in the form of financial assistance but usually in-kind support. Each curriculum project published through Montage includes further collaborative contributions from relevant Arts or Science departments in the British Council with input from schools and support/oversight from Education departments. That collaborative model saw five other countries in the Asia Pacific join in the "Montage" service with their own country sites.

The "Montage" concept is founded on a technological framework that maximises collaborative input of participating organisations and schools as an enabler for effective, moderated and low cost distribution of the information. This framework provides an effective means for a small team of participants to manage the process of international collaboration. Arrangements are in train to establish similar services operating out of the UK. Each curriculum project has a project co-ordinator to manage and moderate the project. The content is agreed by the Education Departments for educational appropriateness.

The project is published through the website establishing linkages between schools and their project interests – and republished using the Content Management System for reuse in various country sites and through e-news letters.

The British Council, using the Inter-Publish CMS hosted by Prosentient Systems, manages ongoing communication with schools. The service provides collaborative interchange of

information between schools and education departments and enabling cost-effective content distribution. A major rebranding and redesign of the sites is expected in early 2004 with a transition to a UK centrally hosted server.

6.2 Montage Requirements – Phase 1

6.2.1 Project Scope – Montage Website

In 1998 the author began work with the British Council on a collaborative project between the Australian Broadcasting Corporation and the British Council. From this early project emerged the first specification for the Montage website. This was developed on commercial terms by the author for the British Council. Their objective in the first instance was to improve the internet-based access by Schools to Projects and Kits facilitated by the British Council. The British Council also facilitated partnering arrangements between schools of like interests, largely on a manual basis. The project deliverables at this stage were:

- Development of web-based school registration with automatic registration in a List Server when schools join a project.
- A site and list serve search engine for inter-school communication
- A moderated feedback page.
- Intranet facilities to add/amend/delete all database elements.

The essentials of the school registration process were to allow:

> "1. Schools to register with Montage and be automatically joined to the Montage teacher list serve.
>
> 2. Schools to register with a project and be automatically joined to a teacher project list serve and a student project list serve. Each project had two list serves which will be for teacher professional development and student project activities. There were 20 projects by the end of 1998.

3. Schools to register for Schools "Link Up" where schools could search for partners having similar curriculum interests. Schools would automatically be joined to the Montage list serve." (British Council Australia, 1999)

The O'Reilly's WebBoard software was chosen for integration with the Montage Website. The architecture for content serving was Internet Information Services.

At this stage a simple Content Management System was deployed (as described in Chapter Five) to facilitate management of projects and registrations, although not specifically at this point directed to content reuse. Their architectural requirements were for NT-compatible database and web serving design with an intranet capable of a limited web site publishing workflow allowing them to manage all aspects of the school and project content for the montage site.

6.2.2 Project data requirements

The initial requirements for project description were minimal – sufficient to describe the project for online access and registration. Table 6.1 has the initial data requirements supplied for the project.

PROJECT INFORMATION	SCHOOL INFORMATION
Field name	Name of School/College
Name of School/College	Country (Controlled vocabulary selection list)
Town/City	Contact person name
State/County/Province/Region	Contact email address
Country (Controlled vocabulary selection list)	Checkbox: Do you want a Montage kit?
Contact person name	Field name
Contact email address	Address
URL of the web site: (Optional)	PO Box
Target Group (Controlled vocabulary selection list)	Town/City (free text entry)
Type of school/college (Controlled vocabulary selection list)	State/County/Province/Region
Select a project to join (Controlled vocabulary selection list)	Postal/ZIP Code

Table 6.1 Initial Montage data requirements

The school registration data requirements did not at that stage differentiate between a "teacher" and a "school", a point of later development.

Once the registration form was completed the school was automatically joined to the project list serve for the teacher and a list serve for the students. An email registration acknowledgement was sent to the teacher and the project co-ordinator.

6.3 MontageWorld Requirements – Phase Two

6.3.1 Project Scope – Content Reuse

The Montage concept proved popular among schools in Australia (see Table 6.4 below) and soon had a growing international following. The limitations of content management at this point were clearly emergent – with a growing demand for country-based branding of content selections, and greater flexibility in the definition and description of content across the website. There was also an emergent demand for syndication of project content.

In 1999 the author presented a proposal to the British Council toward the development of a Content Management System focussed on content reuse. This proposal, an extended field test of the proposed new model for content reuse, was to deliver the new Montage website structure branded under the banner "Montage World". The objective in the first year was to implement websites for Japan, Singapore, Malaysia, USA, Canada and Australia, although Canada and the USA were to be simple home pages with contact details. Subsequent implementations included Finland, Indonesia and England, although budget constraints are likely to limit further implementations.

The services provided in development of the prototype included:

- Development of the new Content Management System application to support Montage publishing.
- Implementation services to the equivalent of approximately 2 days per week, providing support and guidance in:
- Web site statistics
- Existing site ad-hoc support (currently covered by quarterly systems administration fees)
- Site implementation in each country
- Software configuration and annual licensing for each country implemented, comprising:
- Documentation for the publishing software, tailored for the British Council procedures

6.3.2 Publishing Server

In order to facilitate the prototype, the author supplied, installed and managed the server that was to provide the Content Management Service software. The British Council provided the hosting location and networking for the server at their premises. This server provided the central database for British Council Australia publishing activities. The British Council funded the cost of ongoing System Administration of the server.

6.3.3 Publishing Software

Central to the Montage World implementation was the implementation of a content replication approach that allowed the generation of a new country site, including layout generation and content replication/reuse. A key objective of the CMS in this context was the replication and reuse of documents (web pages, and potentially multiple other document formats), images and metadata to be stored in a database through a simple user interface. Separation of content from presentation design, and fragmentation of content for reuse were fundamental to the proposed site replication concept.Design and presentation aspects were to be handled by web designers who set

up the appearance and functionality of the site with HTML templates. After a site has been set up and handed over it can be updated simply and consistently by pasting text into entry forms and single button clicks to preview and publish. Navigational index documents were to be automatically rebuilt and metatags inserted to facilitate the operation of search engines. The separation of content also had the objective of facilitating content reuse and delivery in new technological formats (for example RSS XML syndication).

This Content Management System prototype was, of course, intended to demonstrate more than simple website and content editing, and the proposal to the British Council was directed to:

- A generalised import engine capable to facilitate database capture of existing site content (images and text).
- Embedded XML handling and XSL publishing capability.
- Digital asset management facilities.
- Inherent multi-lingual capabilities.
- Multiple site publishing and improvements in content reuse.
- Variations in information design to provide a flexible document ontology allowing complex document inter-relationships.
- Providing access paths to core information content can were to be maximised through the combined use of generalised search engines and structured index paths to information on a site or in a publication produced from the system.

At the time of project initiation, there were some (very expensive) tools for content management in the market, but few of them took a long-term view of content management in terms of a library of digital assets, to facilitate content reuse. The British Council agreed to fund and support the project.

6.3.4 Design aspects of the MontageWorld website

The precursor to MontageWorld, the MontagePlus website, had functional elements which allow schools to register:

- As discrete entity themselves.
- Optionally subscribe to Projects and to register which are managed by the school.
- Register their interest in another school's (or several other schools') specific projects.
- Register their interest in other school projects by way of subject groupings.
- Register their Interest in other school projects by way of age groupings.

An additional feature of these registrations was the provision of simple email links from school to school. At time of registration for any of these purposes, a school nominates a contact person (usually a teacher) to act as a contact point for that school.

Over 4000 schools registered with Montage Plus by the conclusion of the prototype trial stage. MontagePlus acted as a central point of registration for schools. As new Montage country sites appeared (for example Singapore, New Zealand, Malaysia, USA and Canada), school lists were tailored for both local requirements and global listings.

One of the key reasons that registration of schools was so successful was the ease of the registration process that did not demand an arduous requirement for many items of information. With a growing volume of registrations, several issues associated with duplication could lead to degradation of the value of the database:

- Recording of different forms of the same school name: for example “Manly High School” and “Manly High”;
- Recording of school several times to increase the number of contacts at a school (that is, the contact at a school may wish to register an interest in a set of projects. However another potential contact may register in a separate set of projects)
- A school might register several times with several discrete projects (via say add a project).

The XML content structure for schools was therefore, in 2001, normalised to allow a single school registration and multiple teacher registrations for the school. This allowed more targeted mailouts according to interest area.

The school registration process was extended to provide a complex inter-relationship managed by a single registration script (web-based but published across different sites from a single script definition in Inter-Publish).

6.3 Content Reuse

Content reuse was explicitly realised in the Montage project at many levels of the conceptual model, including the overall website design, specific page templates, reusable scripting components and multimedia content items used across the sites. A template pre-processing language was developed to support the publishing process in Inter-Publish. This pre-generative language provided a high-level meta-language for content inclusion and cascading template design. This could be used both for primary generation of published content and as a precursor to later XSL processing. In the Montage context generalised templates were designed for a "white site" which used localised cascading templates to describe country specific content. Figure 6.1 below has an HTML snippet illustrating such a cascading template. The <<Include>> metatag allows the hierarchical acyclic inclusion of templates or documents within other templates or documents from within the current content section or from standard templates or documents visible to all sections. A “white site” was established using standardised content inclusions to provide effective reuse of the design layouts as well as informational content.

```
<table class="Montagetable" width="770" border="0" cellspacing="0" cellpadding="0"
align="center">
    <tr>
        <td colspan="3" valign="top">
        <<Include "Top Navigation">></td>
    </tr>
```

Figure 6.1: Hierarchical template structures

Standardisation of website scripting components was also achieved through cascading inheritance of templates (for example the construct "<<Include 'Standard Montage Search Script'>>", allowing definition of a single ASP snippet used across pages when published). Standardised content (such as school registration information) could be published in XML to support reuse within the runtime engine, or included in final generative output in "E-Zines" mailouts.

Finally, information resources, such as an "internet teachers' guide" were compiled and distributed to sites that wished to include the resource, once again branded by the country template.

6.4 Montage Project Outcomes

Montage has had considerable success across regional and metropolitan Australia with a program that has now been expanded to 5 countries in SE Asia as well as Argentina, England, and Finland. The program now also has a portal in the British Council UK called MontageWorld. There have been opportunities for teachers and students to travel to other countries as part of the programme. The following is a summarised report from Inter-Publish of the schools registered in 133 countries around the world (with over 5000 schools currently registered). The reach of Montage is illustrated in Table 6.2, which presents the countries with schools participating as at end of 2002.

The Montage service has considerable significance in networking Australian schools with the UK and internationally, establishing positive cultural relationships between teachers and educational departments. Some of the projects have had political significance. The outcome of one project called *Mosaic* is now hanging on the wall of Stadium Australia in Homebush and is part of the Visitor Tour of the venue. The Education Minister for NSW, John Aquilina, opened this Mosaic in 2000. There is also a Mosaic laid in the entrance to the Millennium Rugby stadium in Cardiff that was produced by students from Meadow Flat Primary School in Bathurst, NSW.

Country	#	Country	#	Country	#	Country	#
Afghanistan	3	Central African Republic	1	Ireland	31	Reunion	2
Albania	9	Chile	13	Israel	21	Romania	48
Algeria	9	China	27	Italy	66	Russia	22
American Samoa	1	Colombia	16	Jamaica	2	Saudi Arabia	1
Andorra	1	Congo	1	Japan	18	Scotland	41
Angola	1	Congo, Democratic Republic of	1	Kazakhstan	2	Sierra Leone	1
Antigua & Barbuda	1	Croatia	13	Kenya	3	Singapore	19
Argentina	22	Cyprus	3	Kyrgyzstan	1	Slovakia	3
Armenia	4	Czech Republic	7	Latvia	1	Slovenia	5
Aruba	1	Denmark	13	Lithuania	15	South Africa	31
Australia	2180	Ecuador	1	Macau	1	South Korea	15
Austria	2	Egypt	6	Malaysia	26	Spain	25
Azerbaijan	3	England	636	Malta	2	Sri Lanka	2
Bahamas, The	1	Eritrea	1	Mauritania	1	Sweden	11
Bahrain	2	Estonia	17	Mauritius	1	Switzerland	3
Baker Island	1	Ethiopia	1	Mexico	10	Taiwan	17
Bangladesh	1	Faroe Islands	1	Morocco	4	Tanzania	1
Barbados	3	Fiji	2	Namibia	1	Thailand	28
Belgium	5	Finland	5	Nepal	2	Tunisia	2
Benin	2	France	39	Netherlands, The	4	Turkey	20
Bolivia	1	Gambia, The	1	New Zealand	85	Turks & Caicos Is	2
Bosnia & Herzegovina	1	Georgia	1	Nigeria	11	Uganda	4
Botswana	3	Germany	37	Northern Ireland	16	Ukraine	13
Brazil	44	Ghana	8	Norway	8	United Arab Emirates	3
British Indian Ocean Territory	1	Greece	13	Oman	1	United Kingdom	16
British Virgin Islands	1	Guam	1	Pakistan	26	United States of America	634
Brunei	2	Guatemala	4	Palestinian Territories	2	Uruguay	1
Bulgaria	8	Haiti	1	Panama	2	Venezuela	4
Burkina Faso	1	Hong Kong	45	Peru	7	Vietnam	7
Cambodia	1	Hungary	7	Philippines, The	10	Wales	150
Cameroon	1	India	48	Poland	10	Yemen	3
Canada	75	Indonesia	115	Portugal	20	Zambia	3
Cape Verde	1	Iran	5	Qatar	1	Zimbabwe	18

Table 6.2: Countries with participating schools

The programme is open to all schools across Australia and actively involves schools from remote locations. In other respects also, the Montage project has been a testing ground for innovative applications that have been used later in other areas. For example the online mock judicial trial software that was implemented by the ABC within Montage is now being investigated as a method for arbitrating small claims for clients in remote locations in Australia. Projects arising out of Montage, such as the Baroque Online website, developed through collaboration between the Department of Education, Musica Viva and the British Council have achieved awards for their educational services.

The next chapter presents a detailed evaluation of the Inter-Publish prototype and its ramifications for content reuse.

Chapter 7 – Evaluation

Chapter Three presented the research methodology that was chosen to evaluate the Content Model for Reuse (CMR) presented in Chapter Four. The development of this model was underpinned by a comprehensive literature review from which was derived a set of principles for systematic content reuse. Chapter Five described the Software Development Research prototype and Chapter Six the field test used to explore the model. This chapter presents and evaluation of the Content Model for Reuse through a quantitative and qualitative evaluation of the prototype implementation.

The Inter-Publish experimental prototype was developed in the context of the proposed model and field tested in a challenging web-publishing environment. The prototyping process was an important element of the Software Development Research process, providing an environment for continuous improvement in the theoretical elements of the model through a rich and complex field implementation.

Chapter Five identifies a set of evaluative criteria elicited from the literature and industry review that are explicitly oriented to content regeneration and reuse. These criteria are directed to each layer of the model for evaluation of the effectiveness of implementation specific to the goals of the conceptual model. Section 7.1 presents a qualitative evaluation of the prototype against these criteria to measure the degree to which the model was effectively realised in the context of the prototype developed for the field implementation. Section 7.2 presents a quantitative evaluation of the prototype against metrics for information currency and reuse based on data gathered during the process of the research over three years.

This multi-methodological approach to Software Development Research is intended to enrich the insights arising from the research into the effectiveness of the theoretical (Nunamaker, Chen and Purdin, 1991).

The advantage offered by a substantial field trial (as against a lab experiment) is the significant opportunity for exploring the external validity of the model in a demanding publishing environment. While this approach has inherent weaknesses (in the degree to which the internal validity of the model is established), the scope of the field trial was significant for achieving stronger validation of the external aspects of the model than would be possible through laboratory-based experimentation. The versioning element of the model provided valuable data for detailed analysis of the changes in content publishing and reuse over the duration of the project. In Section 7.2 generalised metrics are proposed for the evaluation of currency and reuse and presents a quantitative evaluation of the outcomes of the field trial in terms of these metrics for evaluation of content turnover in a website publishing environment. This section also explores the ways in which such a field implementation can have complexities arising from external events that affect the trial.

Finally, as part of the post-implementation review of the project, a usage and usability study was conducted with British Council authors that have used the Inter-publish system. This usability study is analysed descriptively (on a small population) in order to give an indication of the end-user perceptions of the effectiveness of the prototype to its purpose.

7.1 Qualitative Evaluation of Reuse against the Model

7.1.1 Content Reuse Outcomes from the Field Trial

The Inter-Publish prototype was the outcome of a sustained effort to implement the four layers of the model. The result was a Content Management System with a flexible ontology highly suited to reuse of heterogeneous media forms. The relational design for the prototype tested on both high-end (Microsoft SQL Server) and low-end (Microsoft Access) relational database environments. While the prototype went through many changes through the three-year life of the project, few of them related to the relational model itself. This relational design proved

suitable for the management of fragmented content at layer one of the model and realisation of layer two of the model for complex document structures. The workflow user interface changed frequency during the duration of the project, particularly in relation to the visualisation and navigation of fragmented content.

The prototype architecture proved scalable, handling over 25,000 documents for the Montage service alone by the end of the three-year period (representing more than 90,000 Economic Content Units). This content collection comprised a mixture of inbound School and Project registrations moderated and accepted in Inter-Publish (and then republished as website content), directly authored content supplied by British Council staff, and content through ODBC and Web Services interfaces through Web Services from other systems.

