

The University of Sydney
AUSTRALIA

INFORMATION SYSTEMS IN THE ARTS AND HUMANITIES

2007

Dr Edmund Balnaves

© Edmund Balnaves 2001, 2004, 2006,2007

TABLE OF CONTENTS

Introduction

Chapter 1: Systems Thinking

Chapter 2: Content Representation Systems

Chapter 3: Content Mark-up

Chapter 4: Scripting Languages

Chapter 5: Content Delivery Systems

Chapter 6: Information Retrieval Systems

Chapter 7: Research Methods for Online Audience Analysis

Chapter 8: “In the Field”

Chapter 9: Intellectual Property Considerations

Index & Glossary

INTRODUCTION

> "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…" The Metamorphoses of Ovid. Book XII

Information Systems now touch all aspects of creative life. Three realms of computing have come together to transform the use, accessibility and relevant of Information Systems to aspects of life:

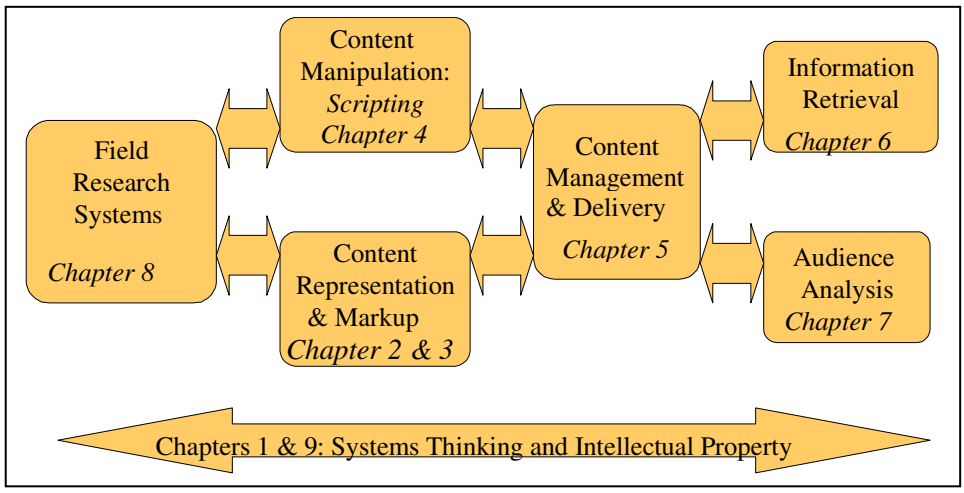
Personal Computers – providing a platform for both creation and delivery of digital systems and experiences.

Networking (and the Internet) – providing effective sharing of information, digital systems and experiences among many users, and pervasive distribution of information resources through searching (eg Google) and syndication (eg RSS) systems.

Richly linked Multimedia delivery – delivering information in an intuitive manner both in representation (images, video and animation enhancing textual content) and in navigation.

Taken together, these three realms have synergistically lent each other an impressive momentum. The ubiquitous web browser and hyperlinked navigational flow have transformed information delivery. Equally important, however, is the ready access to information systems, authoring tools and digital communications technology. Whether the creative individual concerns themselves with the systems that surround their activity, the reality Information Systems will invariably affect aspects of the communication and preservation of their work. Far from distaining Information Systems, many in the Arts and Humanities have actively employed new technological options as a means to their creative end.

This book explores the conceptual elements of the use of Information Systems in the Arts, Literature and Humanities – as distinct from business and government. Content creators and users have not been slow to exploit the potential of this digital convergence. However, there are some common Systems disciplines that are explored in this work that are directed at enhancing the utility of these systems in the Arts and Humanities.



This explores the application of Information Systems to the Arts and Humanities: known as *Arts Informatics*. The first theme of this work is *Systems Thinking,* explored in detail in Chapter One. The purpose of this exploration is to set a framework for understanding software and systems development in a "holistic" framework: that is, tempered by an understanding of the system relationships, feedback cycles, cultural behaviours and other nuances that surround the placement of digital systems in any context.

The second theme of the work is the exploration of *Representational and Markup Systems (Chapter 2). Representational Systems* are those that deliver digital multimedia information and content in one form or another - from CD-Rom-based delivery to Internet based information sites. The technology for *Representational Systems* has itself moved apace in the last two decades, but there are some common threads that can be found: the emergence of common delivery platforms such as the *Internet Browser* which brings together digital text, audio, animation and images; and the definition of image and text representational standards that can transcend the particular technological platform.

Integral to capitalising on new content development technologies are embedded S*cripting Languages.* Scripting languages enable the rapid development of enhanced capabilities in the creative and analytical endeavour. Chapter 4 explores Python and Microsoft Active Server Pages as examples of scripting languages that have powerful text parsing, Internet communication and image manipulation capabilities.

These technologies, used in conjunction with the *Databases* and the *Content Server (Chapter 5)* provide a powerful framework for the deployment of complex systems in a manner which is cost economical and accessible for content developments. The *Content Server* is the key information distribution vehicle for online servers, including both the internet web server (for text images and animation) and the media streaming server (for audio and video).

The technological foundations offer many opportunities in Arts and Humanities, and Chapter 7 and 8 explore research methods into the effectiveness of online systems and methods by which new technologies have been deployed in the Arts/Humanities, systems that in many cases enhance or transform approaches to the endeavours in their discipline.

Technological convergence is everywhere apparent, and examples are easy to cite: online database systems are now collecting full-text content, not just citations and meta-data; the growth in availability of full-text content is bringing in technology for textual analysis and data mining; likewise online textual content requires use of document management and publishing tools, which themselves bring elements of information retrieval technology. The new technology present challenges to many aspects of traditional learning and business and information delivery. The final chapter of the book explores approaches to Intellectual Property protection in the context of new media information systems.

Chapter 1: Systems Theory and Information Systems Practice

Introduction

A genius of humanity lies in our ability to collaborate in the management of our environment. This collaboration is realised and structured by means of systems of thought, behaviour and organisation. Societies are founded on innumerable formal and informal systems of behaviour that tie communities together. We can go back in time to *Stonehenge* in England, or the *Vedic Scriptures* in India to see the efforts to render a model of the universe and the systems of seasons and life: abstract models realised from observed natural cycles.

Sometimes the existence of an underlying system is not obvious: thousands of apparently unrelated events may be occurring, each of which apparently bear little relationship with each other. Taken as a whole, however, we may see a very apparent systematic behaviour. This pattern recognition is very fundamental to the human desire to work as part of a community.

So what is a system? A "system" is really a way of describing the coherent relationships and interactions between a discrete set of parts. Coherency is an important element, because without behavioural coherency of some sort, a thing cannot truly be described as a "system".

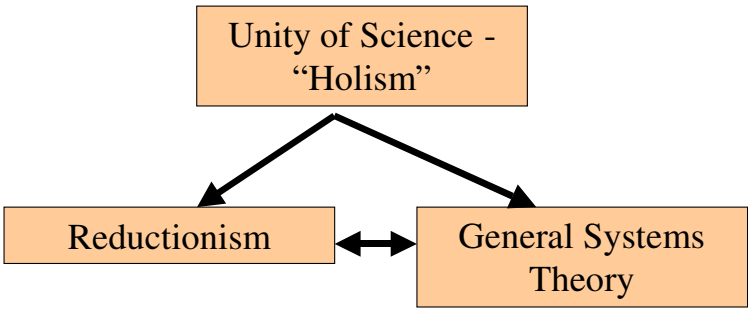
Understanding these relationships, and endeavouring to create more effective, productive and beneficial systems is what is called "systems thinking".

Science itself has long striven to dissect the elements of systems in order to understand better the "whole". Indeed a behavioural tendency in most disciplines has been the greater specialisation of thought as each discipline matures and builds more complex bodies of theoretical literature and understanding. This specialisation has been useful for the focus that it has given each discipline. It can also be at the expense of a renewed understanding of the original "whole".

In the 1960's von Bertalanffy(1968) sought to reverse the trend toward specialisation by broadening the discourse about systems to embrace all scientific disciplines: in other words, a holistic view of the Scientific endeavour.

In his *General Systems Theory* (GST, but not to be confused with the *Goods and Services Tax*) , he sought to provide a framework for coherent understanding of the relationship between the discrete specialisations. In particular he sought to achieve:

- Better integration of the various sciences
- Centring this integration in a GST
- Bringing more concepts across disciplines, and in particular from the physical sciences to the social sciences
- A move toward unifying the underlying principles in each discipline.



We have, in *General Systems Theory* the essential kernel of *Systems Thinking*. That is to recognise not only the individual components and systems, but also the broader interaction between these components and systems.

Systems Theory has played a large role in Information Science as a key tool in the development and deployment of Computer Applications. To build and deploy Applications it has been necessary to describe and model the behaviour of organisational systems. These models were necessary to:

- Ensure that the processes that needed to be "automated" as an information system in fact accurately described what was happening in the organisation

- Represent such processes in a manner that could be understood by programmers who did not necessarily have a fundamental understanding of the systems themselves.

From this emerge issues of system classification, systems analysis and design, system representation and modelling, system optimisation and production control.

The *paradigm* of system thinking has emerged from the more technical arena of building computer systems in parallel with a variety of business theories and approaches to organisational management.

All aspects of the formal and informal systems in business operation have been discussed and analysed. From Peter Drucker's efforts to identify the key factors in successful corporate management have evolved approaches to comprehensive internalisation of Information Systems in the operation

of a business - from *Business Process Reengineering (BPR)* in the 1990's to enterprise logistics offered by *Enterprise Resource Planning,*. Their goal is the formation of Information Systems which:

- reflect (rather than conflict with) the fundamental objectives of the organisation.
- are flexible and amenable to change (in an environment in which change often equates to business survival).

System Thinking

Peter Senge, in 1992, released his work "The fifth discipline." In it identifies a key goal for System Thinking: the development of the "learning organisation." Metaphorically, the purpose of being a learning organisation is survival – because to fail to learn is to fail to thrive. Such a paradigm of behaviour is reflected in all aspects of recent western social thinking: the analytical capability given by computer systems (and the speed with which organisations and systems *can* be changed) create a momentum that is self-sustaining. The importance of Senge's five disciplines of System Thinking are to go beyond the analytical issues of System Description and touch on the philosophical elements that can make System Thinking an effective way of thought.

Senge describes five disciplines of thought:

- Personal Mastery
- Mental Models
- Shared Vision
- Team Learning
- Systems Thinking

In as sense, systems theory taken as a "way of thought" is *System Thinking*.

Personal Mastery

Personal Mastery is the process of "learning to expand our personal capacity to create the results we most desire".

> "personal mastery means not only having and understanding personal vision, but realizing that creative tension exists when a gap exists between where a person is (reality) and where the person want to be (the personal vision)." Sigler (1999)

It is a cultural change of our times to put so much focus on the individual self-realisation personal goals and values against those of the group or society. We will see later cultural dynamics that can affect the applicability of such a paradigm for individual development.

Mental Models

The reflective process described by Personal Mastery is then extended by Senge to our understanding of the processes around us, and the extent to which the individual is entrapped by or reflective off the key Paradigms or "Mental Models" that frame their understanding. Thus Senge sees "Mental Models" as:

> "…reflecting on and continually clarifying and improving our internal pictures of the world…" Senge (1994)

This suggests going beyond your inherited paradigmatic thinking to look at the broader processes affecting a system.

Shared Vision

Senge seeks to building a "learning organisation." To achieve this he believes that it is essential to build a "shared vision":

> "…building a sense of commitment in a group, but developing shared images of the future we seek to create." Senge (1994)

Team Learning

With team learning, Senge delves deeper in the behavioural aspects of effective teams within an organisation. He attributes to these teams the characteristics of:

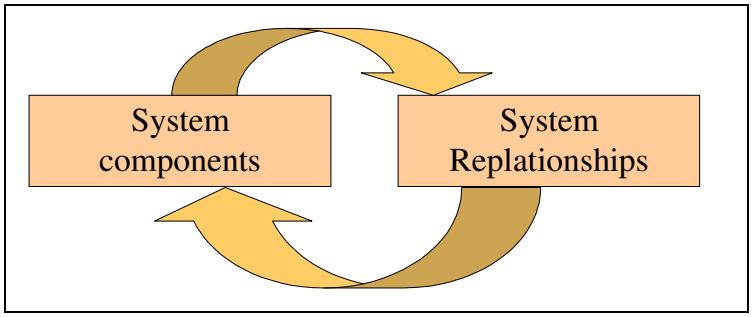
> "… transforming conversational and collective thinking skills, so that groups of people can readily develop intelligence and ability greater than the sum of the individuals members' talents." Senge 1994

Team learning is in this sense the opposite of personal empire building:

> "If you can't make people self-sufficient, your aid does more harm than good." Sigler (1999)