At the cut-off for this analysis of the field trial in December 2003 the British Council Australia had expanded from their original two web sites to manage:

- http: //ww.montageplus.co.uk - primary and secondary school resources expanded from the original www.montage.edu.au and now comprising nine separate and independent country-specific sites for Indonesia, Thailand, Singapore, Australia, NZ, Finland, England, Hong Kong and Japan.
- http: //www.montageplus.co.uk/commonlink - a public information website developed in parallel with the CHOGM conference.
- http: //www.e-linkschools.org - a multilingual school resource database for information sharing between Japanese and UK schools
- http://www.keepintouchuk.net - a multilingual alumni resource database for information sharing between Japanese and UK schools
- http: //www.britishcouncil.org.au - the corporate website of the British Council Australia, now including various scholarship search resources and an education information gateway, and an alumni service.

In the initial stages of the Montage project, content reuse was focussed on the efficient use of web, school and project content and the separation of content from design. Syndication with a number of affiliated organisations was implemented. Content was captured using the Inter-

Publish prototype with template-based publishing in web (HTML), email (text and HTML) and XML form. With the international growth of the service, country-specific branding was required for the Montage site. A conceptual model for each country was required which would provide a framework for building country specific sites branded specifically for the particular country while drawing on the common resources.

Layer	Outcome
Layer 4 – Content Transformation and Delivery	ECU content was published to individual country sites and represented with a national view in the MontageWorld website. This applied to both project and school information. Content was delivered in a number of different forms – through E-Zines (email mail outs), traditional web publishing and syndication to other educational institutions. E-Zine delivery was personalised based on region and interest areas. Generation of RSS syndication feeds The automatic creation of Image credits pages.
Layer 3 – Content workflow	A Web Services layer provided content interchange with the existing ObTree CMS for purposes of content exchange between these systems. The web services layer was also used for content integration with outlook and other sources. The use of the ECU structure supported two types of content reuse: The reuse of the same content across different layouts and websites The reuse of the same layout template across different sites, by defining generic template forms common to all sites (that is, reusing a template for the generic form of the page, with over-riding localised templates for specific instances – for example site-specific navigation elements). This reuse capability was consolidated at the website level, particularly with template inheritance, to support reuse of generic website designs for site replication.
Layer 2 – Document models	Site replication facilities were used to rapidly replicate both the site layout and internal content, providing an effective means for rapidly bringing a new Montage country site to a point ready for publishing. Document models were reused in order to establish complex workflow relationships while maintaining consistent underlying ECU content.
Layer 1 – ECU	The reuse of content in several different forms was facilitated through the ECU fragmentation of content. ECU content was also selectively replicated at the web services layer.

Table 7.1: Summary of content reuse at each layer

Content reuse was the enabler for such a rapid rate of expansion of website content without major staffing increases. This content reuse was expressed in several ways:

- The reuse of "model" white sites replicated through the CMS prototype to form a base for building new Montage country sites, the details of which are described in Chapter Six.

- The reuse of common structural layout elements throughout the sites (such as site navigation).
- The reuse of informational content across multiple sites (for instance the teachers education guide) repackaged for the particular site context.
- The reuse of informational content across generational changes of the site design (the Montage site was re-branded three times in the duration of the project with substantial retention of informational content across the generational changes of site design.
- Use of information content for alternative publishing activities (for example email newsletters) and content reports.

Table 7.1 presents a summary of major outcomes at each layer of the model. Other examples of metadata reuse include the generation for some sites of image credits pages.

7.1.2 Evaluation against the CMR layer 1

The Economic Content Unit (ECU) is the underlying unit of information within the CMR. A hybrid model of relational database storage and XML encoding of fragments was chosen, yielding an architecture that can be readily implemented in most client environments. The ECU was also described and supported through the web services XML schema presented in Appendix Two. Table 7.2 (below) examines reuse against the measures presented in Chapter Five at this layer of the model. The ECU provided the essential element in the management of heterogenous content resources. Some content (such as PDF documents) was managed at a coarsely granular or "chunky" manner at the whole document level. Text content was managed at times in a finely granular manner (for instance for user registration information) and at times in a coarsely granular manner (for instance in managing as a single content unit the text content of an informational web page). Multimedia resources of a variety of sorts were managed through the field trial with both language variants (for British Council Japan) and image encoding variants (Japanese/English navigational images) implemented

within the ECU framework, although there was generally little use of the capability for management of image encoding variants.

Layer 1	Achieved in the prototype
Content reuse requires the fragmentation of content in a manner that allows the separation of content from its final presentational form	The prototype supported the fragmentation of content and the definition of abstract templates supporting hierarchical nesting for references to other content and other templates and use by a complex transformation language.
Support for content in many languages	The prototype was designed from the beginning with Unicode and Double byte character support in mind. While both relational database platforms chosen support native Unicode and double byte character encoding, the prototype was designed to allow mime-encoded storage of characters where such native support is not available. Image and other multimedia objects were also stored in this way.
Internationalisation of user interface	Terminology used in the browser-based user interface was itself registered and managed as content in the prototype, simplifying the internationalisation of the user interface. Such tailoring was generally, however, used for terminological normalisation to meet the needs of particular offices, as most users in the British Council country offices worked in English when authoring using the browser User Interface.
Multi-lingual content variants	Content could be managed across multilingual variants, although the user interface for comparative language editing would benefit from design improvements. Automatic translation capabilities would add value to the user interface presentation of multi-lingual objects.
Temporal Versioning	The ability to version content over time, and recover/retrieve/review variant versions was implemented early in the prototype, and proved profoundly valuable in measuring content reuse over time.
Encoding variants	The Economic Content Unit is the principle content structure, supporting the management of digital content across encoding and language variants. The ECU is realised both through a relational schema and an XML schema. Language variants were used in a number of Montage projects. Text content was used at a fine level of granularity in many cases, but multimedia encoded documents were generally managed at a coarse granularity, and users generally could not be enticed from the separate registration of image variants. A more generalised support for encoding variants requires better integration of multimedia support in the user interface design.
Content discovery, and semantic information that facilitates this discovery, is an essential element of content reuse.	Content discovery was supported within the application through a number of content visualisation and search strategies. A complex search engine supporting persistent search definitions was implemented. This search interface (collections) was also supported for content exchange through Web Services and the XML RPC interface to the content. Content could be shared in fragmented form or transformed to different document ontologies.

Table 7.2: Evaluation against the CMR Layer 1

7.1.3 Evaluation against the CMR layer 2

The document model for harnessing fragmented ECU content proved sufficient to realise both simple web page content and complex data relationships. The specific table design

supporting the content fragmentation was manageable, with large collections of schools, teachers, links and multimedia resources. The document ontology was flexible to definition of standard relational-style document relationships (for instance schools having multiple teachers and teachers having multiple contact details) while being flexible in accommodating complex document structures. The linear fragmentation of ECU content was overlaid at this layer with the document model tailored for the current authoring and workflow processes. The database model chosen to support the document ontology was suitable to support large collections of multimedia documents, but could be enhanced to provide more efficient text search across content resources. Table 7.3 examines the achievements of the prototype against the criteria elicited in Chapter 5 for reuse.

Layer 2	Achieved in the prototype
A formal separation of content from stylistic form is implemented, with Template-based and XSL transformation structures to yield published content	Formal content separation from structural and stylistic elements is achieved in the prototype in two ways – through a relational model that supports multimedia Economic Content Units and associated metadata and an ontology for relating the ECU content in variable document structures. This content separation is also described through an XML schema used in a web services model. Forms-based or XML-based content separation was available in a flexible document ontology.
A flexible ontology that supports the definition of document types amenable to simple change and expressing complex content elements (Economic Content Units)	The ability to create on-the fly document structures without database changes, and to describe complex document hierarchies was achieved in the relational schema and XML web services schema implemented at this layer of the prototype. The document ontology itself proved flexible for reuse across projects and content could be mapped across document types. An ontology "editor" was developed and facility for pre-populating the ontological structure to match particular publishing environments.
Ontology versioning	Where the document object model has changed over time, historical content can be viewed in the prototype in the context of the original or current document ontology – however versioning at this layer of the document model was not extensively explored. A better approach to ontological versional would involve the proper mapping of content to variant document structures over time.
Content can be used outside the particular web page context in which it is published	ECU content could be selected through both static and dynamic selection rules for participation in other content forms. The Inter-Publish transformation Language provided a framework for template-based transformation of content to final encoding forms or to other document ontologies.
The ability to create content collections for use by reporting, indexing and syndication engines for transformation and distribution of content	A report writer, transformation service and content generator was created that can support all media formats. Web Services and ODBC import functions provided a means of mapping existing content objects to the CMR framework.

Support commonly used document structures, and validation against those structures	XML resource documents (for website use), SMIL multimedia document generation and RSS syndication documents, among other forms, were generated as part of the Montage project.
Document ontology can itself be documented or is self documenting	The ability to version content over time, and recover/retrieve/review variant versions was fundamental to the prototype design.
Generalised database support for content storage	Support for Microsoft Access and Microsoft SQL server - with most relational queries designed to ANSI standards except where mandated by the specific database. Web services were supported through a structured XML schema describing the document ontology and content forms.

Table 7.3: Evaluation against the CMR Layer 2

7.1.4 Evaluation against the CMR layer 3

The third layer of the model, the workflow layer, was the point of interface with content authors and authoring systems. Content was captured both directly through the browser-based interface and indirectly through the web services interface and through ODBC and other content capture methods. In some cases Inter-Publish formed the principal repository of content, while in others it was itself part of the continuous cycle of content transformation for ephemeral content served through the British Council web servers.

Security configuration for the prototype presented difficulties during relocation of the software to the UK in March 2003. The service was initially hosted using SSL (Secure Sockets Layer) and public access, but finally a VPN (Virtual Private Network) architecture was implemented by the UK team for access to the service.

There are many elements to the workflow process that are contingent on the particular context in which a service is implemented. The criteria in Table 7.4 below include the elements for workflow that are either integral to effective authoring (and therefore axiomatic to content reuse) or which extend the workflow to enhance systematic content regeneration and reuse.

Layer 3	Achieved in the prototype
Content editors can be defined with different levels of content visibility	A multi-layer user security workflow was required for the project, supporting user-based privileges mapped against content organised by section and document type.
Editorial constraints on modification to templates	Content editors could be classified on a content-section basis into categories of "editors", "designers" and those with read only access to the content.
Content can be discovered through transient and persistent collections.	Transient content search and discovery facilities were developed in conjunction with complex persistent document Collections. These Collections could be used in all aspects of content management and publishing.
Multi-lingual publishing	The ability to publish language variants of pages in a language-based template driven manner was implemented through the Inter-Publish Transformation Language.
Temporal Versioning	The prototype implemented the ability to version documents over time, and recover/retrieve/review variant versions.
Encoding variants	The document ontology supports variable encoding of content and the workflow system supports heterogenous content forms through a variety of public interfaces - including a virtual FTP view of the content and WebDAV support. This was principally used for multi-lingual rather than multimedia encoding management.
Workflow routing capability	Documents could be routed with alert via email to targeted individuals or group email addresses or by translation to different document types. Content visibility could be determined by document type, permitting selective visibility as a means of routing.
Establishment of workflow groups	Simple email groups are supported in the workflow ontology. There was very little collaborative authoring in the target community, where a single (fractional) resource was responsible in each country for their own content management. A more flexible implementation would involve group-based privileges and group memberships – not implemented in the prototype.
Performance metrics	The raw data to develop performance metrics are available through versioning and publishing metadata, and are analysed below in terms of content turnover and information turnover (see Section 7.2 below).
Queue management	There are capabilities for mass queue management but these are not explicitly directed to workflow management. The Content Management features allow reallocation of all aspects of document status, and en-masse management of changes to content, and includes by virtue of this the queue status.
Management of events	Events are driven by "publishing" and "unpublishing" actions by the user. Linear workflow events are driven by such actions. The web services interface provides the option for non-linear event actions.
Authoring Workload	The versioning history provides the means for measuring in detail the workload in the workflow system, as does the "check in"/"check out" as a measure of currently active documents. However, no workload reports have yet been designed, nor has a workflow management console.
Monitoring	Monitoring of workflow activity was limited, but satisfactory to the target field test environment. Principal monitoring of events was through email alerts at the point of a particular event transition, and through summary information showing outstanding counts of moderation and pending action items.
Simulation	Workflow processes could be simulated but there are no automatic content generators to visualise and assess this process in the prototype. A workflow visualisation would be an attractive extension to facilitate productisation of the prototype.
Handling of forms	Complex management of form transition is possible through the flexible document ontology.
Webflow	Complex transition of documents in a "web flow" environment is supported, and is inherent in the browser-based design of the application. Furthermore, several options were included to facilitate the presentation of content transition in complex workflow structures.
Documentation	Documentation was directed specifically to the needs of the Montage service - see Appendix Three.

Table 7.4: Evaluation against the CMR Layer 3

7.1.5 Evaluation against the CMR layer 4

The essence of systematic content reuse lies not in a single document ontology nor in a particular markup approach, but in the establishment of systems that facilitate a process of continuous transformation. The prototype environment served both as a primary content repository and as a vehicle for transient content transformation from other database systems. This process of continual content transformation is supported at this layer through the underlying document ontology and ECU elemental content structure, the workflow processes for content capture and finally the transformation and delivery vehicles of the publishing agent at this layer of the model. This was perhaps the most demanding element of the field trial, as content publishing included elements of publishing in multiple encoding formats and bringing together content not only for primary document generation but for complex document structures and from dynamic and static content collections. Table 7.5 presents an evaluation of the field test achievements against the criteria elicited in Chapter 5 against this layer of the model.

Layer 4	Achieved in the prototype
Primary generation of web pages	The prototype could publish web content in HTML, XML and other formats, dynamically or on demand. It used a lightweight transformation language - the Inter-Publish Transformation Language (IPTL) for such primary page generation.
Syndication to varying document ontologies	There was support for NewsML, RDF, RSS, PRISM, SCORM, SMIL, with specific publishing of content in RSS and SMIL in the test environment. Generation of different document ontologies was supported through persistent and dynamic collections and the Inter-Publish Transformation Language in conjunction with the underlying layer-2 flexible document ontology.
Ability to support alternative transformation approaches	There was support for secondary XSLT transformation of content generated initially through the IPTL transformation process to generate the initial XML and XSLT documents required for such transformation.
Website versioning	Editioning of an entire website would be possible through the ECU and versioning infrastructure, but was not a requirement for the target community.
Flexible delivery agency	A flexible delivery agent was developed with capability for retry and a scheduling agency for time delivery
Personalisation	Content reuse and delivery based on the known interest areas of content subscribers was provided through the Indexing capability and the Inter-Publish Transformation Language. This allowed selective dissemination of information by email targeted to known demographics and interest areas. There was also some minor personalisation of information through scripting on the web server.

Table 7.5: Evaluation against the CMR Layer 4

7.1.6 Completeness of the Prototype against the Content Model for Reuse

The credibility of the Content Model for Reuse rests on the degree to which it can be implemented, and the degree to which implementing the model realistically demonstrates benefits in the purpose of the model. The layered approach to content reuse proved an effective design approach to facilitate many aspects of content reuse. The Inter-Publish prototype represented a coherent and credible realisation of the Content Model for Reuse, and the environment in which it was situated, of rapid content growth and continuous transformation of both ephemeral and strategic content resources across multiple web sites, proved an effective environment to explore issues at each layer of the model.

7.2 Quantitative evaluation of Reuse Analysis

This section presents a quantitative evaluation of the prototype using the content versioning and publishing metadata history for the period of the project. From this data it is possible to determine the differential aspects of original publishing and content reuse. The purpose of this analysis is to elicit the ways in which the software facilitated the reuse of content, and the ways in which content reuse changed over the duration of the project. The results are indicative of the value of the prototype and the model in content reuse.

7.2.1 Resources used in the Reuse Analysis

The following analysis was prepared after the Inter-Publish prototype had been used in the Montage project for 3 years. The versioning of content at each layer of the conceptual model, being a fundamental part of the ECU data design, was implemented in the first generation of the prototype. This continuous, three-year log of authoring and publishing metadata yielded valuable information on the nature of content authoring for the project, and how this content was deployed and reused. As discussed in Chapter Six, the Montage project represented multiple website domains sharing common resources but with individual branding. Each of these domains was

published from the central CMS server. This server also served as the global clearinghouse for all Montage country sites**.** The Montage website acted as an educational project and information resource, directly hosting projects and acting as a gateway for primary and secondary school teachers to external projects. The domain saw several name changes and design makeovers in the last two years: from www.montage.edu.au to www.montageplus.co.uk, and more recently to www.montageworld.co.uk.

The Montage project started in 1999, and from mid-2000 onward, the Inter-Publish prototype CMS was used for content management of Montage web sites. External website repositories and logs are combined with Inter-Publish ECU versioning and publishing metadata to provide a rich picture of authoring and publishing activity for the period from 2001 through to 2003. While the versioning changes were directed toward meeting the requirements of the emergent CMR model, the downstream consequence of this change has been the collection of a valuable history of content as it was created and deployed. This detailed version history also represents a rich data mining resource for the purpose of analysis of the impact of the content management software use over the three-year period analysed. The versioning table of the Content Management System has a snapshot of all content changes where some element of a content item has been changed. Publishing metadata and persistent content collections provided a source over time for tracking original content use and content reuse.