The role of System Thinking

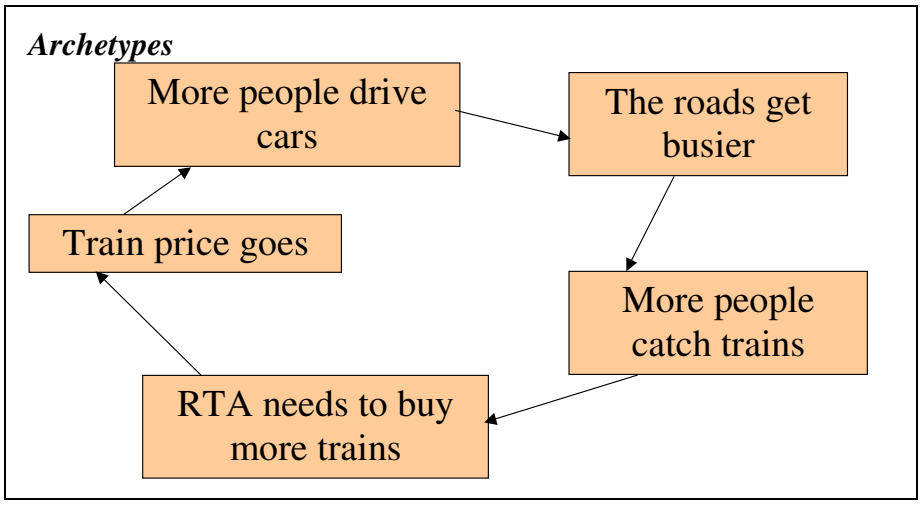
A system is an entity that functions through the interaction of its individual parts. The interaction is through a process of *feedback* (in the sense of information returned as a response to an originating source), and *reciprocal feedback* by which the different component parts of a system interact. The following diagram illustrates this relationship:



System thinking is that process of continually revisiting our understanding of the processes around us to see the models inherent in the systems that we deal with. The relationships in a system are of course far from trivial. Senge seeks to build, often through illustration, an understanding of the characteristic patterns that systems exhibits.

Systems can have interactions that are not always obvious. Problems can exist in a circular feedback loop of the type described below, where a change to one element of the loop (for example building more roads) has an impact on other systems. Senge defines archetypes that can be used to understand and the recurrent patterns of behaviour in organisations and systems. The engagement of the organisation in the process of reflecting on its systems and processes through recognition of archetypes is part of the process of change. This process is taken further in *Action Research,* a methodology that actively engages the researcher as an agent of change rather than an impartial observer.

Feedback cycles are central to the understand of system interactions, but are not always intuitive in the understanding of how systems interact. It is very easy to think in terms of simply linear sequences of events when looking at a system. Feedback patterns of a system look to address issues of causality through action, interaction and reaction rather than the modelling of linear causal relationships.



Archetypes

Archetypes are those patterns of causality that are comment to many contexts and circumstances. The “tragedy of the common” is one of them, and exhibits many of the issues of looking beyond the simple linear sequences of desire and action. An archetype for the "tragedy of the common" might be:

- Hating the noise and smell of the city, I buy my idyllic sea-side escape
- Others discover this beautify retreat, and move in
- The first apartment block goes up
- The shopping mall arrives
- The Freeway goes past the front door!

How are common resources managed and how are limits set? What effect to behavioural elements and cultural directions have (eg an aging society starts building more and more retirement homes in coastal communities)? What are the limits at which a resource will be stressed, and what incentives are there to share resources. Above all, it is important to note that such problems can only be resolved at the community level through communication and negotiation.

The effect of population is similar:

- I devise a shorter route to work
- As the population of the city grows and congestion increases, others arrive at just such solution
- Eventually all possible routes will take roughly the same time as the population seeks to optimise the route to work.

It is not always possible to predict the effect of population growth on a system model. Scalability of systems is one of the biggest issues in Internet content delivery. The size of the population of Internet users and the public nature of sites make it impossible to predict the limit of demand on a website. E-commerce and content providers continually fail to meet their promises of handling peak usage: because the peak using on the Internet is unknowable! Some even boast of access failures because they represent a level of interest which is perceived to be enviable – such boasts however may belie the level of frustration of the users who have experienced such demand-based failure to reach a site.

All systems have weaker and stronger points. A system can fail due to apparently trivial causes. These points of weakness are the subject of focus both to address deficiencies in a system, *and* as a focal point to exercise change.

A dilemma facing software developers is complex interaction of heterogeneous systems. Systems that otherwise work perfectly well independently may be a minor disaster when working together. In 2000 baggage handlers reached just such a systems integration crisis at Sydney International Airport, where the two independent systems controlling the

logistics and operational elements of their baggage handling system example each independently passed all tests, but when bought together failed catastrophically (see below).

Systems Theory and Chaos

The coherent nature of systems may not be immediately obvious. Chaos theory has lent insights into the way in which complex apparently discrete events can interrelate to form coherent and potentially model-able patterns of behaviour. Lorenz in his famous "butterfly" speech identifies them as processes which:

> "…appear to proceed according to chance even though their behaviour is in fact determined by precise laws." Lorenz (1993) p.4

The patterns of system behaviour may not be predictable in advance, and even apparently random patterns may in fact follow a broader system.

A case study: Systems Integration – The project announcement

http://www.bankstownairport.com.au/news/may-11(b).htm

Sydney Airport International Terminal Baggage Handling System

11 May 2000:
A $34 million baggage handling system is to be fully operational at Sydney Airport International Terminal by July 2000. This system introduces important new technology to Australia and will significantly expand the existing baggage capabilities. The new system will deliver the following benefits:

- Substantially improved baggage system
- Quicker in-bound passenger baggage recovery.
- Greater volume and operational efficiency.

The system is also part of the extensive upgrade that has been undertaken at Sydney Airport to prepare the airport for the Olympics. The new baggage handling system was started up in March and a pre-commissioning shakedown operation has been underway for a number of weeks. The operation of the new high capacity baggage system has been reviewed by an independent safety expert following management and staff concern safer in-built safety devices in the overhead conveyor system did not operate to the required standard. Contingency plans were also implemented to ensure that passengers were not inconvenienced. Following intensive consultations with the baggage handlers, airport staff and building contractors, new safety measures are being put in place. Sydney Airports has also convened an ongoing working party of all those involved in the construction and operation of the International Terminal baggage system to ensure that appropriate technical and staff resources are in place to maintain the required system reliability and operational throughput consistent with our Olympic objectives. Issued by Sydney Airports Corporation Limited Public Affairs #010003

Further information: Julian Green 02 9667 9978
Peter Gibbs 02 9667 6470

A case study: Systems Integration – Part 2

HTTP://WWW.GLOBE.COM/DAILYNEWS/229/WORLD/OLYMPIC_BRIEFS_:.SHTML

Olympic Briefs

By Associated Press, 8/16/2000 15:09

SYDNEY, Australia (AP) With less than a month to go until the Sydney Olympics, a trouble-prone new baggage handling system broke down briefly Wednesday at Sydney airport.

Sydney Airport Corporation said a failure occurred as a new bag screening system was integrated with the main baggage handling system. The breakdown delayed four departures.

"We are hopeful they can solve the problems before the Olympics," Transport Workers Union spokesman Richard Olsen said.

Olsen said the problem was separate to a series of glitches last week involving a breakdown with the main baggage system.

That was believed to have been related to a software problem and involved several bags being sent to the wrong baggage sorting bin.

The Sydney Airport Corporation admitted it had computer software integration problems when the baggage handling system was put into use in July.

The airport also has had two power outages in recent months leading to long delays for passengers.

MONACO (AP) The International Modern Pentathlon Union is surprised by criticism that it failed to agree to drug controls for its Olympic athletes, claiming the attack was based on "inaccurate information."

The federation (UIPM) was one of three mentioned by the World Anti-Doping Agency this week for failing to sign up for out-of-competition testing of competitors before the Sydney Games.

WADA criticized the three holdouts modern pentathlon, gymnastics and volleyball for placing an "unnecessary stigma of suspicion" on their competitors. UIPM said it was ready to sign and support drug testing.

"UIPM maintains one of the most thorough and open anti-doping policies in the Olympic movement, and has done so for many years," the organization said.

It said it had begun out-of-competition testing in January 1999, and its policy was recognized as an example for other sports federations to follow at a major conference on doping in sport last year.

"UIPM has openly supported WADA from its inception, but had two concerns regarding the confidentiality of the data linked to the protection of athletes, and the international federation's autonomy," it added.

The statement said meetings were held to iron out the differences, and it was agreed that the amended WADA contract would be sent to UIPM headquarters for signature. However, that contract never arrived.

"UIPM remains ready to sign the amended agreement upon receiving it and wishes to start the number of agreed controls as soon as possible," the statement said.

COPENHAGEN, Denmark (AP) Wilson Kipketer, the world record-holder at 800 meters from Denmark, has been cleared by doctors to compete in the Olympics.

Kipketer injured a leg muscle after winning the 800 June 23 at Helsinki, Finland. He has not competed since then, but expects to run again in the Golden League meet at Brussels, Belgium, Aug. 25 or in a meet the following day at Gateshead, England.

The Kenyan-born Kipketer also holds world indoor records in the 800 and the 1,000.

Be aware that every change to a system can have its own "revenge effect" brilliantly described by Tenner(1996) with numerous examples:

> "Technological optimism means in practice the ability to recognize bad surprises early enough to do something about them…But vigilance does not end there. It is everywhere. It is in the random alertness tests that have replaced the "dead man's pedal" for train operators. It is in the rituals of computer backup, the legally mandated testing of everything from elevators to home smoke alarms…Revenge effects mean in the end that we will move ahead but must always look back just because reality is indeed gaining on us." Tenner(1996). P.277

This is the heyday of the economist, but not all things can be effectively measured. Tenner gives the example of the unquantifiable problems (whiplash, back pain, and psychological symptoms) that arise for the car accident survivor. Such effects are not easily quantified into a formula for recompense or a clean set of rules for resolution of problems:

> "Like other forms of pain and suffering it presents the problem, uncomfortable to quantitative investigators, of being both real and unmeasurable. And it poses the dilemma, disturbing to legal analysts, of ignoring suffering or rewarding fraud. Tenner(1996) p53.

The limits of classification

There are times when the process of analytical reduction of a systems components can itself be problematic, particularly when combined with social and cultural assumptions. The classifications that are encoded into systems are in many cases neither value-neutral nor flexible to ambiguous responses. One of the most ambiguous and value-laden classifications schemes over time is that of "Race" and "Ethnicity". The traditional data entry form for a computer system is often intolerant of the ambiguities of classification in this area. Les Earnest describes his encounter with a 1960's Security Clearance form the *Defence Communications Agency* in the US:

> "…a glance in the mirror indicated that there was Middle Eastern blood in my veins. I have a Semitic nose and skin that tans so easily that I am often darker than many people who pass for black….In answer to the race question on the security form I decided to put "mongrel"…. I asked them what had prompted this investigation. After some glances back and forth, one of them admitted that they were putting their clearance database on IBM cards and found there was no punch for "mongrel."…I was surprised to learn that nearly everyone believed in the concept of racial classification. It appeared that even people who were victims of discrimination acknowledged it as pert of their identity. Clearly the security people believed that I had caused this problem, but I felt it was the result of a stupid question… they apparently found it impractical to obtain the hour or two of a programmer's time that would have been needed to fix the computer program, so they chose instead to work with their standard tools. This led to an expenditure of hundreds of man-hours of effort in gathering information to try to intimidate me into changing my answer." Earnest (1989), p 177)

The Cultural Context of Systems

In 1980 Hofstede undertook a fascinating survey of behaviour and perceptions in IBM offices around the world. Arising from his analysis of this survey were the "Dimensions of Culture", which he formulated to describe consistent patterns of thinking and behaviour. The IBM study went across 40 countries and identified five dimensions behavioural dimensions of cultural difference. Hofstede describes them as "polarities" not "characterisations", that is, individuals might be mapped at different points on each of the five scales that he describes. The focus of his work was in organisational dynamics, but his insight is valuable in the understanding of the cultural aspects of any systems building activity. To be aware of the cultural context that ideas, information and systems are to be delivered is of considerable importance if the desired outcomes are to be achieved.

Power Distance

The first dimension that Hofstede identifies is Power Distance:

> "the extent to which the less powerful members of institutions and organizations within a country expect and accept that power is distributed unequally."

A society with a focus on "large power distance" might tend toward centralised decision processes, and with hierarchical structures that actually reflect the structures of power and decision-making. A society with a small power distance might reflect organizations in which peer-based decisions are sought and the formal structures only loosely reflect the processes for reaching decisions. These organizational and cultural factors can have a dramatic impact on the effectiveness of systems if they do not accommodate such variances across different organizations, cultures or countries.

Individual/Collective

The individual *vs.* collective cultural orientation of a society or familial network has consequences in the behaviour within a systems context. Those with an orientation toward a culturally collective interaction would have a focus on strong kinship networks, an emphasis on harmony and mediation, and "high text" communication (that is, focussing on ensuring that all issues are elaborately described and explored). This is against an individualist orientation that might have a focus on identity as the individual, a task achievement orientation rather than a "process" orientation:

> "Individualism pertains to societies in which the ties between individuals are loose. Collectivism as its opposite pertains to societies in which people…are integrated into strong cohesive in-groups…"

Cultural "Masculinity" vs. "Femininity"

These are Gender roles & perceptions as they affect cultural values, not simply in the sense of the individual but as a general organisational trait. The focus culturally would be on the degree to which societies might focus on "nurturing" traits (his "femininity" dimension). In his terms a culturally "masculine" focus would be on corrective approaches to discipline within in-groups, and a societal focus on performance.

Uncertainty Avoidance

"Uncertainty avoidance" is the degree to which a level of ambiguity creates anxiety / stress is tolerable. A cultural value of high uncertainty avoidance might be more focused, but would place higher value on maintaining circumstances where ambiguities and confusions are minimized. A society with a highly individualist focus on achievement might tolerate a higher degree of uncertainty as part of the essential functioning of a change-oriented society.