7.2.2 Website Content Profiles

Table 7.6 is a summary of publicly visible web page content categorised by type for the two principal domains managed by the British Council Australia, as at December 2000, 2001 and 2002 and 2003. By 2003 the number of published web pages had nearly tripled. This change represented a large number of school registrations and several new Montage country sites coming online during the year.

	Total Docs	Web Pages	Images (gif)	Images (jpg)	Video	PDF	Other (flash, director, etc)
2001	20597	14620	4664	1248	10	2	28
2002	28365	18820	7094	2238	179	4	30
2003	33699	24980	7458	1015	179	9	58

Table 7.6: Published website content summary for Montage sites

The following is a summary of content authoring activity for these sites over the period 2001, 2002 and 2003 representing text-like (HTML, PDF, Word) content items changed on a monthly basis and published or republished as original content (as distinct from content reused for syndication or redistribution). Table 7.7 gives an indication of the text authoring activity on a monthly basis.

A research project involving a field implementation is, of course, vulnerable to organisational changes that affect the trial. Several changes in direction in relation to the Montage project had an impact on the quantitative measures presented below. A makeover and regeneration of the Montage site, rebranded as "montageworld" occurred in February 2002, resulting in considerable additional content being added to the site. In 2003 the British Council undertook a reorganisation of the project organisation of the Montage project. Up to the end of 2002 management of the Montage initiative had been centred in Sydney (where the project originated). After an organisation review, the project was deemed to be sufficiently large to require management out of the London office. As part of this change, the publishing server was relocated to the UK. Occurring at a time of dramatic growth in the international reach of the Montage project, this organisational change had a significant impact on the operation of the publishing system. The affects of these changes are apparent in the quantitative analysis presented below. The Sydney team supporting Montage was largely disbanded and replaced by a management team in London. A subsequent relocation to Manchester, along with transfer of the Inter-Publish software to the UK resulted to a three month hiatus (April 2003 through to June 2003) during which most access to publishing activity was disabled (although the public website

remained active and external content submission continued). The figures presented below are therefore significantly affected by this major event. The subsequent site updates (publishing web pages, school and project entries and other resources that had not been updated during the hiatus) are apparent in the peaks of September and October 2003. The development of a custom web server itself required time-consuming development to address peculiarities of non-conformant proxy servers (the proxy server in Malaysia, for instance, generated a partial duplication of posts for all HTML form transactions – resolving this low-level technical problem causing a delay of several weeks before Malaysian staff could use the CMS).

The affect of the 2002 rebranding of the site and the 2003 organisational changes are apparent in the publishing history shown in Figure 7.1.

	2001		2002		2003	
Month	Edited[1]	Published[2]	Edited[1]	Published[2]	Edited	Published[2]
Jan	46	247	548	152	431	374
Feb	110	24	280	1799	954	421
Mar	599	467	366	774	495	270
Apr	255	188	690	521	159	55
May	306	205	593	247	176	38
June	622	417	419	97	1373	87
July	2517	486	203	109	256	178
Aug	753	625	3418	738	172	90
Sept	321	175	2738	759	5060	2701
Oct	117	28	1094	567	5201	2650
Nov	2227	552	1979	579	513	197
Dec	325	66	593	277	1474	633
	8198	3480	12921	6619	16264	7694

Table 7.7: Content authoring profile by month for 2001,2002 and 2003

1. The column "Edited" represents actual content changes for all informational content types (including images, user registration information). It excludes content types which are used by Inter-Publish itself (such as Word and Task) for purposes of user interface management. These figures represent actual editing work (as distinct from imported, automatically assembled or syndicated content).

2. The column "Published" represents the number of original pages "published" in the specified month (a document which has been created or changed and then finalised through at least one stage of workflow at least once during the month). It excludes content types which are used by Inter-Publish itself (such as Word and Task) for purposes of user interface internationalisation and syndicated content.

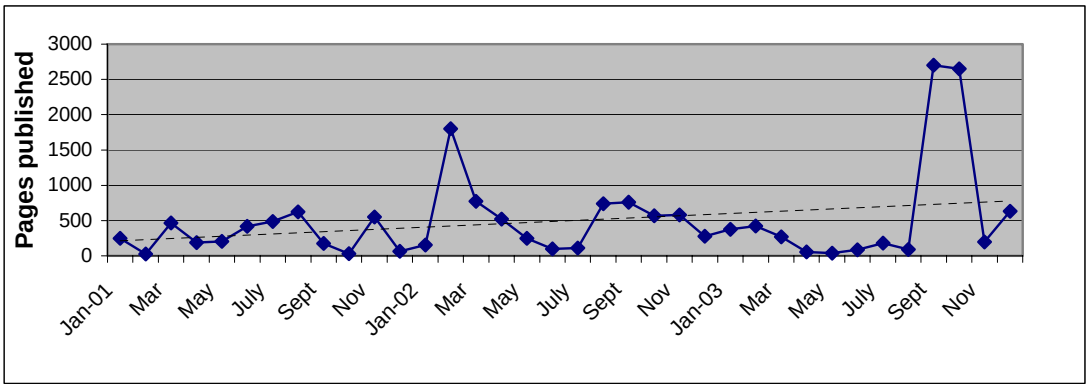


Figure 7.1: Original content published over the three-year field trial

7.2.3 Metrics measuring Content Currency

One important role of the Content Management System is to maintain currency of information on the website. The theoretical model for Economic Content Unit dissection of resources and versioning across all layers of the model enables the definition of generalised measures for website currency. In a retail environment, an important measure of efficiency in resource management is the "turnover ratio", a measure of how often stock is changed (or "turned") within a given year. As with traditional financial turnover calculations, the following measures look at the number of times content pages are refreshed as a ratio to the number of pages overall.

In quantitative terms it is possible to measure content turnover through an analysis of the versioning history of authoring changes and classification of content publishing into two categories: "original" content publishing (the first instance of use of an updated ECU) and the "refreshing" of content across the entire site.

The following two metrics are proposed as an indicator of information currency on a published website:

- *Information Turnover*. This ratio is intended to measure the degree to which actual content is changing excluding the effect of content refreshing and reuse. This measure is calculated from the number of documents for which ECU elements have been changed as a ratio of total published *primary* web pages (that

is, excluding navigation indexes and pages generated as secondary content resources). The primary web page is understood to be the first web page published from the parent document in which an ECU element participates.

- *Website Turnover*. This ratio includes the effect of content reuse in measuring the turnover of content on the website. The content turnover figure measures the number of web pages changed (through primary web page generation or content reuse) as a ratio of total published web pages.

The Information Turnover metric is useful as a measurement of original authoring activity. The Website Turnover metric provides a measure of the degree to which content reuse is contributing to currency of information in the website.

The Information Turnover is a measure of authoring activity and actual content change. Consider an example a website with 100 pages. An ECU is published to a single content page on the website but its title is referenced in two other web pages for navigational purposes. A modification to this ECU would result in an Information Turnover ratio of 1/100 while the Website Turnover would be 3/100. In principle the higher the turnover ratios, the better the indication of information currency in the website. The Information Turnover is a useful metric in measuring the authoring activity. The Website Turnover metric provides a measure of the degree to which content reuse is contributing to currency of information in the website.

These ratios are naturally contingent on the type of site that they measure. A large established website with a rich body of archived and static material may show low values. The measures can be considered reliable over time in a context where ECU and versioning concepts are implemented, and have validity as a measurement of change over time in the relative currency of a website. The versioning history of the Inter-Publish prototype offers a useful metric for evaluating the currency of information through measures of content turnover. Figure 7.2 presents the ratios for the period of the field trial.

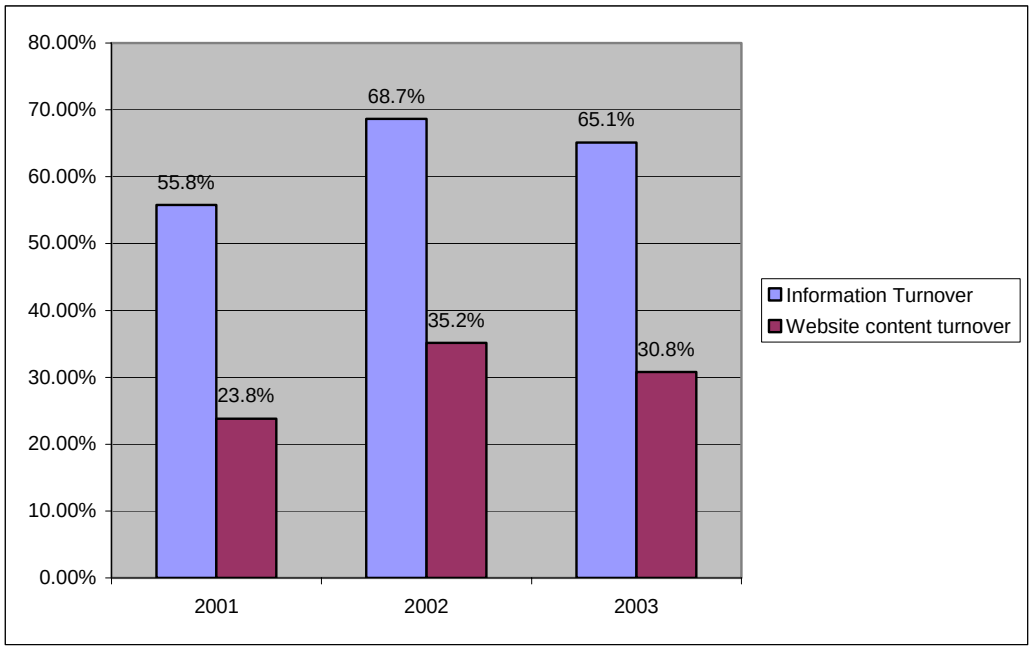


Figure 7.2: Website content turnover and Information Turnover

Similar metrics are not possible for the small, static site that the project started with in 2002. The slight decline in 2003 is indicative of the impact of the 3 month freeze in publishing due to organisation changes, although the net affect of this freeze hiatus on the yearly metric is clearly offset by the republishing "catch-up" in subsequent months. The improvement between 2001 and 2003 both in website Content Turnover and Information Turnover metrics, despite a substantially increased pool of published pages, is indicative of the benefits of operating within the proposed content management framework and of the effects of content reuse. They also suggest a sustained improvement in the currency of published text content. It should be noted that this growth in and greater currency of content was achieved despite static or declining staffing resources in relation to the project at different times of the British Council funding cycle for the project. For the first year of the project the British Council had one person working full time on content management and authoring. Thereafter the individual country sites undertook day-to-day content administration on an ad-hoc staffing basis. The workflow process was important in alerting staff as to content actions required.

In addition to the manual editing of web pages, a substantially larger number of pages were frequently updated as a result of scheduled publishing and syndicating activities of the Content Management System Inter-Publish. This aspect of content reuse is the subject of further quantitative evaluation in the next section.

7.2.4 Further Measurement of Content Reuse

As with many Content Management Systems, a key purpose of the CMS for the end users was the primary publishing of website content. The metrics presented above (Information Turnover and Website Content Turnover) illustrate the opportunity that the fragmentation and effective versioning of content presents in providing a comparative measure of original publishing against content reuse.

These indicative measures justify further analysis of the underlying contribution of original editing and reuse to the currency of information on the website. The versioning history provides a finely granular means for quantitative analysis of the nature of reuse within the field trial.

Versioning and metadata history permits the differentiation between:

- Primary content. ECU content that was attributed only to a single website location or was used once only in publishing (usually in the parent document to the ECU) was classified as "primary content.
- Reuse content - the content was used two or more times in publishing actual content (as distinct from metadata concerning the content). Metadata content reuse, such as automated site maps generation, was not included in this category.

Inter-Publish associates content units through a document definition called a “document type”. These "document types" were further classified into a "content type" to bring together documents of similar type. While many document types are pre-defined to accommodate the most frequently used content capture types, the prototype allows flexible user definition of other document type definitions. Table 7.8 below presents a dissection of the total number of published documents registered in Inter-Publish for the years 2001, 2002 and 2003 classified by "document type”. The

data collection was achieved through an analysis of a snapshot of ECU publishing and versioning history as at the end of 2001, 2002 and 2003 established as separate data sets.

Content growth from both internal and external editors was muted in 2003 due to problems associated with relocation of the Inter-Publish and external web servers to the UK. Security restrictions on web server access prevented external submission of project and image resources and internal access by content authors, compounded by relocation of the new support team from London to Manchester and the subsequent outsourcing of server support roles, all of which effectively prevented access by both internal content authors for a period of nearly 4 months – effectively two school terms. This principally affected publishing of content to the web server – content submissions in XML were accumulated during the publishing freeze until the location and configuration server was resolved. That is, while mechanisms for content reuse and syndication were still possible, the relocation and subsequent support issues had a detectible impact on publishing for the period April through to June.

Content Type	Role	2001	2002	2003
Artwork	Project-specific artwork objects		5	6
Article	Montage Australia news item			5
Bulletin	Montage Japan document		0	9
Calendar Event	Current events being promoted through the website		3	12
CSS	Site style sheets		60	23
Document	Web page content	600	762	792
Document No SiteMap	Web page content and scripts (not be included in automatic site maps	60	242	337
Email Index	Email mail out (e-Zines etc) – relocated in 2004 to another server	0	4	7
EcardImage	Image (commonlink project)	2	2	
Event	Current events being promoted through the website	2	60	4
ElinkSchook	Japanese Linkup School	214	189	198
ElinkSchool UK	UK Linkup School	75	62	73
EAlumn	Montage Australia image competition document			14
Feedback	User website feedback (republished to the websites)	122	149	923
Gif	GIF images	101	2119	2337

HTML	Web page content		128	146
Image	GIF/JPG encoded images.	597	1307	1970
Index	Content reuse indexes	108	281	390
JPG	JPG images	1	128	165
JS	Javascript		3	21
Layout	Design templates (used to define the presentational form of the published pages)	465	1130	1521
Link	Website link recommendations (republished as online portal-style website resources)	27	485	583
PDF	Portable Data Format			296
PPT	Powerpoint slides			272
Project	Project registrations (online curriculum projects)	105	244	1148
Registration	Japanese e-link registrations		140	700
School	School and teacher registrations	4182	5362	5633
SchoolPlus	School registrations		4420	4906
Script	Javascript and ASP scripts	41	186	534
Survey	Online surveys	5	7	8
Teacher	School and teacher registrations		4464	4770
TeachersGuide	Web page content (teachers guide)	52	176	176
Web Page	Web page content	146	291	221
XLS	Spreadsheet			274
TOTAL		6904	22409	28474

Table 7.8: Number of documents by document type, 2001 - 2003

By the end of 2001 there remained a large number of resources on the web server not yet reflected in the Content Management System (image and historical web page content), although all new content creation was through the CMS. There was progressively increased use of the CMS for image as well as text management through 2002. This trend continued in 2003, with inclusion of additional document types to meet specific publishing requirements and to facilitate particular types of content reuse (for example an HTML Teachers Guide). Teacher/School and project resources were largely sourced through online submission recommendations that were moderated and published.

The analysis in Table 7.9 presents a dissection of ECU content on the CMS as it was used for reuse. This was determined through an analysis of the ECU versioning history and publishing metadata for the years 2001 through to 2003. ECU content that was attributed only to a single

website location or publishing destination, or only participated in a site map role, was treated as "Primary" content. ECU content that has been used in two or more locations on the website (other than as metadata) or for two or more content presentational types has been classified as "Reuse Content".

An example of content reuse is the School content type. This content type represents a series of linked content units describing a school subscription to the Montage site. The content units are used in three ways:

- To populate the online search engine with the XML school profile for (a) searching and categorising schools on the public website, and (b) online maintenance and updates by the schools (schools were emailed a username and password on registration).
- To generate email lists for manual creation of mailouts.
- To generate system publishing of emails personalised through Inter-Published and with a strongly designed HTML layout.
- To research and report on school subscriptions within particular countries and regions.

Content within each document type was directed either to "primary content" or "reuse content". Table 7.9 below analyses the number of reuse instances for each document type in 2001, 2002 and 2003 by counting any use of an ECU for any publishing purpose. The high reuse levels of School and Teacher entries can be attributed to emails and could be considered a trivial reuse of the school entries, although e-Zines were personalised based on teacher and school region and interest. Personalisation of outbound communication was facilitated by the Inter-Publish Transformation Language and associated dynamic and static content selection rules, which could be scheduled to distribute updated website content in the form of content indexes, categorised web pages and outbound syndication of content (such as projects) in XML form.