Social Orientation

By Long-term orientation, Hofstede refers to the capability of the society to change over the long term but remain coherent. He examines here the different cultural traits that are reflected in long-term and short-term thinking and the value that is placed on long-term consequences of actions or short-term achievements in a familial and social background.

These five dimensions are interesting as a model for understanding the social context into which systems may be introduced: the introduction of representation or analytical systems in a high-context, highly-collective society might mandate a very different focus in the emphasis of design from those to be delivered into a low-context, highly performance oriented society. Indeed, the very things that annoy users in one cultural framework might be pleasing to other cultures.

Obsolescence

How long will our systems last, and do we, or should we care about this? The following news piece reflects an interesting insight into the difficulties of technological obsolescence. It is an interesting debate in the artistic community whether it is the responsibility of the artist or the archivist/preserver to consider issues of transience and obsolescence. It is an issue that will confront systems increasingly over time as major resources become inaccessible simply through the pace of technological development. The rapidity of technological development brings considerable long-term difficulties in the management of creative products. Libraries and Museums have a key role in the preservation of analytical and creative endeavours over the long term. However, most libraries are ill equipped to undertake research into the preservation of new media artefacts and creations. Where the preservation of printed

works is well understood, issues of obsolescence of new media technologies affect all aspects of the new media artefacts.

As each new technological innovation introduces new methods of creative content delivery, our long-term horizons of archive planning appear to reduce. The widespread adoption of Information Technology as an integral part of the research process, and the speciation of software vehicles for content creation, mean that on the basis both of cost and volume of content creation the meagre budgets of most libraries simple are not sufficient to sustain the role of comprehensive collection builders. Longstaff (Longstaff, Chittister et al. 2001) recently argued that the risks associated with moving critical business and government infrastructure to an Information Technology framework are poorly addressed. He identified several adverse impacts at the national level of increased reliance on Information Technology:

- "Increased complexity of our information systems because of the added interconnectedness and interdependencies between and among infrastructures
- Reduced operational buffer zone in most infrastructures, and the ever-increasing adherence o the just-in-time philosophy as a vehicle for cost reduction and efficient operation.
- Enhanced accessibility of would-be terrorists to our defence, banking and financial institutions, and to other critical infrastructures." (p.43)

Similarly, the move to subscription-based digital library collection building has associated with it inherent risks of technological obsolescence. Most libraries are in no position financially to undertake fundamental research in areas of technological migration from different versions encoding, media storage or content delivery platform (Ekman 2000).

Digital libraries now discuss the provision of "persistent" access to new media artefacts, rather than "archiving". The management issues go beyond simple system continuity management, including the management in the long term of all aspects of:

a) Hardware architectural requirements to operate the software. In the context of dynamic websites, this might include the server hardware, the client hardware, and intermediary network architectures.
b) The physical and logical storage media requirements.
c) The operating system and network communication protocols used to delivery /retrieve the information
d) The layered software products to interpret/present the content.

Failure to consider issues of obsolescence can prejudice our ability in the long term to preserve creations that are of cultural significance for future generations. Valuable information *can* be lost, and the recovery of information from damaged media or obsolescent software or hardware architectures can be a very expensive proposition.

A case study in Obsolescence

http://news.bbc.co.uk/hi/english/sci/tech/newsid_654000/654116.stm

BBC NEWS

Wednesday, 23 February, 2000, 17:31 GMT

Old computers lose history record



Computer files: Which is the dinosaur?

Vital archaeological records could be lost as the computers on which they are stored become obsolete.

The physical site is nearly always completely destroyed during a dig, but archaeologists claim the knowledge they glean from the ground is then available for posterity.

Studies in York have revealed that in fact data stored on computers could disappear in little more than a decade.

"The irony is that archaeological information held in magnetic format is decaying faster than it ever did in the ground," warns William Kilbride of the Archaeology Data Service (ADS) at the University of York.

Important evidence

The ADS was asked to examine computer records of 180 Bronze Age excavations in North East London conducted between 1991 and 1996 by the Newham Museum Archaeological Service, which has now closed.

The Newham excavations yielded important information about London and the Thames during the Bronze Age, but that data was never published.

The records comprised more than 6,000 database, geophysical and CAD (Computer Aided Design) files held on 220 floppy disks.

When they came to examine them, the archivists found that 5% of the older disks had become corrupted. The magnetic coating on the disks had simply succumbed to the slow erosion of time.



London's history at risk

State of the ark

Another problem they encountered was obsolete formats. In computer terms,1991 is ancient history. Some of the word processor and database programmes used then are no longer available.

"The formats of computer files change rapidly. A file created in state-of-the-art software one year becomes obsolete the next, as the software is updated. Old disks are useless when the hardware is no longer available to read them," says archaeologist Keith Westcott.

> **Old disks are useless when the hardware is no longer available to read them**
>
> **Keith Westcott, Archaeologist**

Finding a computer that will physically accept old-fashioned 5¼-inch or Amstrad-style three-inch floppy disks is not easy.

The Newham records have now been saved to a modern server which is linked to the internet. If the records had been left on floppy disk for much longer, they would have been lost forever.

The only secure answer to data conservation would seem to be the internet. Servers can go down or will need upgrading, but in theory, information on the internet will last forever.

Kept on standalone computers or on disks in a shoe box, data from sites will be of less use to tomorrow's archaeologists than if the site had not been excavated in the first place.

The laptop on which much of this book was written is an example of the pervasiveness of obsolescence in technological design. The devise has 17 connectors, each different, for interfacing to other devices. Of these, 10 of the interfaces represent different generational approaches to the same problem: interfacing to other digital devices. The accretion of new connectors has lead to ingenious uses for side edges and cavities, and the bottom of the unit, all of which are strewn with connectors. At least four of the devises are entirely proprietary to this model – which would only be functional with suitable drivers explicitly for this machine. The keyboard contains keys that would only have meaning on this range of computers, and must therefore have drivers to suit. How long will drivers still be available for this specific model? The batteries that drive the machine are moulded specifically for this range. They are easy to get at present, and will be for several more years, being a recent purchase. Will this last 10 years? What about two generations on in Operating System? Will the supplier, however large, still be releasing drivers for this particular model If not the short life-expectancy of laptop hardware, the complex software obsolescence issues will certainly relegate this particular machine to history in (if lucky) in little more than 5 years.

Standards Organisations

Standards organisations have played a key role in mitigating the issues of obsolescence in Information Technology. Standards Organisations have played a role in not only the establishment of agreed common approaches to various aspects of Information Systems, they have in some cases also served an exploratory role in the formulation of standards in advance of any existing generally used approach. Standards bodies can be Standards organisations can make a useful reference point in the evaluation of technological options. The existence of an accepted standard is a starting point in the evaluation of Information Systems approaches.

The credibility of the standards organisation is crucial. It is not unknown for industry consortiums of computer software or hardware suppliers to be partisan rather than universal in their approach to standards definition. However, within Information Technology the actual Industry adoption of a standard is equally crucial to its utility. Many standards have been formulated which have had little value due to a lack of industry acceptance.