Document Type	Reuse instances 2001	2002	2003	Average Reuse 2001	2002	2003	Notes
CalenderEvent	0	43	239	0.00	14.33	18.38	Events (new style)
CSS	0	2	7	0.00	2.00	3.50	Style Sheet
Document	1275	2417	30541	9.24	12.46	38.32	Web Pages and other text resources
Event	39	23	26	13.00	7.67	8.67	Events (old style)
Image	14	20	2375	2.00	2.00	7.64	Images (GIF/JPG)
Index	22	210	1042	3.67	8.75	11.20	Indexes
Link	6943	29655	24373	59.85	112.33	72.32	Link cross-references
Project	3318	18271	22570	15.08	37.36	60.84	Montage Project entries
School	9943	148244	15539	2.23	13.81	2.85	School entries
SchoolPlus	0	108693	125529	0.00	12.30	13.94	Split School/Teacher
Script	0	12	1456	0.00	2.00	22.06	ASP and Javascirpt
Teacher	0	116279	108270	0.00	13.02	11.88	Split School/Teacher
TeachersGuide	50	85	3232	0.47	3.86	15.32	Teachers Guide

Table 7.9: Content reuse by Document Type

Figure 7.3 below illustrates the way in which content reuse is favoured in certain Document Types. The average reuse represents the number of reuse instances divided by the total number of content units for that document type.

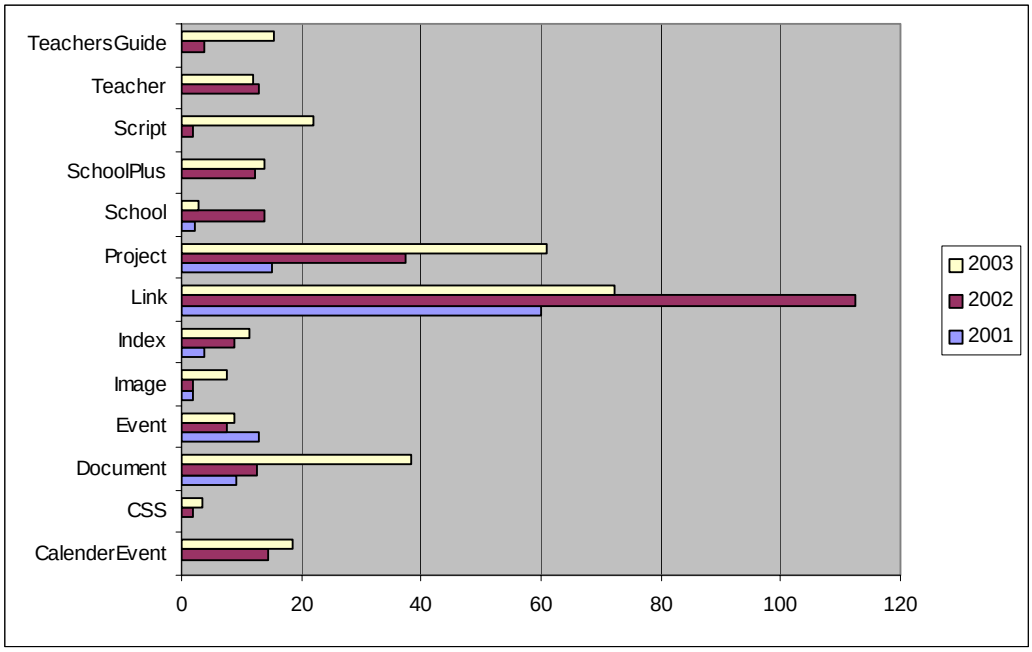


Figure 7.3: ECU reuse by Document Type

Figure 7.4 below presents a summary of the Web-based content reuse for the three years of the experiment (that is, excluding email mail-outs and E-Zines). The underlying reuse metric gives an indication of the progressively increasing reuse of content across the website.

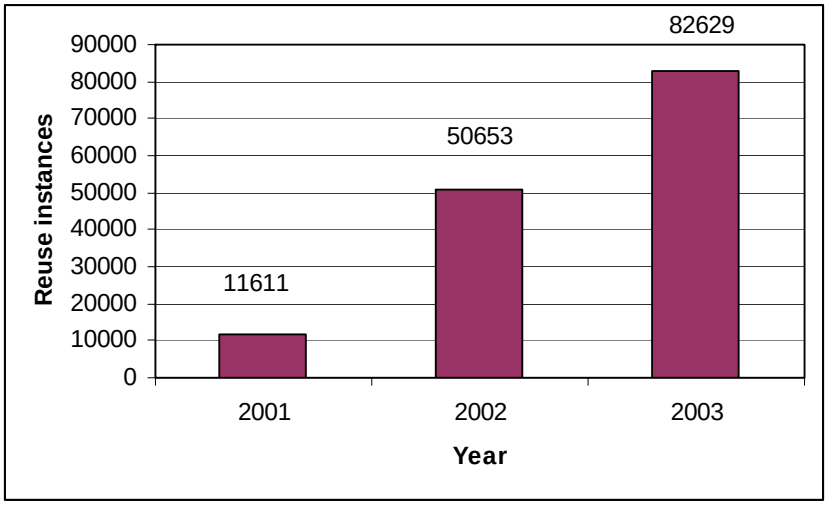


Figure 7.4: Summary of ECU reuse

The CMR model brings with it the capability to dissect not only publishing activity but also, through implicit versioning history, the behavioural changes over time of the content authors and their points of focus on primary content authoring and content reuse. The value of this content model is in the ability to define metrics for content currency and content reuse. The metrics relating to the Inter-Publish prototype indicate the efficacy of the model in maintaining content currency and in delivering both primary content publishing support and content reuse in the Montage project.

7.2.5 Focal points for content reuse

Flexibility of the document ontology emerged as an important element of the utility of the software to the British Council, and for the Montage project alone more than 58 document types had been defined for various aspects of website content management – many of these to facilitate workflow management of original publishing.

It became clear during the course of the project, and is evident in the reuse evaluation above, that content reuse was significant in the following key areas:

1. In the reuse of design structures for country replication, workflow and document ontology reuse and template design inheritance
2. The reuse of content for content discovery and navigational support
3. The reuse of content for syndication and content exchange with other systems.

Certain categories of document, and in particular, link information (that is links to other educational resources and websites), projects, schools and particular categories of web page content, were clearly more important for content reuse than others. While there was increasing use of the prototype for multimedia content management – this was principally directed to use of the software as a multimedia resource repository rather than for extensive image reuse.

The Web Services framework became increasingly important as the principle point of mediation between the CMS and other services at the third layer and fourth layers of the model. Outside RSS syndication, some projects were transferred using web services to the other CMS product used by the British Council – ObTree. Similarly, XML interfaces were used to transfer content from ObTree to Inter-Publish. Moreover, the exchange of information from local client databases (such as outlook) to Inter-Publish, achieved through the CMR web services layer proved an effective means of content gathering – centralising disparate resources, including database resources, in the central CMS clearinghouse. Once again, much of the focus of such content clearinghouse activity centred on specific categories of content, and the continuous transformation of content.

The literature review in Chapter Two pointed to the emergent role of the CMS in continuous transformation. This role may not be across the full universe of content, but emerges

as an important aspect of content reuse through the role of the CMS as a central content clearinghouse and content repository.

It would be valuable to undertake further research to determine if this pattern of primary content regeneration and content reuse is similar in other segments of web publishing (such as tertiary education, or government) with a similar analysis of content usage over time.

7.3 Post Implementation Survey

In order to provide feedback to the British Council on the user acceptance of the prototype implementation, a Post Implementation survey was conducted among participants. The participants were widely dispersed and the survey was directed to currently active Inter-Publish authors (located in Australia, Indonesia, Thailand, Hong Kong, UK, New Zealand, Malaysia). An online survey approach was therefore adopted. Participation by authors in the survey was voluntary and the target community was small. There were18 content authors in total at the time of the instrument delivery. The survey includes some questions on content reuse, but was principally directed to eliciting the content authors reaction to issues of utility for purposes of publishing and ease of use. Research in the Information Systems field on why organizational users accept or reject information systems has resulted in a theoretical explanatory model called the Technology Acceptance Model (TAM). The TAM has been developed as a means of avoiding more ad-hoc approaches to eliciting information from users in end-user surveys (Bagozzi, & Warshaw, 1989). The need to discriminate in the survey design between usefulness and ease of use is an important facet of Technology Acceptance Model and provides an effective means of discriminating attitudes to functionality from attitudes to ease of use. Such a survey can act as an "early warning" of issues. The TAM survey approach explicitly differentiates "actual usage", "usefulness" and "ease of use" (Davis, 1989; Davis, Bagozzi, & Warshaw, 1989). A Likert scale was used with 5 categories of response, from Strongly Disagree to Strongly Agree. The survey is described in more detail in Appendix Four.

There were 12 respondents to the survey. While this is a small survey population, the results of the survey provided some useful feedback to the British Council and have some relevance to this research. Given the small population of users, the survey was delivered principally as a means of evaluating the user response to Inter-Publish and Obtree (some authors had experience with both). Questions elicited feedback about the products regarding actual usage, usefulness and ease of use. Of the 12 respondents, 10 had five or more years experience with computers, and 8 indicated that they were "familiar" with web publishing, and 7 were using Inter-Publish at least once a week. While 8 respondents expressed awareness of Obtree, only 2 were using this software at least once a week, and as a result comparative measures do not warrant analysis.

7.3.1 Responses relating to "Ease of Use"

Questions pertained to the degree to which the software was “flexible to interact with”, “easy to use”, or more specifically “I would find it easy to get Inter-Publish to do what I want”. In relation to Inter-Publish, the average response on the issue of ease of use was 2.5, or somewhere between “Neutral” and “Agree”. However, work on workflow and content visualisation had clearly yielded positive results, with a strong affirmation on the question “My interaction with Inter-publish is clear and understandable” (Figure 7.5).

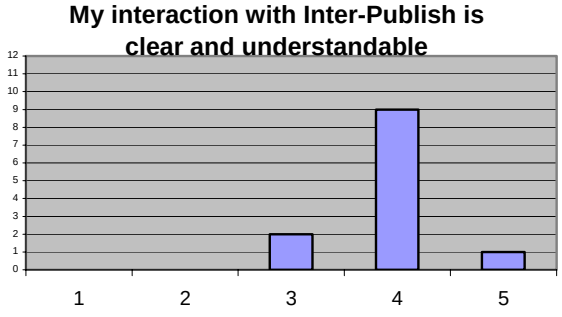


Figure 7.5: Responses on user interaction with Inter-Publish

7.3.2 Responses relating to Task utility

The responses to the questions on usefulness were generally positive. The average response to these questions was 2.8, or leaning strongly to "Agree", and particularly in relation to the questions pertaining to the publishing capability (see Figure 7.6).

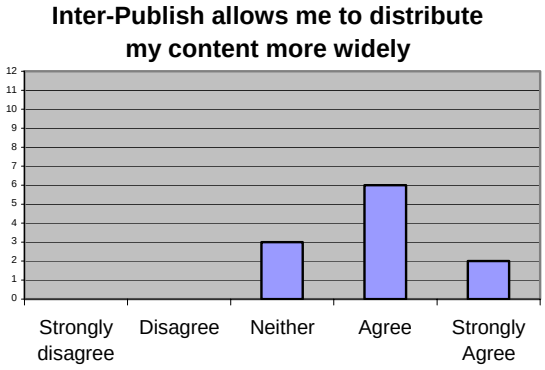


Figure 7.6: Task utility in publishing.

The differential factor between "Ease of Use" and utility of the software suggest a need for further work required on the user interface, but general task-related satisfaction with the software. A lower rating by survey respondents for Obtree in this category of questions may simply reflect their greater familiarity with Inter-Publish at the time of the survey.

7.3.3 Responses in relation to Job utility (Job related questions)

Responses to the questions on Usage in relation to the participant's job-role (Figure 7.7) were once again generally positive, particularly in relation to execution of the tasks in relation to publishing.

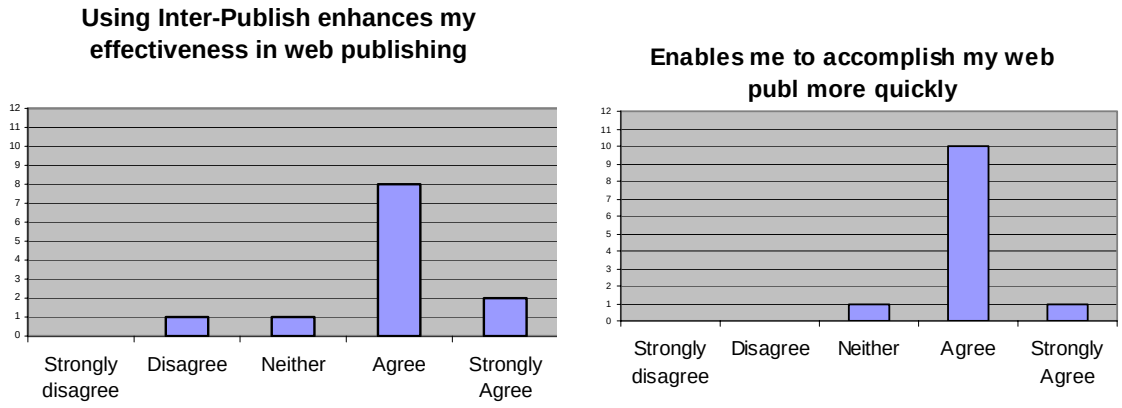


Figure 7.7: Responses on utility in relation to Job Role

7.3.4 Post Implementation Review Summary

Given the small numbers involved the findings cannot be generalised, but do point to satisfaction with the functional purpose of the software, and its ease of use. A lower rating for "ease of use" is consistent with the "raw edges" nature of the prototype, and points to further improvements in user interface design and content visualisation that may be required.

7.4 Conclusions from the Evaluation

The model presented in Chapter Four evolved from a comprehensive literature review evaluating industry practice against theoretical elements of content management and content reuse. This led to the layered model for content reuse described in Chapter Five.

The CMR model was tested through a rich prototype implementation that explored all layers of the conceptual model. This prototyping research process was directed to a non-trivial web-publishing environment that explore the complexities of the core concepts of the model. The outcomes of the Software Development Research were a prototype that provides a rich presentation of the functions at each layer of the model, exposed through Web Services, and a document ontology describing each layer of the model. A variety of interfaces were implemented

at each of the layers of the model, and particularly at the workflow layer, to demonstrate the flexibility of the model to different interfacing standards.

This review used versioning history and detailed authoring and publishing activity to track the outcomes of the project in facilitating content reuse as against primary publishing and the aspects of the model that facilitate this process. It established that over the period of the field trial, there was evidence of increasing content reuse, as distinct from simple site editing. There was a focus on the reuse of particular content types, an issue that was highlighted by the layered management of content and the ECU chunking process. In addition the prototype analysis indicated that the framework allowed a high level of information currency, sustained despite significant growth in published web content during the duration of the project. The ECU model opens the opportunity to define generalised metrics for the measurement of content currency and content reuse.

The field experiment however, explored other dimensions of content reuse. Beyond syndication and reuse of informational content across multiple sites and media, design reuse became an increasing factor in site management. Templates reached progressively higher levels of abstraction to facilitate the white site replication process for new country sites and the selective addition and removal of sections within country sites.

In the nature of a field implementation, the focus of content reuse was in areas that were most likely to meet the immediate needs of the client while still exploring concepts in content reuse. Content was syndicated through data interchange with another CMS (Obtree) and through generation of RSS (Rich Site Summary) XML. Further research would be valuable in the context of a client with a different domain of interest and of a client with a demand for syndication in more complex ontologies.

Web Services were developed as a means of cross-integration of the CMS with other applications (such as the ObTree CMS also used by the British Council). Through extension of

existing XML and URL interfaces the architecture readily supported the SOAP–based approaches represented by Web Services for information interchange. The document ontology supporting the Web Services was also supported through XML-RPC services and HTTP query strings to provide developers working with the CMS simple standardised interfaces at each layer of the model.

VB6, being familiar to the author, was ideal for the RAD-style prototyping used in this project. The inefficiencies of VB6 in string processing are notorious and do not make it an ideal language for scalable performance of the application layer. VB6 also lacks multi-threading capability, limiting the extent to which the application is scalable to multiple concurrent users within a single server – although the application was designed to support multiple concurrent servers operating as a front-end. Elements of the application would significantly benefit from implementation in a more scalable architecture providing better concurrency and multi-threading.

The implementation of content in the ECU framework also offered possibilities for text mining techniques to facilitate the generation of descriptive metadata to aid the authoring process. An interesting avenue for further research is the extension of text mining techniques at the authoring stage to enhance the metadata categorisation of content and recommendations for deep hyperlinking to similar relevant documents.

The Software Development Research presented in Section 7.2 culminated in a sophisticated application which implements both the functional elements of the model and a detailed Web Services interface for further extension of the application. The evaluation of the content reuse benefits of the implementation discussed in Section 7.3 demonstrated a strong reuse in particular content items, indicating also what may be a strong distinction between content types. Extension of this analysis in other publishing environments could yield further insights into this aspect of content reuse, especially if extended to other domains of publishing. Nevertheless, the evaluation of content reuse in the Montage project clearly demonstrates "systematic" rather than "opportunistic" content reuse (Rockley, 2003).

Content regeneration and reuse are not shaped by a particular XML ontology but require a systematic approach that addresses the reality of continuous transformation of objects in the digital environment. Given the small size and budget supporting the Montage project, it is arguable whether the Montage project could have been achieved without content reuse capabilities of the prototype. Based on this evaluation of the prototype, the author is confident that the Content Model for Reuse can be implemented, can meet the needs of a challenging environment for content publishing and reuse; and finally, shows interesting avenues for further research in content reuse.