Some key standards bodies in Information Technology that are relevant to the Arts and Humanities are:

W3C – A key standards organisation in the formulation of standards for the World Wide Web is the W3C – the World Wide Web consortium (www.w3c.org).

ANSI - The American National Standards Institute (www.ansi.org). ANSI has been a key leader in the establishment of standards that are later adopted internationally.

ISO – The International Standards Organisation (www.iso.org). The adoption of a standard by the ISO is one of the last stages of attaining final credibility for a standard outside a specific national context.

For example, the establishment of the *Text Encoding Initiative* standard for SGML markup of literary texts in 1994 has been a key step toward encouraging software-independent markup of texts in a way that will survive particular word processing and document viewing systems. The capture of texts following such a standard for markup achieves two goals:

- the identification of meta information relating to the text which will be independently useful for textual analysis and criticism.
- the establishment of software and implementation independent approaches to sharing the texts.

Working Collaboratively

The new mechanisms for collaboration that email, videoconference and hypermedia allow. The *Stamford Encycopedia of Philosophy* illustrates how effectively workflow systems can be deployed to facilitate the rapid development of online resources in a fully peer-reviewed environment where the online collaborative tools facilitate the version control and publishing process. The system does not constrain the author to particular environments for authoring, but does provide templates for presentational form of the product and an automated and highly structured peer-review workflow environment. Email is used to alert peer-review participants on the arrival of new articles and to manage their prioritised work schedule.

However, it must also be remembered the effectiveness that can be achieved through focussed sessions of collaborative work can achieve:

> "It took Blake Camp to begin turning what was starting to feel like defeat -- the three of us were talking among ourselves about how to lower our sights and cut our losses -- into what felt like a chance at victory. But as we talked through the problems one by one with the Archive right in front of us all, with everyone who was relevant to a solution there to contribute, we discovered that many of the smallest problems could be dealt with on the spot, with John pounding the keyboard to change the SGML or the search logic or the format of a button as we exposed the source of one misunderstanding after another. Several larger problems would

> take longer. But along the way we regained our optimism and our confidence, I think largely because we realized that we had managed to come back to a common understanding of what we were after and what could be achieved." (Eaves 1997)

Information Systems Practice

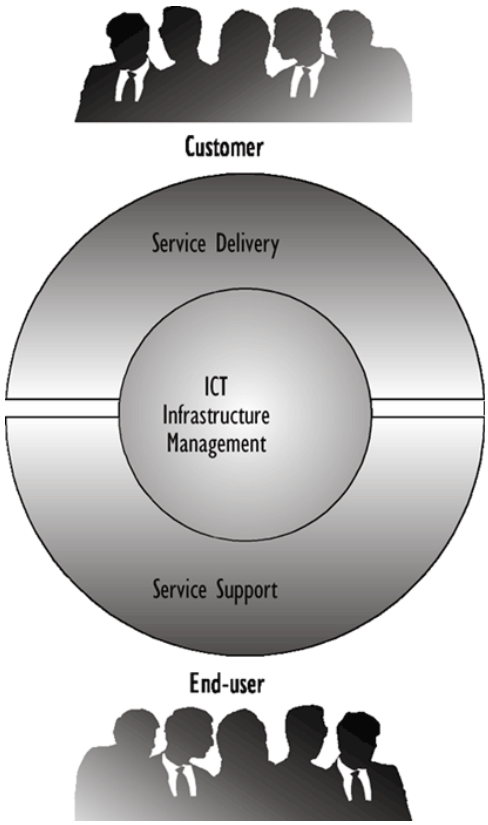
Project failures with Information Technology (IT) projects are pervasive. A survey conducted in 1988 by Peat Marwick Mitchell & Co of the 600 largest clients indicated that:

> "...some 35% currently had major runaways...the Big Eight accounting firms, computer suppliers, and even in-house data processing staffs are fast building a record of mediocrity."

By some estimates up to 56% of IT projects fail (Standish Group, 1999). While there is no one overriding factor that causes a project to fail, it is widely acknowledged that projects are more likely to be more successful with a team that is steeped in IT project management (Dantas et al, 2004). Information Systems practice has been plagued with a reputation for unpredictability in project outcomes, cost over-runs and outright failures. Nevertheless, Information Technology-based systems have been extant for sufficient time for a body of practice to emerge that can limit the risk associated with Information Technology Systems management. The ITIL (Information Technology Infrastructure Library) exemplifies one of the most useful codifications of Information Systems Management Practice. Its objective is to eliminate some of the risk associated with IT projects by establishing as set of operational, change and risk management standards for initiating and managing IT systems.

ITIL

The UK's Office of Government Commerce developed ITIL as a set of guidelines to codify good practice across the spectrum of services that delivery information technology systems in an organisation. In that sens the methodology is based on the concepts of Service and Customer care. ITIL defines an IT service as “a set of related functions provided from the IT infrastructure in support of one or more business areas perceived by the customers as a coherent and self-contained." (Stern, 2001) The centrality of the customer, and the unified view of IT outcomes as a service are crucial to the ITIL model:



(Office of Government Commerce., 2002)

The objective of ITIL is therefore to instantiate the management of archetypal Information Systems problems in a body of management practice that removes some of the risks associated with Information Technology projects and adds more predictability in the outcomes through the pursuit of well-tried practice.

While ITIL is gaining international recognition as an effective standard for IT systems management, the implementation of ITIL is not without cost and difficulty, especially in organisations with a rich, pre-existing Information Technology management culture. The beginning point, therefore, in implementing ITIL is undertaking an audit and catalogue of existing services. The Service Catalogue provides the essential prerequisites to:

- Define and better utilise and publicise IT services to internal customers
- Establish Service Level Agreements (SLAs) that are aligned to business objectives
- Implement standardised and repeatable service fulfilment processes
- Determine the associated costs and therefore predictable service budgets
- Measure service quality and track SLA compliance

The key elements of ITIL can be described in two broad categories:

Service Delivery

- Service Level Management
- Financial Management
- Capacity Management
- Availability Management
- Business continuity

Service Support

- Service Desk
- Incident Management
- Problem Management
- Change Management
- Configuration Management

IT GOVERNANCE

One of the paths to realising effective ITIL implementation in an organisation is a "capability/maturity" model that views the process as a set of stages that gradually improve the IT governance by progressive enculturation of the service delivery culture in the organisation. The Information Technology Investment Model (ITIM) is such a "maturity model that looks at a staged implementation of ITIL through

1: creating investment awareness in the organisation regarding the benefits of ITIL
2: building investment foundation, especially through the development of an IT asset management process
3: developing a complete investment portfolio reflecting "risk-adjusted costs"
4: improving the investment process and ensuring organisational engagement in this process
5: leveraging IT strategically to gain coherent benefit from integrated IT service delivery.

Summary

The understanding of Systems and Systems Thinking are the prerequisites for the effective construction of complex systems in the Arts and Humanities. This chapter explores a Systems approach to thinking about all Information Technology processes, and examines the practice of Information Technology implementation from a Service approach, with particular reference to the Information Technology Infrastructure Library. The ideas presented in this Chapter will be revisited as different aspects of Information Systems are explored.

Questions

1. What is the role of systems thinking in the Arts and Social Sciences?

2. Describe, briefly, Bertalanffy's General Systems theory.

3. Describe Hofstede's 5 dimensions of culture.

4. Describe three types of obsolescence that are problems in the medium and long term management of multimedia content.

5. What is the purpose of the Information Technology Infrastructure Library

Further Reading

DANTAS, A., BARROS, M., & WERNER, C. (2004). A simulation-based game for project management experiential learning. 16th international conference on software engineering & knowledge engineering, Banff Alberta, Canada, Retrieved 10/01/05,

DRUCKER, Peter F. (1964) Concept of the corporation . New York, N.Y. : New American Library, 1964.

EARNEST, Les. (1989) "Can computers cope with Human Races?". Communications of the ACM, Feb 1989, Vol 32, No 2, pp.174-182.

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.

HOFSTEDE, Geert H. (1991) Cultures and organizations : software of the mind / Geert Hofstede. Publisher London ; New York : McGraw-Hill, c1991.

LILIENFIELD, N (1978) the rise of systems theory: an ideological analysis. NY: Wiley

LONGSTAFF, T. A., C. Chittister, et al. (2001). "Are We Forgetting the Risks of Information Technology." Computer 33(12): 43-51.

LORENZ, E (1993) "The essence of Chaos" Seattle, University of Washington Press, 1993.

NEUBECKER, Robert. "It's late, costly, incompetent - but try firing a computer system", in Business Week, Nov. 7, 1988, pp.63-4.

OFFICE OF GOVERNMENT COMMERCE. . ICT infrastructure management: Best practice for ICT infrastructure management. London, 2002

SENGE, Peter M (1992).. The fifth discipline : the art and practice of the learning organization. Milsons Point, N.S.W. : Random House Australia, 1992

THE STANDISH GROUP. (1999). CHAOS: A recipe for success (Publication No. 1). West Yarmouth, MA: The Standish Group.

STERN, Andrea (2001). Reinvesting the IT Dollar: From IT Firefighting to Quality Strategic Services. Educase Quarterly (3) 2001, 8-14.

TENNER, Edward (1996). Why things bite back. London: Fourth Estate: 1996.

VON BERTALANFFY, L (1968), General Systems Theory: Foundations, development, applications. New Your, Brazille

Chapter 2 – Content Representation Systems

Digital Convergence

The powerful synergies of affordable desktop computing and readily accessible networked communication have bought a revolution in the range of representational systems available for the content creator. Authoring systems for text, image, video and animation design abound. Software is readily available from freeware, to low-cost shareware and traditionally licensed software.

The evolution of capabilities for representation exhibits a classic exponential rate of progress. We see, from 50th Century BC to the 15th Century BC the use of ocre for wall paintings, the etching of wall art and inscriptions that show how fundamental the desire to communicate is within the human endeavour. At about 3500 BC we see the stone carving of songs in Mesopotamia, for immediate communication and obviously for preservation of a cultural heritage. Stone, papyrus, bark, hide and later paper became essential media for the commerce, governmental and creative communication. The value placed on written and artistic communication cannot be doubted when we view the products of the medieval monks with their long and often exquisitely beautiful labours in copying books.

The invention of the printing press in Gutenburg in 1450 saw accelerated the rate at which works could be created and distributed, and the innovation of photographic techniques heralded by the Daguerrotype in 1837 bough progressively more media for the communication of ideas and creative endeavours. Marconi in 1901 bought sound recording, and less than 14 years later we see the first modern film in “Birth of A Nation” (Griffith). In 1926 the first techniques for broadcast television were demonstrated by Baird. In 1940 Walt Disney’s *Fantasia* introduced animation to popular culture.

Through to the middle of the 20th century we see the evolution of technologies for printed and visual communication, and the emergence of mass broadcasting media. The last decade of the 20th century saw a rapid succession of innovations that introduced the digital era:

1958 - Pong - the first Video Game
1965 - Ivan Sutherland creates VR systems
1969 - ARPANET - TCP/IP is born
1972 - Willie Crowther releases *Adventure*
1974 - First PC (the Alair)
1983 - First CD’s
1985 - Bulletin Board Systems take off - the first network sharing of information
1988 - Macromind (now Macromedia) released - first consumer multimedia authoring tools

The 1990's were a decade of digital innovation on all fronts, with the progressive integration of digital technologies in all forms of media, with the evolution of *multimedia* or *new media*. 1992 saw the emergence of the World Wide Web as a common backbone for multimedia communication. 1992 also saw the first 3D game to reach widespread popular use: *Wolfenstein 3-D*. Games technology was an early driver in the popular uptake of Personal Computer technology, with *Doom II* selling 500,000 copies on the Christmas after its release in 1994. The emergence of Audio streaming in 1995 and Video streaming in 1998 rode a progressive swell of interest in networked computing activities and communication.

We arrive at the 21st Century with a multitude of pervasive content formats, including:

- Book
- Mass Media - Radio
- Mass Media - Television
- Interactive Media
- Internet & Multimedia
- PC & networking convergence
- Virtual Reality & Multimedia

All these exhibit aspects of *Digital Convergence*. Books are transformed by multimedia to include audio and visual experiences. Hypermedia transform the Book from a serial experience to an infinitely complex set of relationships. Databases transform the book for a static resource to an Information System. Similar change affects audio and visual communication of all sorts.

Recent decades have seen the expansion of Digital technologies out of Western, often English-language, cultural context.

MULTIMEDIA

Multimedia applications are those that incorporate any combination of text, audio animation and video content in a digital form. When we refer to "digital" form, this refers to the encoding according to some standard of the information in a binary format which is interpreted by some form of runtime software. The encoding may contain information pertaining to the content itself, the runtime software needed to interpret the content or indeed the structure of the content. It is immediately evident that several additional dimensions are implicit in the multimedia artefact:

- The presence of runtime information systems to interpret the content
- The presence of digital presentation equipment to render the content in analogue or physical form (as displayed text, sound, animation, video, etc).

Central to multimedia content is the concept of "hypermedia". Content is necessarily bound by the highly structured experience of reading a book or as a recipient of traditional broadcast media. Multimedia delivers interactivity where the user of the content can be presented with choices regarding the sequence of events. The experience of the system may be non-linear, and there is generally an expectation that

the integration of multimedia elements is seamless: explicit invocation of multimedia elements to bring them into action may detract from the delivery of the entire content as conceived by the authors of the content.