Potential directions for further research and development surrounding the model and the prototype are explored in the next Chapter.

Chapter 8 - Summary and Conclusion

"…the enjoyment of this flower is of brief duration: for it is so fragile, its petals so lightly attached, that it quickly falls, shaken from its stem by those same winds that give it its name, anemone." Ovid. Metamorphoses *Book X.*

This chapter summarises the outcomes obtained in the course of this research. In particular it assesses the Content Model for Reuse (CMR) and the prototype implementation of this model (Inter-Publish), which are the main contributions of the thesis project. The CMR introduces a novel and unique approach to content management aimed specifically at multimedia content reuse. In the course of this research a set of principles for content reuse were elucidated through an industry survey of current functions and a comprehensive literature review pertaining to content management and content reuse. A synthesis of these principles resulted in an abstract model - the CMR, which was evaluated through Software Development Research comprising a rich prototyping development project and an empirical evaluation of the effects of the prototype in changing patterns of content use and reuse. Section 8.1 provides a summary of the thesis findings, re-introducing the main research question and shows how it was answered by addressing the six subsidiary research questions introduced in Chapter 1. The section also lists the major strengths and contributions of the thesis resulting from the rich prototyping process. The Economic Content Unit (ECU) serves to provide a common approach to management of heterogeneous content forms. Reuse is enhanced through the inherent ECU versioning structure and layered model for CMS design.

The thesis limitations and implications for IS research and practice are discussed in section 8.2, which discusses the performance, further evaluation and validation studies, and possible extensions to both the method and the prototype. Special focus is placed on the broadening of the thesis's main concepts and transfer of technology developed in this thesis to other areas of possible interest.

8.1 Summary of Findings and Contributions

More than ever people are both content creators and consumers of digital media – of both text and multimedia. While systems for original publishing of content abound, there is a general failure to leverage this content in regeneration of content resources and reuse of content in other media formats. This project investigated the issue of multimedia content reuse with a particular focus on the question:

> What approach to content management will allow effective reuse of content across heterogeneous multimedia resources tracking version changes across language and time?

This question lead to several other subordinate questions (see section 1.2), which were considered and then addressed in this thesis as follows.

1. What are the methods, techniques and tools available to support content reuse? The Research approach taken on this question comprised a comprehensive Literature Review leading to a set of principles for content regeneration and reuse.

2. What reuse models so far have been successfully applied to the process content publishing? This question is explored in an Industry Software Review as part of the Literature Review.

3. Is there an alternative model to support content reuse? The question was addressed through a synthesis of the theoretical research leading to a proposed new Content Model for Reuse.

4. To what degree can the proposed model be implemented in practice? This was explored through the Software Development Research and a rich field implementation testing different aspects of content reuse.

5. To what degree is the proposed model effective? This is explored through an empirical study of the reuse consequences of deploying the model in the multi-national field trial. The results of the field trial in the context of content regeneration and reuse were evaluated quantitatively and qualitatively.

6. In what other ways does situating the content in this new model benefit the ongoing management of the CMS? This question was also explored through Software Development Research, the outcomes of which were quantitatively examined for their implications in content reuse.

The comprehensive survey of theoretical literature in the area of content reuse and regeneration elicited the major principles underpinning theoretical domain (research question #1), supported by a detailed industry review of current techniques (research question #2). It was apparent from this review that the issue of content reuse is larger than can be addressed by a single document or semantic ontology - that this is, in fact, an information systems issue. A content reuse system must straddle a heterogeneous set of encoding formats, semantic metadata frameworks and document structures, and continuous transformation of ephemeral and strategic content to a range of delivery systems, and mediated through environmentally contingent workflow practices. Principles for such an Information System were elucidated in Chapter Two and these principles guided the formulation of a research methodology (Chapter Three) yielding a layered model as a conceptual framework for content reuse presented in Chapter Four (research question #3). This conceptual architecture is then described in terms of Web Services interfaces at each of four layers supported by a Web Services document ontology. Software Development Research was pursued to create a functionally rich Information System "Inter-Publish" which realised the model (research question #4). This model was tested in a complex web-publishing environment that experienced rapid growth and made extensive use of elements of content reuse (research question #5).

This field trial was evaluated through quantitative analysis of empirical data of primary publishing activity supplemented by final user satisfaction survey (descriptive in nature given the small community of publishers). A lightweight transformation language was developed to support high-volume publishing demands, with additional capability to support XSL and other transformations. In addition to the merits of the model for content reuse, the field trial highlighted possible directions in enhanced metadata and content similarity matching which are facilitated by such an approach to content reuse (research question #6).

The Inter-Publish prototype CMS illustrated the flexibility gained by the Economic Content Unit approach to content fragmentation and the flexibility of the ontology to deal with a wide range of content forms without specific coding.

8.2 Potential Implications for IS Research and Practice

8.2.1 Authoring and WYSIWYG

With the importance of content separation and semantic enhancement, the WYSIWYG paradigm for content authoring has been relegated to the status of an end-user compromise. However, WYSIWYG has been a persistent survivor in the industry implementation of content authoring systems. So ubiquitous is the demand for WYSIWYG from the content author that it would be a brave CMS developer that omitted such a feature. This tension warrants re-evaluation toward understanding the ways in which this might impede effective metadata description and content separation.

While the process of content markup, semantic enhancement of the data and fragmentation of content elements has considerable advantages for the publisher, it can represent an imposition on the author. CMS discussion lists reflect this tension:

> There is a fundamental imbalance here between cost and reward: most content providers get no benefit converting to structured authoring; however, it does costs them both time and effort to switch. It is not

> just being deprived of a WYSIWYG view of your work. There is a lot more you have to know and monitor carefully in order to produce a structured XML document, whether or not your authoring tool actually makes you type angle brackets. (Baker, 2002)

This imbalance could be an expression of the poverty of existing authoring systems. Alternatively it could be an expression of a basic tension between the semiotic process of content creation and effective content separation and metadata enhancement.

When the burden of metadata enhancement is placed on the content author, this is a move from the phenomenological (expression focused) approach to authoring represented by WYSIWYG to the noumenonogical (meaning focused) approach. Content authoring can be regarded as a semiotic process of creating a particular syntactic expression of the authors meaning. Semantic metadata enhancement, and to some degree the abstraction of structure from the "content", can be regarded as a reductive *a postoriori* process. The reluctance of many editors to discard WYSIWYG is symptomatic of the tension implied by combining the constructive and reductive process of semantic attribution and content separation. The attraction of WYSIWYG is precisely the extent to which it necessarily engages the author with issues of structure and style in fully expressing the meaning.

Having worked with the clients undertaking content authoring for this CMS, it is clear that understanding the place of WYSIWYG will yield insights that may assist in better design of authoring systems toward the enrichment of the production process. It was clear in the evolution of this prototype that a base-line user requirement for the CMS was WYSIWYG editing. This applied not only to the process of content editing, but also in the broader site-contextualisation and visualisation

The *editorial* intervention at the larger level can also be considered an issue of WYSIWYG in the sense of the selectivity of content for the final expression at the macro level – of the *publication* as a whole rather than just the *item*. Staging systems of one sort or another are a

fundamental element of most CMS products. This process of staging is a further expression of WYSIWYG at the whole site level.

Content separation is a central plank of effective content management, especially for purposes of syndication. Yet in many ways the author may be best situated to judge the appropriate use of content across different contexts (Overby, 1999). In the field trial the author explored approaches to adapting the authoring system to retain the benefit of WYSIWYG presentation as the central pivot for content capture, while seeking effective methods of enticing the semantic and structural information from the author grounding in *a postoriori* understanding of the relational information understood about the content. Semantic metadata enhancement can be positioned in a manner which respects the central role of authoring while enticing or eliciting the provision of semantic metadata by a combination of automatic classification against relevant domain taxonomies that are confirmed by the content creator and workflow management of the collection of essential metadata as a prerequisite for completion of the publishing process.

The knowledge resource represented by the document collection of a large CMS lends itself to the application of text mining techniques. Such techniques can be used not only to enhance document discovery, but also to improve the authoring stage elicitation of semantic metadata as a means of remediation of the underlying semiotic tensions in content authoring and the decomposition of content for reuse.

Content authoring on a CMS is rich in contextual information. The proposed location of the website content, semantic metadata of surrounding documents are known, and the content itself is often reliably separated from style. The capture of valuable semantic information can be greatly enhanced by a multi-methodological approach to analysing and confirming with the author semantic categorisations of the document and possible extended inter-linkage for the document based on similarity with other documents on the website. This process can be further enhanced where website activity logs can be integrated into the analysis.

Two strategies have emerged among CMS vendors for content transformation: custom template-based transformation and XSLT. Template-based approaches facilitate integration of WYSIWYG presentation of content in the authoring process to enhance the visual presentation during content capture through a process of:

- Editorial contextualisation of content at all levels of publishing to ensure authoring and editorial credibility is retained.
- Content analysis combined with inter-linking and document similarity analysis to provide structured guidance to the content author in semantic metadata enhancement grounded *a posteriori* in the content itself.

Chapter Five explored some of the early semantic similarity algorithms in the context of the rich structural information available in the context of CMS authoring. Further research in this area would be needed to compare different methods of text data mining and their efficacy in a context-rich environment such as Inter-Publish

8.2.2 Detecting content suitable for systematic reuse

The field trial and empirical analysis of content reuse activity indicated that there were clear boundaries between types of content that were commonly reused and types of content directed solely to primary generation. Further research would be valuable, perhaps through field trials in other domains, to expand further on this differential between content types and its implications in design of content management and reuse systems. In particular, a combination of initial text mining to explore structural elements of existing websites and semantic relationships in such sites could greatly facilitate the adoption of existing website resources in a CMR-style of Content Management System, and could yield further insights in the early recognition of content that can be targeted for syndication. Such research would be particularly interesting if bolstered by external web log information on actual web page usage as a further data mining resource.

8.2.3 Open Source Directions

There are now a number of Open Source content management products available. A detailed functional evaluation of the work required to take such products into an ECU-based approach to content management would be a natural extension to the industry survey conducted in the course of this research. The opportunity also exists to consider placement of the prototype into open source, although this would sensibly follow a transition to a more suitable open source community development language.

8.2.4 Performance

While the Inter-Publish prototype clearly transformed the capability for content reuse in the field trial environment, the prototype could be significantly improved in a number of ways. The final survey indicated a gap between utility and ease of use, and further enhancement of the end user design for content authoring would help the commercial transition of this software.

The Inter-Publish prototype, being built around the ECU framework, is flexible in the accommodation of multiple different content profiles, either for different organisations within the same content store, or for distributed content across different databases. The system performance measured here was measured on a Pentium IV with 1GB of RAM, 1K GHz CPU running Windows 2000 Server and using MS SQL Server or MS Access for database management of content.

This prototype framework was scalable to support of more than 20,000 documents in a single database and comprising more than 100,000 ECU elements. User interface delays were experienced at times, but these were principally attributable to the limited bandwidth of the connection to the content server.

The prototype was designed to maintain its own session state. This enabled stability and scalability using standard techniques of load balancing, and in particular the operation of multiple content servers with a single database in a load balanced array, and the splitting of content over multiple database platforms in a many-to-many relationship between servers and content stores. A relational table was used to maintain persistence of state information. There are limitations to the scalability of such an approach to very high numbers of users. A more appropriate approach would be a greater use of Transaction Processing concepts and/or CORBA-style state-persistence techniques.

Each server platform uses an underlying XML configuration specification identifying for each "client" the relevant database connection details and network service addresses for the service, and whether that service was responsible for various scheduled chronological tasks (such as future publishing of content).

There are several design changes to the principle content store, the *DocumentContents* in the prototype table that would improve versioning and provide better efficiency in content management. In particular, there were inefficiencies in the content storage of text BLOBs (Binary Large Object) in conjunction with affiliated metadata. Searching content across BLOBs in relational tables is notoriously inefficient. These efficiency issues were addressed through the development of a word frequency table that formed the basis both for content discovery and similarity matching across document sets (see Section 5.5.1.1 on text mining of content resources). Decoupling the BLOB data to a separate table would further extend content view as a linear slice across the terminal nodes of a content hierarchy, but would require further testing for their relative efficacy. Other relational design changes to facilitate versioning of document ontologies would also improve the capability of the prototype for full website versioning generated dynamically from versioning history.

The performance of Inter-Publish could be considerably improved. The development in a Microsoft VB6 framework facilitated rapid prototyping but is not an ideal platform for scaling to very large document sets and has known weaknesses in its efficiency of string handling.

8.2.5 Future directions in the prototype

No Software Development Research of an actively used system can ever be considered "finished". Planned enhancements to the prototype include changes for practical usability, performance improvement and forward directions in interesting areas of theoretical research. These include the redevelopment of the system architecture, and the extension of the Web Services model, enhancement of the user interface, integration of further text mining in the authoring process, and possible integration of automatic translation interfaces to enhance the multilingual versioning elements of the system. Further experiments comparing the relative merits of XML database engines against relational databases for fragmentation of content for reuse would also be valuable.

The empirical evaluation conducted as part of this project focused on a particular content reuse environment intending to verify the efficacy of the proposed model and evaluate the effectiveness of the Inter-Publish reuse application. The field trial was invaluable as a complex and challenging website in which to evaluate the prototype and give impetus to its development. The underlying text mining resource offers opportunities in further development to support research into cross-analysis of authoring metadata, external web usage logs and structural, ontological and semantic information for authoring support.

Further studies are needed to determine aspects of Inter-Publish effectiveness in other contexts. Such studies may include determining Inter-Publish methods for automatic fragmentation of content from existing websites and content resources, as well as automatic mapping systems to facilitate fragmentation of content from well structured existing document ontologies.

There have been a number of behavioural and organisational assessments of software reuse, most notably that of Sherrif and Vinze (2003). The importance of the role of the "reuse champion" merits evaluation in the context of content reuse.

A major theoretical undertaking should address the broadening of the content reuse model to address issues of long-term archival management of content, an issue poorly addressed in most current publishing activity. As the semantic web achieves greater momentum, its role in the CMR model warrants further assessment.

Tables 8.1 through to 8.4 describe a plan for extending and evolving Inter-Publish in both research and commercial terms. The medium-term goals indicate the path towards the Inter-Publish "productisation" and to fill out the content reuse functionality for other publishing contexts, so that the resulting system could be of practical use to other organizations. They address the actions needed to move away from a specifically research-oriented system prototyping and scientific exploration, and to move towards a platform that supports more widespread adoption. The long-term goals are described in terms of targets for the system planning to realize emerging opportunities and interesting research directions.

Layer 1	Medium Term Goals	Long Term Goals
Support for content in many languages	Explore the integration with automatic translation systems.	
Internationalisation of user interface	Undertake trial with British Council Japan users to test Unicode internationalisation of the Inter-Publish user interface	
Multi-lingual content variants	Translation support or presentation of semantically identical content on a single page with multiple language variants. Improved WYSIWYG integration handling spell checking across different language classes	Difference checking across language variants, particularly in a multi-lingual ECU setting.
Encoding variants	Review of the WYSIWYG design for multimedia presentation of encoding variants, especially in image handling.	
Data Mining	Explore semantic similarity analysis in conjunction with additional external resources such as web server usage logs.	

Table 8.1: Future directions (Layer 1)

Layer 2	Medium Term Goal	Long Term Goals
A formal separation of content from stylistic form is implemented, with Template-based or transformation structures to yield published content.		
A flexible ontology that supports the definition of document types amenable to simple change and expressing complex content elements (Economic Content Units).	Review the implication of the Semantic Web as the OWL ontological content definition language achieves industry acceptance.	
Ontology versioning.	Improve versioning to allow editing of previous versions of the document ontology without disturbing the current document object (currently not possible).	Ontology differencing and the ability to import ontologies based on different ontological definition languages.
Content can be used outside the particular web page context to which it is published.	Implement additional well-accepted syndication approaches (for example NewsML).	
The ability to create content collections for use by reporting, indexing and syndication engines for transformation and distribution of content.	Support for content definition based on existing XML/DTD definitions.	
Support commonly used document structures, and validation against those structures.	To build a facility for versioning of ontological structures synchronised with the ECU content using these structures.	
Document ontology can itself be documented or is self documenting.	Parallel viewing of current and previous versions. Sophisticated version differencing.	
Generalised database support for content storage.		Support for other relational database systems, and experimental work with XML database systems.

Table 8.2: Future directions (Layer 2)

Layer 3	Medium Term Goal	Long Term Goals
Content editors can be defined with different levels of content visibility.	Group-based or LDAP-style control on access to content resources.	
Editorial constraints on modification to templates.	Group-based or LDAP-style control on access to content resources.	
Content can be discovered through transient and persistent collections.	Support for Z39.50 query.	Examine the use of XML Query.
Multi-lingual publishing.	Content translation component integration.	Explore approaches to presentation of different language variant with complex language handling in a WYSIWYG authoring context.
Temporal Versioning.	The ability to preview the historical versions on an "as at date" browsing the site as it looked at that time, generating the preview "on the fly".	
Encoding variants.	Better image and multimedia component handling integration and WYSIWYG presentation of variants	
Workflow Routing capability and establishment of groups.	Extend the authentication system to provide logical group support and user subscription to groups with privilege-based tokens.	Examine complex routing to multiple target groups and in particular in a multi-lingual authoring context with relevant differencing support.
Performance metrics.		Reports that analyse the detailed tracking of content version information would allow detailed analysis of performance metrics.
Management of events and workload.	Complex event handling (particularly data driven event handling as distinct from the current publishing driven event model).	Improve the visualisation of workload measures against the authoring page.
Task Management.		Extended task management – a change that would require a more demanding workflow process than Montage requires.
Availability of graphic tools.		Explore VML for modelling of the workflow process.
Documentation.	Documentation is never complete enough to support a dynamic, evolving system. Enhanced online help and visualisation of the workflow process are key end user documentation resources that would be valuable.	

Table 8.3: Future directions (Layer 3)

Layer 4	Medium Term	Long Term
Primary generation of web pages.	All suppliers can meet this requirement in one form or another. Template rather than XSL transformation, often through supplier-oriented template-based methods remain the most popular means of content transformation.	
Syndication to varying document ontologies.	Major suppliers have developed plug-ins for specific syndication ontologies. Explore the options for integration of suitable transformation interfaces to existing syndication technologies.	Work with new developments in Semantic Web/OWL.
Ability to support alternative transformation approaches.	Expose this element in the Web Services interface for Layer 4.	
Website versioning.	While the theoretical elements of this feature are implicit in the ECU and ontological versioning elements, there remains considerable effort in resolving this as a full editioning capability.	
Delivery agency.	Develop improved content expiry capability in the delivery agent, and better alert on repeated failure or content alerts where the external content does not match current content. Explore other delivery agents (for example secure delivery protocols).	

Table 8.4: Future directions (Layer 4)

8.3 Conclusions

Content reuse is an important area of software engineering and information systems research. Key factors affecting systematic content reuse are the heterogeneous nature of digital encoded content, the need for ontological structures to coherently organise content for reuse, the need for workflow systems to facilitate content authoring, the importance of systems for content discovery and the reality of continuous transformation that is demanded by regeneration and reuse of content. This research has explored a conceptual model directed by a set of principles to facilitate systematic content reuse and address these underlying complexities. An information systems model, the Content Model for Reuse, is proposed rather than a single document or semantic ontology. This model was presented in four layers, each layer building in complexity on the lower layer, and each layer addressing specifically aspects of the content

regeneration and reuse problem. The Content Model for Reuse was explored through Software Development Research and a field implementation in the process of which a functionally rich semantic interpretation of the model was completed.

This thesis has explored many aspects of the theoretical and practical issues surrounding content regeneration and reuse. The Content Model for Research has interest across a number of research communities while also offering a timely contribution to industry developers at a point of rapid change. For those researching the Content Management Systems specifically, the outcomes of the research indicate a direction forward which can:

1. Address issues of inconsistency in the presentation of the same semantic information across different media.
2. Ameliorate the economic costs of content recapture during generational changes of content.
3. Reduce the opportunity cost represented by the failure to deploy content in all relevant content forms (for example through syndication).

The information systems approach to content reuse would be of relevance to extending research approaches to Software Reuse, and particularly with respect to the ubiquitous versioning described in the CMR approach to content management and the possibilities this offers in the complex management of open-source community development.

The recognition that content reuse is founded on a series of principles that can be described in a coherent systems model has relevance in the area of Information management generally and Knowledge Management specifically. The use of text mining offers opportunities founded on this model can extend text mining by capitalising on the context-rich information environment provide by the CMR in a manner that improves the subsequent utility and consistent semantic description of the content feeding through to Knowledge-based systems. The CMR model also has some relevance to the organisation of information in heterogenous corpuses of resources (such as Digital Libraries).

As a complex implementation of a multi-layer model in Web Services, but also expressing similar services in an identical XML document ontology through other delivery agencies (HTTP through standard form posts and XML-RPC), the thesis offers some insights for those exploring the area of content reuse in the context of Web Services, and particularly in the model-based application of document ontologies in a manner that facilitates use in a Web Services delivery architecture. The model also offers insights into a multi-layer approach to web services definitions, and the possibilities for a layered XML document definition to be implemented through multiple services, and multiple delivery agencies. The research outcomes also have relevance to researchers examining models for content retrieval across heterogenous content. The use of the Web Services explored in this thesis takes effective elements of Z39.50 (allowing distributed searching and static search lists) in conjunction with the CMR layered approach to the underling content resources and an XML framework for query specification.

The metrics described in this research, based on the CMR with ubiquitous versioning across layers of the model, provide reliable and generalised approaches to measuring content currency in published web-based resources. These metrics could be extended to offer insights into the latency involved in the life cycle of the content resource, from creation through to publication and expiry.

Finally, obsolescence remains a major research issue in the long-term management of digital content. The approach to continuous transformation realised as an element of the model are of relevance to researchers in the archival management of digital multimedia content. Such long-term archival management of digital content would benefit from the principles for content reuse realised in the CMR model both in the organisation of content to facilitate such transformation and in the methods to enhance long-term content discovery and exchange through a layered model-based approach to the problem.

GLOSSARY

- A -

ANSI X12: A business document exchange ontology. Favoured by North American businesses.

ASCII: American Standard Code for Information Interchange. Along with EBCDIC an early English-language character encoding system for digital text encoding. Most text editors can save content in ASCII form.

ASP (script): *Active Server Page.* A web page that contains *VBScript* - a web based interpretive version of Microsoft Visual Basic.

ASP (server): *Application Service Provider*. A particular business model for licensing of applications using a centralised server delivered over an Internet framework.

- B -

Bandwidth: the capacity for transfer of information between computers. A wider bandwidth means more content can traverse the network in a shorter amount of time.

BLOB: *Binary Large Object.* A relational database field supporting the encoding of (often) indefinitely large objects. Such objects can include double-byte text encoded data.

Browser: software for navigating the Web, retrieving documents and other files, commonly in HTML markup format.

- C -

CMR: Content Model for Reuse. The model proposed in this thesis for systematic reuse of multimedia content.

CMS: Content Management System.

Codec: The compression and decompression algorithm for audio and video content.

CSS: Cascading Style Sheet. The CSS defines rule-based presentational instructions for HTML content markup. The Style Sheet has the merit of gaining a greater freedom from the specific encoding of procedural markup within the text itself (with the <font> tag and others).

- D -

DCMI: *Dublin Core Metadata Initiative* **-** a standard for consistent meta-identification of website publications

Digital (content): Information that is encoded in binary (discontinuous) form particularly and mediated by computers.

Digital Watermarks: A unique digital signature is embedded in the document, image or multimedia item – in a manner very similar to the traditional watermark on paper. Digital watermarks however, can contain meta-data about the content or the content creator. This meta-data might identify:

Ownership

Extra data and information

Embedded hyperlinks.

Directed Acyclic Graph: A directed graph where no path starts and ends at the same vertex.

DLS: *Digital Library System.*

DMS: *Document Management System.*

DOM - *Document Object Model.* The strictly hierarchical specification for the ontological structure or organisation of a document. HTML is an example of a DOM.

DTD: *Document Type Definition.* The specific set of rules defining what elements and attributes may be used in SGML and XML.

- E -

ECU - Economic Content Unit. The content "primitive" of the CMR. The ECU contains a single semantically coherent fragment of information, with variants across language and encoding.

EDI - Electronic Data Interchange. The exchange of business documents (and financial transactions) in the course of business operation.

EDIFACT - A business document exchange ontology. Favoured by European businesses.

- F -

Flash: An animation component from Macromedia for use in web Browsers. Open source documentation for the Flash document format has been released.

FTP: *File Transfer Protocol*. One of the earliest file interchange protocols on the Internet. Still a very popular protocol. Generally passes passwords in free text and so has major security limitations.

- H -

Host: Any computer that is the central point of connection to run an application or obtain information (for example a Web server). In the Internet a Client Web Browser connects to a Host Web Server to exchange HTML and other information.

HTML: HyperText Markup Language. A set of markup instructions for creating documents for use on the World Wide Web. The HTML standard is defined and controlled by the World Wide Web Consortium (W3C). An SGML-compliant DTD for HTML (XHTML) has been published by W3C.

HTTP: Hypertext Transfer Protocol. This defines the communications protocol by which Web Browsers and Web servers communicate.

Hypermedia: the general conceptual approach to interlinking multimedia documents through all forms of object links (including text hyperlinks).

Hypertext: the specific implementation of hypermedia in text form. A particular word or phrase is made active (through mouse click or keyboard action) to launch another related document. The term was coined by Ted Nelson in 1965. The HTML "a" anchor tag is used for hypertext formatting in the World Wide Web.

- I -

IEEE/LOM. The Institute of Electrical and Electronics Engineers Learning Object Metadata - an ontology for describing learning objects - popular in the IT community for describing IT technical/theoretical literature.

Internet: An Internet is a group of networks of computers that are connected by some common protocol. The Internet refers to the global connection of computers using the TCP/IP protocol.

IP: *Internet Protocol*. A protocol defining the numerical addressing and routing rules on the Internet.

IP: *Intellectual Property*. The tangible output of creative intellectual activity in a particular expression – eg, a book, a programme, a piece of music, a poem, an invention.

IS: Information Systems. Those systems which provide functional support to organisational processes, management and communication through use of computing technology.

- J -

Java: a high-level, object-oriented programming language developed by Sun Microsystems. A "p-code" language, it is designed to be portable across most operating platforms through the use of a small "virtual engine" specific to each operating system. That portability and its object-oriented design has been a factor in its popularity.

Javascript: A popular scripting language developed originally by Netscape Communications in order to animate HTML pages. It is only loosely based on Java.

- K -

KM: *Knowledge Management.* The class of software and domain of research concerned with the encoding and discovery of knowledge as a resource.

KMS: *Knowledge Management System.*

KWIC: *Key Word In Context.* A search result display which shows the keyword searched in the sentence context in which it occurs.

- L -

LAN: *Local Area Network.* A group of computers connected together for high-bandwidth file and application sharing.

- M -

MARC - *Machine Readable Cataloging*. A metadata ontology for exchange of bibliographic information.

Markup: The placement of identifiers in text from which can be inferred information regarding the presentation, formatting and structure of the text or which adds additional commentary regarding the text (but not part of the text). *Procedural markup*

Multimedia: Any combination of text, audio animation and video content in a digital form.

- N -

NewsML: *News Markup Language* - a content exchange framework specifically designed for XML interchange and syndication of news items.

- O -

Obsolescence: Specifically in the context of technology: the way in which computer hardware or software becomes out of date in a way that renders its use progressively more difficult or costly.

Ontology: A formal definition of the relationships between content objects and framework for describing these content objects.

Open Source: The *Open Source & Free Software Foundation* is a trust-based means of developing high quality software. Distribution of the source code is free, and redistribution on this same basis is mandated through a licensing agreement. The economic argument for such an approach depends on the "reputation value" of the product leading to income through services and as a means of ensuring that a particular software product remains and develops in the open community of developers.

OWL: Web Ontology Language**.** The ontological language framework for the Semantic Web.

- P -

Parser: An application that semantically deciphers content according to specific rules or structures. An XML parser facilitates the hierarchical exploration of an XML document. A language parser may attempt to discover the grammatical constructs in a sentence or computer algorithm.

PCDATA: The "content" portion of XML - whatever is contained between XML element tags. There are some encoding conventions for PCDATA in XML.

Perl: *Practical Extraction and Report Language*. A scripting language with strengths in text parsing and processing. Perl is an *interpretive* language.

Persistence: Establishing a reliable and long-term (rather than transient or anonymous) presence that can last beyond a particular interaction. URL persistence concerns the availability of a web page over the long term at a known location. Session state persistence relates to the use of Cookies to maintain a specific information relationship over time between a browser and a web server.

PHP: PHP Hypertext Processor (the definition is self-referential, or recursive).

Protocol: the formal set of rules for communication between network devices or applications. Protocols are generally managed and published by international standards organisations.

- R -

RAD: *Rapid Application Development*. The use of a heterogeneous mix of software development tools and development methodologies to accelerate the design process.

RDF: *Resource Description Framework***.** The RDF specification (Lassila & Swick, 1999) aims to provide a formal model using directed graphs to describe the semantics of metadata and of cataloguing web-based resources.

- S -

SCORM: *Sharable Content Object Reference Model*. Similar to the IEE/LOM, but providing a richer framework describing the metadata ontology describing educational objects and resources.

Search Engine: a means of cataloguing, classifying and searching based on ranking rules for content on the Web.

Script: a loosely timed, often interpretive, computer programme. Often embedded within an application framework to add user control or dynamic functionality to an application.

Semantic Web: a theoretical initiative aimed at improving the semantic description of web-based resources in order to facilitate better knowledge discovery on the Internet.

SGF: *Structured Graph Format* - Defines an XML metadata format for exploration of overlapping hierarchies of content - especially websites.

SGML: *Standard Generalised Markup Language*. A universal syntax for defining markup language. A "meta-language".

SMIL: *Synchronized Multimedia Integration Language* - a multimedia document ontology for presentation of heterogeneous multimedia objects - it allows specification of timing rules (in parallel and sequence) for the multimedia objects.

SOAP: *Simple Object Access Protocol*. A protocol, now integral to Web Services, for process interaction with a Web site over standard HTTP communication channels.

- T -

TCP/IP: *Transmission Control Protocol/Internet Protocol*. The protocol-level for communication on an Internet. Defines the addresses to be used, the routing rules for traversal of the network and the protocols for file and data interchange.

TEI: *Text Encoding Initiative*. A key text markup standard for SGML markup of texts in the Humanities.

- U -

Unicode: An international standard for binary character set encoding of text in different languages.

Unix: an operating system orginiating in the 1960's and a popular platform for Internet applications. Linux is closely modelled on Unix.

URI*: Uniform Resource Identifier*. A generalised format for resource identification. A URL is a specific implementation of a URI.

URL: *Uniform Resource Locator*, the address of a document or other Internet resource - a particular instance of a URI for purposes of web-based addressing.

- V -

VRML: Virtual Reality Modelling Language**.** A generalised designer modelling language for constructing Virtual Reality environments – particularly for gaming or simulation.

- W -

W3C: *World Wide Web Consortium.* Responsible for publishing the Internet standards pertaining to the World Wide Web.

Web Services: A set of protocols called "Web Services" which enable the discovery and integration of business functions (for use by applications) and accessible through the internet.

WSDL: *Web Services Description Language.* An ontological specification language for Web Services.

WYSIWYG: *What You See is What You Get.* Multimedia content is edited on-screen with the markup hidden and presented as it would be finally published.

- X -

XHTML – *Extensible HyperText Markup Language.* An XML-compliant variant of HTML.

XML – *Extendsible Markup Language.* A popular implementation of SGML used for information exchange.

XSL – A set of standards for transforming XML into some final form. XSL defines a scripting language for style sheets (XSLT) that can transform an XML markup format to another format based on transformational rules, with the source XML and XSLT style sheets defined by XPATH (the workflow language of XSL).

- Z -

Z39.50: a network-aware OSI-based search engine used to share Digital Library collections. It defines query language properties and methods for persistence of searches.

BIBLIOGRAPHY

3rdGen. (2003a). 3rdgen Technical Description V7. Retrieved 14/4/2003, from http://www.3rdgen.info/resource/3rdgen_Technical_Description_v7.pdf

3rdGen (2003b). 3rdGen Quick Reference (Accessed 14/4/2003). 3rdGen. (2003). 3rdgen Quick Reference. Retrieved 14/4/2003, from http://www.3rdgen.info/resource/3rdgen_Quick_Reference_Guide_v2-2.pdf

Addey, D., Ellis, J., Suh, P., & Thiemecke, D. (2002). *Content Management Systems*. Birmingham (UK): Glasshaus.

Advanced Distributed Learning. (2001). Sharable Content Object Reference Model Version 1.1. Retrieved 6/8/2001, from http://www.adlnet.org/.

Agoulmine, N., Dragan, D., Gringel, T., Hall, J., Rosa, E., & Tschichholz, M. (2000). Trouble management for multimedia services in multi-provider environments. *Journal of Network and Systems Management, 8*(1), 99-123.

Althoff, K.-D., Bomarius, F., & Tautz, C. (2000). Knowledge Management for Building Learning Software Organisations. *Information Systems Frontiers, 2*(3/4), 349-367.

Anderson, K. M. (2001). Data scalability in open hypermedia systems. Paper presented at the *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots*, February 21 - 25, 1999, Darmstadt Germany.

Apache Software Foundation. *XML: Apache.Org*. Retrieved 18/3/2004, from http://xml.apache.org/index.pdf

Apache Software Foundation. *XML showcase environment*. XML Showcase Environment. Retrieved 18/3/2004, from http://cocoon.showcase.surfnet.nl/.

Baker, M. (2002). Structured Content: What's in It for Writers? Retrieved 22/4/2003, from http://www.cmswatch.com/Features/OpinionWatch/FeaturedOpinion/?feature_id=79.

Balnaves, E. (2003a). Escrow management of risks associated with digital resources. Paper presented at *the 4th Australian Information Warfare and Security Conference (AIWSC) 2003*, Adelaide.

Balnaves, E. (2003b). *Inter-Publish Designers Manual*. Retrieved 12/3/2004, from http://www.prosentient.com.au/docs/training%20manual.pdf

Balnaves, E. (2004). *Inter-Publish Self-Paced Training Manual*. Retrieved 14/3/2004, from http://www.prosentient.com.au/docs/training%20manual.pdf

Barnard, D. T., Burnard, L., Gaspart, J.-P., Price, L. A., Sperberg-McQueen, C. M., & Varile, G. B. (1995). Hierarchical Encoding of Text: Technical Problems and SGML solutions. *Computers and the Humanities, 29*, 211-231.

Barrett, C. (2001). *Camworld: Content Management Systems*. Retrieved 20 December 2002, from http://www.camworld.com/cms

Baskerville, R. L., & Wood-Harper, A. T. (1996). A Critical Perspective on Action Research as a Method for Information Systems Research. *Journal of Information Technology*, 11(3), 235-246.

Belle, J. (2001). Is Content King? *EContent, 24*(5), 38-43.

Bergeron, B. P., & Bailin, M. T. (1997). Managing the complexities of hypermedia development: challenges for on-line digital publishing. *Journal of Educational Multimedia & Hypermedia, 6*(3-4), 413-424.

Bergius, Henry (2003). *Aegir CMS - Default branch* (Accessed 14/8/2003). from http://freshmeat.net/projects/aegir/

Boiko, B. (2002). *Content Management Bible*. NY: Wiley.

Boll, S., & Klas, W. (2001). ZyX - A multimedia Document Model for Reuse and adaption of Multimedia Content. *IEEE Transactions on Knowledge and Data Engineering, 13*(3), 361-381.

Borenstein, N., & Freed, N. (1992). *RFC 1341: Mime (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies.* Retrieved 27/7/2001, from http://www.ietf.org/rfc/rfc1341.txt.

Boss, R. W. (1995). Technical Services Functionality in Integrated Library Systems: Standards and Interfaces. *Library Technology Reports, 31*(6), 764-771.

Boss, R. W. (2000). Information Technology Standards. *Library Technology Reports, 36*(4), 1-106.

Brett, P. A., & Nash, M. (1999). Multimedia language learning courseware: a design solution to the production of a series of CD-ROMs. *Computers & Education, 32*(1), 19-33.

British Council Australia. (1999). *Montage Requirements* (Internal specification report.). Sydney: British Council Australia.

Broder, A. Z. (1997). On the Resemblance and Containment of Documents. Paper presented at the *Compression and Complexity of Sequences (SEQUENCES'97)*, Salerno, Italy. 21-29.

Bullock, A., Stallybrass, O., & Trombley, S. E. (eds) (1977). *The Fontana Dictionary of Modern Thought*. London: Fontana.

Bulterman, D. (2001). SMIL 2.0. *IEEE Multimedia, 8*(4), 82-88.

Burnett, I., van de Walle, R., Hill, K., Bormans, J., & Pereira, F. (2003). MPEG-21 : Goals and Achievements. *IEEE Multimedia, Oct/Dec 2003*, 60-70.

Byrne, T. (2003a). *The CMS Report: Web Content Management Products and Practices* (4th Edition): CMS Watch.

Byrne, T. (2003b). *Cms Watch*. Retrieved 21/3/2003, from http://www.cmswatch.com.

Cameron, B., Beam, M., & Beam, P. (2000). Creation, costs and management: the measure of full-service online learning. *18th Annual Conference on Computer Documentation. ipcc sigdoc 2000. Technology and Teamwork. Proceedings. IEEE Professional Communication Society International Professional Communication Conference and ACM Special Interest Group on Documentation Conference (Cat. No.00CH37136). IEEE. 2000*, 99-109.

Candler, C. S., & Andrews, M. D. (1999). Avoiding the great train wreck: Standardizing the architecture for online curricula. *Academic Medicine, 74*(10), 1091-1095.

Casillas, A., Abaitua, J., & Martinez, R. Advantages and difficulties with TEI tagging: experiences from an aided document composition and translation tool. *Conference Proceedings. Extreme Markup Languages 2000. Expanding XML/SGML Universe. Graphic Commun. Assoc. 2000*, 29-35.

Celentano, A., Pozzi, S., & Toppeta, D. (1992). A Multiple Presentation Document Management System. Paper presented at the *Proceedings of the 10th annual international conference on Systems documentation (SIGDOC 92)*, Ottawa, Canada, 63-71.

Ceri, S., Fraternali, P., & Paraboschi, S. (1999). Data-Driven One-to-One Web Site Generation for Data-Intensive Applications. Very Large Data Bases. Proceedings of the *Twenty-Fifth International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers. 1999, 615-626.

Chamberlin, D., Clark, J., Florescu, D., Robie, J., Siméon, J., & Stefanescu, M. (2001). *Xquery 1.0: An XML Query Language - W3c Working Draft 07 June 2001*. Retrieved 23/10/2003, from http://www.w3.org/TR/xquery.

Cheung, S. C., & Chanson, S. (1999). A model-based authorware for the construction of distributed multimedia systems. *Information and Software Technology, 41*(11-12), 715-727.

Chung, J.-Y., Lin, K.-J., & Mathieu, R. G. (2003). Web Services Computing: Advancing Software Interoperability. *Computer, 36*(10), 36-37.

Clark, J. E. (1999). *XSL Transformations (XSLT) Version 1.0 (http://www.w3.org/TR/1999/REC-xslt-19991116)*

Clyman, J. (2002a). *Cms Directory*. Retrieved 29/9/2003, from http://www.pcmag.com/article2/0,4149,480784,00.asp.

Clyman, J. (2002b). *Vignette V6 Content Suite*. Retrieved 29/9/2003, from http://www.pcmag.com/article2/0,4149,479176,00.asp

Cocoon. (2003). *Apache Cocoon Gettogether*. Retrieved 11/8/2003, from http://apache.mirror.positive-internet.com/xml/cocoon/events/gt2003/presentations/.

Colbert, M., Peltason, C., Fricke, R., & Sanderson, M. (1997, August 18-20, 1997). The application of process models of information seeking during conceptual design: the case of an intranet resource for the re-use of multimedia training material in the motor industry. Paper presented at the *Conference on Designing interactive systems: processes, practices, methods, and techniques*, Amsterdam, the Netherlands.

Coombs, J. H., Renear, A. H., & DeRose, S. J. (1987). Markup systems and the future of scholarly text processing;. *Communications of the ACM, 30*(11), 933-947.

Cope, J., Craswell, N., & Hawking, D. (2003). Automated Discovery of Search Interfaces on the Web. Paper presented at *The Fourteenth Australasian Database Conference (ADC 2003)* February 4 - 7, 2003, Adelaide, Australia.

Cox, J. E. (1998). The Changing Economic Model of Scholarly Publishing: Uncertainty, Complexity, and Multimedia Serials. *Library Acquisitions: Practice & Theory, 22*(2), 161-166.

Craswell, N., Hawking, D., & Robertson, S. (2001). Effective site finding using Link Anchor Information. Paper presented at the *24th annual ACM SIGIR conference 2001 (SIGIR'01)* September 9-12, New Orleans, Louisiana, USA.

Crawford, W., & Gorman, M. (1995). *Future Libraries: Dreams, Madness and Reality*. Chicago: American Library Association.

Crespo, A., & Garcia-Molina, H. (1998). *Archival Storage for Digital Libraries*. Paper presented at the *Third ACM International Conference on Digital Libraries*, Pittsburgh, Pa. 69 - 78.

Crocker, D. H. (1982). *RFC* 0822: Arpa Internet Text Messages, 13/7/2000, from http://www.ietf.org/rfc/rfc0822.txt.

Cybulski, J. (1996). *Introduction to Software Reuse* (Technical Report TR 96/4). Melbourne, Australia.: University of Melbourne.

Dagtas, S., Al-Khatib, W., Ghafoor, A., & Kashyap, R. L. (2000). Models for motion-based video indexing and retrieval. *IEEE Transactions on Image Processing, 9*(1), 88-101.

Dahl, K., & Hedberg, S. (1996). Electronic Publishing: availability and legal deposits. *Publications of Essen University Library, 20*, 119-131.

Dalton, J. P., Manning, H., & Gardiner, K. (2001). *Managing Content Hypergrowth*. Cambridge, MA: Forrester Research.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly, September 1989*, 318-341.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User Acceptance Of Computer Technology: A Comparison Of Two Theoretical Models. *Management Science, 35*(8), 982-1004.

Davis, J., Subrahmanian, E., Khonda, S., Granger, H., Collins, M., & Westerberg, A. W. (2001). Creating Shared Information Spaces to Support Collaborative Design Work. *Information Systems Frontiers, 3*(3), 377-392.

Day, Y. F., Liu, P., & Hsu, L. H. (2001). Transforming Large-scale Product Documents into Multimedia Training Manuals. *IEEE Multimedia, 8*(3), 39-45.

D.C.M.I. (2001). *Dublin Core Metadata Initiative*. Retrieved 14/7/2002, from http://www.dublincore.org.

DeRose, S. J. (1999). XML and TEI. *Computers and the Humanities, 33*, 11-30.

Di Iorio, E., Diligenti, M., Gori, M., Maggini, M., & Pucci, A. (2003). Detecting near-replicas on the Web by content and hyperlink analysis. Paper presented at the *IEEE International Conference on Web Intelligence (WIC 2003)*, October 13-17 2003, Halifax, Canada.

Dorai, C., & Venkatesh, S. (2001). Computational Media Aesthetics: Finding Meaning Beautiful. *IEEE Multimedia, 8*(4), 10-12.

Documentum. (2003). *Documentum Server Ecu Platform*. Retrieved 14/4/2003, from http://www.documentum.com.

Downs, R. R. a. F., Edward A. (1999). Digital library support for scholarly research. *Information Processing & Management, 36*(3), 281-291.

Duffy, K. (2000). Content management comes of age: streamlining the information creation and delivery process. *Information Management & Technology, 33*(4), 183-191.

Eaves, M. (1997). Behind The Scenes At The William Blake Archive: Collaboration Takes More Than E-mail. *The Journal of Electronic Publishing, 3*(2), http://www.press.umich.edu/jep/03-02/blake.html.

Ekman, R. H. (2000). Can Libraries of Digital Materials Last Forever? *Change, 32*(2), 23-35.

Erdmann, M., & Struder, R. (1999). Ontologies as Conceptual Models for XML Documents. Paper presented at the *KAW'99: Twelfth Workshop on Knowledge Acquisition, Modeling and Management*, Voyager Inn, Banff, Alberta, Canada: Saturday 16 to Thursday 21 October, 1999.

Ewusi-Mensah, K. (1997). Critical issues in abandoned information systems development projects. *Communications of the ACM, 40*(9), 74-80.

Fensel, D., van Harmelen, F., Horricks, L., McGuinness, D. L., & Patel-Schneider, P. F. (2001). OIL: an ontology infratructure for the Semantic Web. *IEEE Intelligent Systems, 16*(2), 38-45.

Fetzer, J. H. (1999). The Role of Models in Computer Science. *The Monist, 82*(1), 20-36.

Franks, J., Hallam-Baker, P., Hostetler, J., Lawrencem, S., Leach, P., Luotonen, A., et al. (1999). *RFC 2617: HTTP Authentication: Basic and Digest Access Authentication*. Retrieved 14 November 2001, from http://asg.web.cmu.edu/rfc/rfc2617.html.

Fraser, S. R. G. (2002). *Real-World ASP.Net:Bulding a Content Management System*. Berkeley, CA: Apress.

Fraternali, P. (1999). Tools and approaches for developing data-intensive Web applications: a survey. *ACM Computer Survey, 31*(3), 227 - 263.

Fraternali, P., & Paolo, P. (2000). Model-Driven Development of Web Applications: The Autoweb System. *ACM Transactions on Information Systems, 18*(4), 323-341.

Galliers, R. D. (1995). A manifesto for Information Management Research. *British Journal of Management, 6*(Special Issue), S45-S52.

Garzotto, F., Paolini, P., & Schwabe, D. (1993). HDM—a model-based approach to hypertext application design. *ACM Transactions on Information Systems, 11*(1), 1 - 26.

Geng, X., Gopal, R. D., Ramesh, R., & Whinston, A. B. (2003). Scaling Web Services with Capacity Provision Networks. *Computer, 36*(11), 64-72.

Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). *RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1*. Retrieved 14/11/2001, from http://www.w3.org/Protocols/rfc2616/rfc2616

Goland, Y. Y., Whitehead, E., Faizi, A., Carter, S., & Jensen, D. (1999). *RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV:*. Retrieved 14/11/2001, 2001, from http://asg.web.cmu.edu/rfc/rfc2518.html.

Gruber, T. R. (1993). A Translation Approach to Portal Ontology Specifications. *Knowledge Acquisition, 5*(2), 199-220.

Gu, A., Henderson-Sellers, B., & Lowe, D. (2002). Web Modelling Languages: The Gap Between Requirements and Current Exemplars. Paper presented at *The Eighth Australian World Wide Web Conference*, Sunshine Coast Queensland from Saturday 6 July to Wednesday 10 July 2002.

Guenther, R. (1997). *RFC 2220: The Application/Marc Content-Type*. Retrieved 15/7/2001, from http://www.ietf.org/rfc/rfc2220.txt.

Hackos, J. T. (2001). *Content Management for Dynamic Web Delivery*. NY: Wiley.

Hammer, E. M., & Zalta, E. N. (1997). A Solution to the Problem of Updating Encyclopedias. *Computers and the Humanities, 31*(1), 47-60.

Haveliwala, T. H., Gionis, A., Klein, D., & Indyk, P. (2002). Evaluating Strategies for Similarity Search on the Web. Paper presented at *WWW2002*: *The Eleventh International World Wide Web Conference*, Honolulu, Hawaii, USA.

Haustein, S., & Pleumann, J. (2000). Easing Participation in the Semantic Web. Paper presented at the *Semantic Web Workshop 2002*, Hawaii, USA.

Haustein, S., & Pleumann, J. (2002). Is Participation in the Semantic Web Too Difficult? Paper presented at *The Semantic Web - First International Semantic Web Conference*, Sardinia, Italy.

Heflin, J., & Hender, J. (2000). Semantic Interoperability on the Web. Proceedings of *Extreme Markup Languages 2000*, Montreal, Quebec. 111-120.

Heitmann, J. (1999). Content management systems for television production. *EBU Technical Review*(280), 24-34.

Hirschheim, R., & Klein, H. K. (1994). Realizing emancipatory principles in information systems development: The case for ETHICS. *MIS Quarterly, 18*(1), 83-109.

Hirschheim, R., Klein, H. K., & Lyytinen, K. (1995). *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge: Cambridge University Press.

Hjelmslev, L., & Whitfield, F. J. (1961). *Prolegomena to a theory of language* (Rev., English ed.). Madison University of Wisconsin Press.

Hoffman, P. (2000). Internationalization of Internet Domain Names. Paper presented at the *17th International Unicode Conference*, San Jose, California.

Hofstadter, D. R. (1979). *Gödel,Escher,Bach: an eternal golden braid*. Harmondsworth, Middlesex, England: Penguin.

Holdcroft, D. (1991). *Saussure : signs, system, and arbitrariness*. New York: Cambridge University Press.

Hunt, J. J., & Reuter, J. (2001). Using the Web for Document Versioning: An Implementation Report for DeltaV. Paper presented at the *23rd International Conference on Software Engineering (ICSE)*. 12-19 May 2001., Toronto, Ontario.

Ide, N. M., & Sperberg-McQueen, C. M. (1995). *The TEI: History, Goals and Future*. Computers and the Humanities, *29*, 5-15.

IEEE. (2002, 15/7/2002). *Draft Standard for Learning Object Metadata*. Retrieved 23/11/2003, from http://grouper.ieee.org/p1484/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Iivari, J., Hirschheim, R., & Klein, H. K. (1998). A paradigmatic analysis contrasting information systems development approaches and methodologies. *Information Systems Research, 92*(2), 164-193.

InterDoc Corporation. (2002). *Comparitive Analysis of Reddot Cms Professional 4.5 and Microsoft Cms 2002*. Technical Report November 2002, 1/11/2003, from www.interdoc.com

International Organization for Standardization (ISO). (1986). *ISO 8879:1986(E). Information processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)*. (First edition ed.). Geneva: International Organization for Standardization.

ISO. (1996). *ISO/IEC 10179 (1996): Information Technology - Processing Languages - Document Style Semantics and Specification Language*. (ftp://ftp.ornl.gov/pub/sgml/WG8/DSSSL/dsssl96b.pdf).

Jewitt, C. (2000). Digital chaos and professional standards. *Multimedia Information and Technology, 26*(2), 148 - 150.

Jones, M., & Webb, C. (1998). *Electronic Publications and the Survival of Information*. Paper presented at the *`Robots to Knowbots: the wider automation agenda': Victorian Association for Library Automation 9th biennial conference*, Melbourne, Victoria. p.313-324.

Kartchner , C. (1988). Getting from Concept to Reality. *Journal of Electronic Publishing*., 3(4), Retrieved 13/10/2003, from http://www.press.umich.edu/jep/2003-2004/kartchner.html.

Kelly, B. (2000). Will it all fit together? The need for standards and the technical challenges. *Multimedia Information and Technology, 26*(1), 67-72.

Kirda, E. (2001). Web engineering device independent Web services. Proceedings of the *23rd International Conference on Software Engineering. ICSE*.

Krueger, C. W. (1992). Software Reuse. *ACM Computing Surveys, 24*(2), 131-183.

Kuhlthau, C. C. (1993). *Seeking a Meaning: A process Approach to Library and Information Services*. NJ: Norwood.

Kunze, J. (1999). *RFC 2731: Encoding Dublin Core Metadata in HTML*. Retrieved 15 Jan 2001, from http://www.ietf.org/rfc/rfc2731.txt

Lancaster, F. W. (1968). Interaction between requesters and a large mechanized retrieval system. *Information Storage and Retrieval, 4*(2), 239-252.

Lassila, O., & Swick, R. (1999). *Resource Description Framework: (Rdf) Model and Syntax Specification -W3c Recommendation 22 February 1999*. Retrieved 05/03/2003, from http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

Lauer, T. W. (2001). Questions and Information: Contrasting Metaphors. *Information Systems Frontiers, 3*(1), 41-48.

Lawrence, S., Pennock, D. M., Rovetz, R., Coetzee, F. M., Glover, E., Nielsen, F. A., et al. (2001). Persistence of Web References In Scientific Research. *Computer, 34*(2), 26-31.

Li, R. (2003). *Redhat Content Management System Overview*. Retrieved 14/4/2003, from http://www.redhat.com

Libby, D. (1999). *Rss 0.91 Spec, Revision 3 (Http://Mynetscape.Com/Publish/Formats/Rss-Spec-0.91.Html)*. Retrieved 05/03/2002, from http://mynetscape.com/publish/formats/rss-spec-0.91.html.

Liebowitz, J., & Wilcox, L. C. (1997). *Knowledge management and its integrative elements*. Boca Raton, Fla.: CRC Press.

Liu, P. a. H., Liang H. (1999). Multimedia Document Systems From Authoring perspective. In B. Furht (Ed.), *Internet and Multimedia: Systems and Applications*: IEEE Press.

Love, P., Irani, Z., Warren, M., Coldwell, J., & Warren, S. (2001). *Barriers to implementing Ecommerce in construction SME's*. Melbourne: Deakin University.

Lu, S., Dong, M., & Fotouhi, F. (2002). The semantic web: Opportunities and challenges for next-generation web applications. *International Journal of Information Research, 7*(4).

M.E.S.O. (1995). *Multimedia Educational Software Observatory*. Retrieved 15/12/2002, from http://europa.eu.int/comm/education/meso/

Maeda, A., Dartois, M., Fujita, T., Sakaguchi, T., Sugimoto, S., & Tabata, K. (1998). Viewing multilingual documents on your local Web browser. *Communications of the ACM, 41*(4), 63-65.

Marchionini, G., Plaisant , Catherine and Komlodi, Anita. (1998). Interfaces and tools for the Library of Congress National Digital Library program. *Information Processing & Management, 34*(5), 535-555.

McGann, J. (1998). Textual Scholarship, Textual Theory, and the Uses of Electronic Tools: A Brief Report on Current Undertakings. *Victorian Studies, 41*(4), 609-6190.

Mendes, M. A. S., & Almeida, V. A. F. (1998). Analyzing the impact of dynamic pages on the performance of Web servers. Paper presented at the *CMG Proceedings, Proceedings of the 1998 24th International Conference for the Resource Management & Performance Evaluation of Enterprise Computing Systems*, CMG. Part 1 (of 2), Dec 6-11 1998, Turnersville, NJ, USA.

Merialdo, P., Atzeni, P., & Mecca, G. (2003). Design and development of data-intensive web sites: The Araneus approach. *ACM Transactions on Internet Technology, 3*(1), 49 - 92.

Michalas, V. (2000). A Comparison of Multimedia Application Development Platforms Towards the Object Web. *Computer Standards & Interfaces*, 22(1), 13-26.

Microsoft. (2002). *Microsoft Content Management Server*: Product Overview. Retrieved 11/2/2002, from http://www.microsoft.com/cmserver/default.aspx ?url=/cmserver/evaluation/overview/

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: an expanded sourcebook*. Thousand Oaks, California: Sage.

Moczar, L., & Aston, J. (2003). *Cocoon Developers Handbook*. Indianapolis, Indiana: Developers Library.

Nahm, U. Y., & Mooney, R. J. (2002). Text Mining with Information Extraction. Paper presented at the *AAAI-2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, March 2002, Palo Alto, California.

Nebel, E., & Masinter, L. (1995). *RFC 1865 - Form-based File Upload in HTML*. Retrieved 12/12/2001, from http://www.ietf.org/rfc/rfc1867.txt

Needleman, M. (2000). Z39.50 - a review, analysis and some thoughts on the future. *Library Hi Tech, 18*(2), 158-165.

Neigus, N. (1973). *RFC 0542: File Transfer Protocol*. Retrieved 21/8/2001, from http://asg.web.cmu.edu/rfc/rfc542.html

Newman, J. (2001). Some observations on the Semantics of "Information". *Information Systems Frontiers, 3*(2), 155-167.

Newman, M., & Sabherwal, R. (1996). Determinants of commitment to information systems development: A longitudinal investigation. *MIS Quarterly, 20*(1), 23-54.

Norrie, M. C., & Signer, B. (2003). Issues of Information Semantics and Granularity in Cross-Media Publishing. Paper presented at the *CAiSE'2003, 15th International Conference on Advanced Information Systems Engineering*, June 2003, Klagenfurt/Velden, Austria.

Nunamaker, J. F. J., Chen, M., & Purdin, T. D. M. (1991). System development in information systems research. *Journal of Management Information Systems, 7*(3), 89--106.

O.E.C.D. (1998). Content As A New Growth Industry. Paper presented at the *European Audiovisual Conference* 6-8 April 1998, Birmingham.

Obtree Technologies Inc. (2002). *Obtree C4 Data Sheet 2002*. Retrieved 11/12/2002, from http://www.obtree.com

Obtree Technologies Inc. (2002). *Obtree C4 Product Brouchure 2002*. Retrieved 11/12/2002, from http://www.obtree.com

Obtree Technologies Inc. (2003). *C4 Obtree Technical White Paper. 2003*. Retrieved 12/12/2003, from http://www.obtree.com.

Odlyzko, A. (2001). Content is not king. *First Monday, 6*(2), 1-28.

Ogbuji, U. (2003). *The Past, Present and Future of Web Services*. Retrieved 7/12/2003, from http://webservices.org.

Open eBook Forum. (2001). *Open Ebook™ Publication Structure 1.0.1*. Retrieved 12/12/2003, from http://www.openebook.org/oebps/oebps1.0.1/download/2001-07-02_OEBPS1.0.1_recommended/hoeb101.htm.

Overby, E. (1999). Problems with linking, and reuse of text: why publishing on different targets might not be as easy as it first seems. Proceedings of *XML EUROPE '99. OASIS. W3C World Wide Web Consortium*. 26-30 April 1999., Granada, Spain.

Page, L., Brin, S, Motwani, R and Winograd, T. (1998). *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford Digital Libraries Working Paper. Stanford, Computer Science Department, Stanford University.

Patrick, J., Balnaves, E., Kam, E., Ler, D., & Yeates, T. (2000). A Case Study of multimedia Architecture Design: workflow and version control for regeneration of multimedia systems. Paper presented at the *AWSAA3: The Third Australasian Workshop on Software and System Architectures*, Sydney.

Perez, M., & Rojas, T. (2000). Evaluation of Workflow-type software products: A case study. *Information and Software Technology, 42*(7), 489-503.

Phanouriou, C., Kipp, N. A., Sornil, O., Mather, P., & Fox, E. A. (1999). A Digital Library for Authors: Recent Progress of the Networked Digital Library of Theses and Dissertations. Paper presented at the *Proceedings of the Fourth ACM International Conference on Digital Libraries*, Berkeley, California.

Phillips, M. E. (1998). Tomorrow's incunabula: preservation of Internet publications. *Lasie, 29*, 5-10.

Prasad, S., Babbar, S., and ,Calis, A. (2000). International Operations Management and Operations Management Research: a comparative analysis. *Omega, The International Journal of Management Science, 28*, 97-110.

Prieto-Diaz, R. (ed.). (1989). *Classification of reusable modules*. New York: Addison-Wesley.

PRISM Working Group. (2001). Prism: Publishing Requirements for Industry Standard Metadata. Version 1.0. Retrieved 20/7/2002, from http://prismstandard.org/

Raggett, D., Le Hors, A., & Jacobs, I. (1999, 24/11/2003). *HTML 4.01 Specification*. Retrieved 23/11/2003, from http://www.w3.org/TR/REC-html40/

RedDot Solutions. (2003). *Reddot Enterprise Content Management Suite. Product Overview*. Retrieved 11/7/2003, from http://www.reddot.co.uk

RedDot Solutions. (2003). *Reddot Ecms - Technical Data - Status 3/2003*. Retrieved 11/7/2003, from http://www.reddot.co.ukhttp://www.reddot.co.uk

RedDot Solutions(2003). *RedDot Integration with SAP* (Accessed 11/7/2003). RedDot Solutions. (2003). Reddot Integration with Sap. Retrieved 11/7/2003, from http://www.reddot.co.uk

Renear, A., Durand, D., & Mylonas, E. (1996). Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies. In *Research in Humanities Computing*: Oxford University Press.

Renear, A., Hockey, S., & McGann, J. (1999). Panel: What is text? A debate on the philosophical and epistemological nature of text in the light of humanities computing research. Paper presented at the *ACH-ALLC '99: Association for Computers and the Humanities and the Association for Literary and Linguistic Computing Joint annual Conference*, June 9-13, 1999 in Charlottesville, Virginia.

Reynolds, J., & Kaur, A. (2000). *Content Management*. Retrieved 25/7/2001, from http://www.Microsoft.com.technet/ecommerce/contmgt.asp.

Richards, J. D. (1997). Preservation and re-use of digital data: the role of the Archaeology Data Service. *Antiquity*, *71*(274), 1057-1059.

Rockley, A. (2003). *Managing Enterprise Content: A unified content strategy*. Boston: New Riders.

Rutledge, L. (2001). SMIL 2.0: XML for Web Multimedia. *IEEE Internet Computing, 5*(5), 78-84.

Saddik, A., Fischer, S., & Steinmetz, R. (2001). Reusable Multimedia Content in Web-Based Learning Systems. *IEEE Multimedia, 8*(3), 30-38.

Santini, S. J., R. (1998). Beyond Query by Example. Paper presented at *ACM Multimedia 98*, Bristol, England, Sept 12-16 1998.

Schuster, E., & Wilhelm, S. (2000). Fundamentals of Content Management Systems (Cms). *Informatik Spektrum*, 23(6), 373-375.

Semanticweb.Org. (2003, 14/11/2003). *Ontologies and Markup Languages*. Retrieved 14/11/2003, 2003

Senn, J. (1998). The challenge of relating IS research to practice. *Information Resources Management Journal, 11*(1), 23-28.

Sherif, K., & Vinze, A. (2003). Barriers to adoption of software reuse: A qualitative study. *Information & Management, 41*(2), 159-175.

Shneiderman, B., Byrd, D., & Croft, W. B. (1998). Sorting out Searching: a user-interface framework for text searches. *Communications of the ACM, 41*(4), 96-98.

Silver, B. (2001). When Will Documents Become Content? *Transform Magazine*, 10(7), 23.

Smith, W. (1997). *Pandora-boxing for survival. Archiving preservation and access issues related to Australian Internet based publications.* Paper presented at *`On the edge': The Seventh Asian Pacific Specials, Health and Law Librarians' Conference.*, Perth, Australia.

Smith, W. (1998). Lost in cyberspace: preservation challenges of Australian Internet resources. *Lasie, 29*, 6-25.

Song, Y., Clayton, M. J., & Johnson, R. E. (2002). Anticipating reuse: documenting buildings for operations using web technology. *Automation in Construction, 11*(2), 185-197.

Sonnenwald, D. H., & Pierce, L. G. (2000). Information Behavior in Dynamic Group Work Contexts: Interwoven Situational Awareness, Dense Social Networks and Contested Collaboration in Command and Control. *Information Processing & Management*, 36(3), 461-479.

Souzis, A., Popkin, L., Khoury, S., & Hunt, B. (2000). Ice Implementation Cookbook: Getting Started with Web Syndication. Retrieved 17/7/2001, from www.idealliance.org.

Sowa, J. F. (2000). Ontology, Metadata, and Semiotics. In G. Ganter & W. Mineau (Eds.), *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Berlin: Springer-Verlag.

Sperberg-McQueen, C. M., Huifeldt, C., & Renear, A. (2000). Meaning and Interpretation of Markup. *Markup Languages: Theory & Practice, 2*(3), 215-234.

St.Laurent, S., Jonston, J., & Dumbill, E. (2001). *Programming Web Services with XML-RPC*. Sebastapol, California: O'Reilly.

Staab, S., & Maedche, A. (2001). Knowledge Portals: Ontologies at Work. *AI Magazine, 22*(2), 63-72.

Stahl, F. (2003). Empirical Aspects on Content Management. Paper presented at the *Third Open Source Content Management Conference*, Cambridge, Massachusetts (Harvard Law School).

Steinacker, A., Ghavam, A., & Steinmetz, R. (2001). Metadata Standards for Web-based Resources. *IEEE Multimedia, 8*(3), 70-76.

Stohr, E. A., & Zhao, J. L. (2001). Workflow Automation: Overview and Research Issues. *Information Systems Frontiers, 3*(3), 281-296.

StrongPoint (2003). Product Sheet overview (Accessed 12/5/2003). *StrongPoint. (2003).* Product Sheet Overview. Retrieved 12/5/2003, from http://www.strongpoint.com.au

StrongPoint.(2003). Page Life Cycle (Accessed 12/5/2003). *StrongPoint. (2003). Page Life Cycle. Retrieved* 12/5/2003, from http://www.strongpoint.com.au

StrongPoint(2003). CMS Pricing (Accessed 12/5/2003). *StrongPoint. (2003). Cms Pricing.* Retrieved 12/5/2003, from http://www.strongpoint.com.au .

Suzuki, M., Inoue, N., & Hasimoto, K. (2001). A method for supporting Document Selection in Cross-language Information Retrieval and its Evaluation. *Computers and the Humanities, 35*, 421-428.

Thomas, P. (2000). Arbeitsabläufe unter integriertem Content Management (Work processes of integrated content management). *Rundfunktechnische Mitteilungen, 44*(1), 23-29.

Tilley, S., & Huang, S. (2001). Evaluating the Reverse Engineering Capabilities of Web Tools for Understanding Site Content and Structure: A Case Study. Paper presented at the *Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on Software Engineering*, Ontario, Canada. May 12--19. 514-523.

Udell, J. (2003). Databases get an XML infusion. *Information Age*, 61-64.

W3C URI Planning Interest Group. (September 2001). *URIs, URLs, and URNs: Clarifications and Recommendations 1.0:Report from the joint W3C/IETF URI Planning Interest Group*. Retrieved 17/7/2003, from http://www.w3.org/TR/2001/NOTE-uri-clarification-20010921/.

Walter, M. (1999). Infooffice's Reddot: WYSIWYG Editing in a Mid-Level Web Publishing System. *Seybold Report on Internet Publishing*, 4(3), 3-6.

Weiderhold, G. (1995). Digital libraries, value and productivity. *Communications of the ACM, 38*(4), 85-97.

Whitehead, J., & Wiggins, M. (1998). WebDAV: IETF Standard for Collaborative Authoring on the Web. *IEEE Internet Computing*, 2(5), 34-40.

Wilkinson, R. (1998). *Document computing : technologies for managing electronic document collections*. Boston: Kluwer Academic Publishers.

Williamson, T. (2001). *RE: [cms-list] Reddot and NCompass*. Retrieved 23/11/2002, from http://cms.filsa.net/archives/cms-list/2001/Jun/0027.html

Wilson, D. W. (1998). Ontological pluralism and information systems research. *Systemist, 20*(spec. issue.), 260-270.

Wilson, J. (1997). Electronic document management; Electronic documents. *Computer Law & Security Report, 13*(2), 124-125.

Wood, M. E. J., Campbell, N.W. and Thomas, B.T. (1998). Iterative Refinement of Relevance Feedback in Content-based Digital Image Retrieval. Paper presented at the *ACM Multimedia 98*, Bristol, England, Sept 12-16 1998, Bristol.

Wright, M. A. (1996). Automating the business office. *Patient Accounts, 19*(10), 2-4.

Wynekoop, J. L., & Russo, N. L. (1997). Studying system development methodologies: An examination of research methods. *Information Systems Journal, 7*(1), 47-65.

Wyona.org. (2003). *Lenya Roadmap*. Retrieved 10/9/2003, from http://www.wyona.org/roadmap.html.

Xu, L. D. (2000). The contribution of systems science to information systems research. *Systems Research & Behavioral Science, 17*(2), 105-116.

Yeh, J. H., Chang, J. Y., & Oyang, Y. J. (2000). Content and Knowledge Management in a Digital Library and Museum. *Journal of the American Society for Information Science*, 51(4), 371-379.

Yuan, L., Abiza, Y., & Karmouch, A. (1999). Self-Guided Multimedia Courseware System over the Internet. Paper presented at the *IEEE Canadian Conference on Electrical and Computer Engineering*, May 9-12 1999, Edmonton, Alberta.