The World Wide Web and the CDROM have, in somewhat different ways, been significant technological platforms for the development of multimedia. Games and gaming systems, distributed widely through CDROM, brought considerable innovations in animation techniques, the integration of audio, video and animation. Games also inspired the development of readily available desktop authoring systems for animation. Similarly, the World Wide Web was a driving factor in the innovation and widespread uptake of technologies for digital scanning of images. More recently audio and video encoding systems have become immensely popular. Video games on proprietary home systems have also had phenomenal success.

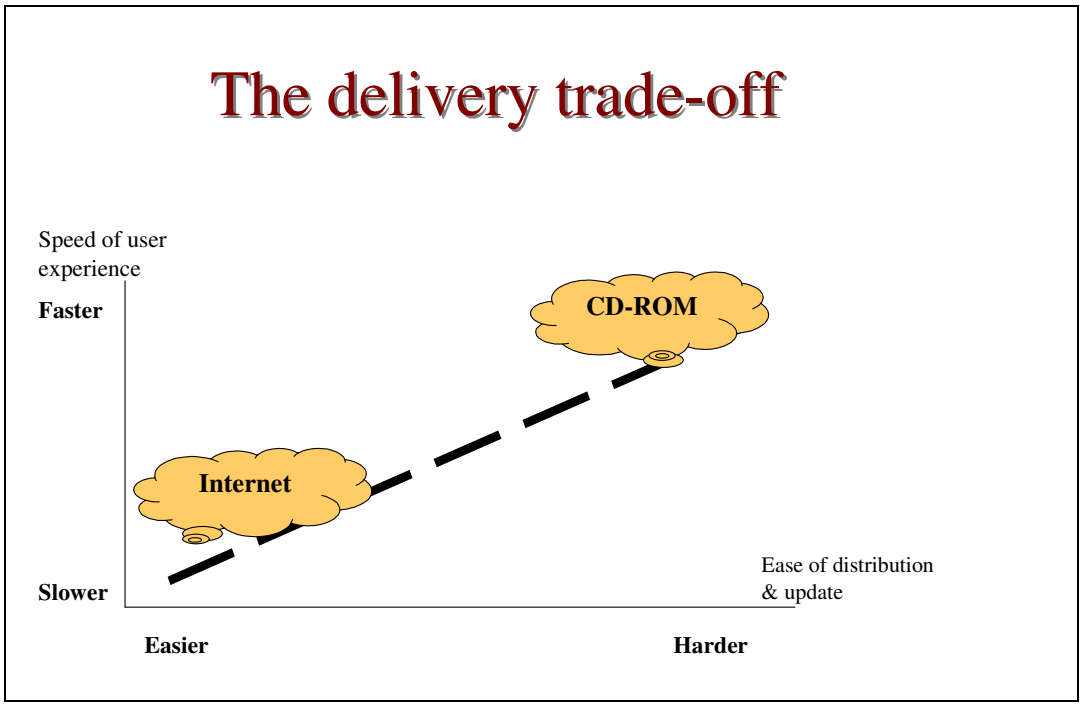
Three key multimedia systems have common usage:

- Reference tools, often CDROM based, such as Britannica or Encarta encyclopaedias'.
- Information delivery and exchange systems on the World Wide Web – for example Virtual Museums and Online Education sites.
- Computer Games.

However, the convergence of technologies for multimedia visualisation are also realising systems such as:

- Online avatar-type virtual reality systems
- Integration of multimedia into email information exchange
- Embedding of multimedia communication systems into portable communications equipment (*eg* WAP)

Nevertheless, the content creator is faced with considerable architectural issues in the development of content. Price is still a significant factor for advanced authoring tools and for content delivery systems. Technological obsolescence greatly limits the life of multimedia creations. Finally, choosing the platform is still a very significant element of the cost in actually delivering a multimedia creation. Hidden costs include the long term maintenance of the creation and continual translation into new delivery architectures.



There are significant cost trade-offs between the choices of different delivery media. For instance, CDROM or DVD delivery allows the integration of considerably more content with faster delivery, but has high costs of production and distribution, and significant step-wise costs for the release of periodic updates. With CDROM and DVD delivery, the user is constrained to predefined activities that have been "burned" on the CD. Internet delivery has advantages in the potential immediacy and currency of the information delivered, and is unconstrained by a single application. On the other hand, while delivery over the World Wide Web can have lower cost barriers in distribution, and greater immediacy of updates, bandwidth issues and potentially also licensing costs can be problematical

TEXT MEDIA

By far the most common media element is text. Systems for textual representation in digital form include:

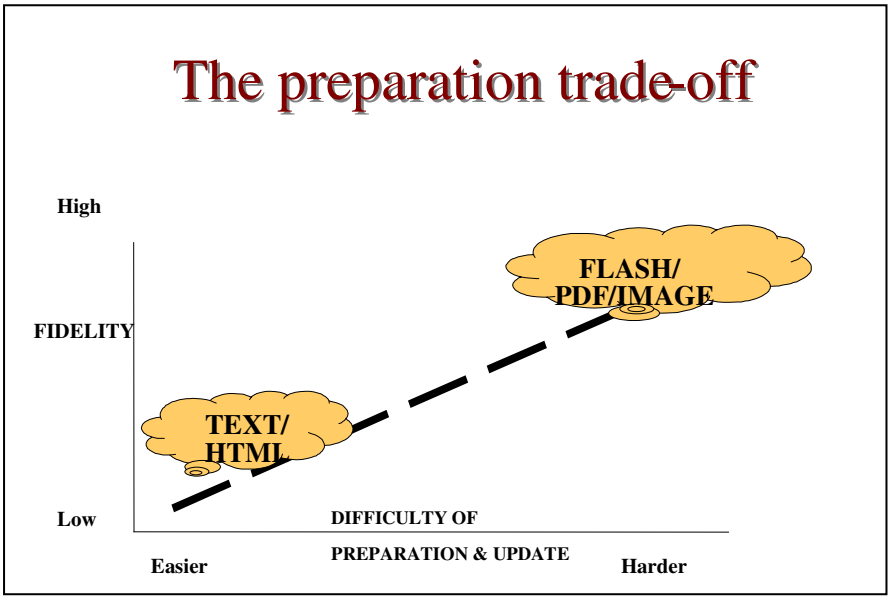
- Word Processing and DTP systems
- HTML marked up text
- XML marked up text
- Images with rasterised or bitmap text forms
- Macromedia Flash with animated text forms

The delivery of text in digital form itself has significant architectural decisions. For instance, using HTML for text preparation the author will discover that the colours, font and presentation of content may vary wildly from computer to computer depend on:

The hardware architecture (*eg* MAC , PC or Unix workstation), where colour saturation can vary dramatically.

The version of the browser for delivery of the text. Style sheets may be ignored altogether, fonts may be over-ridden or may not be available on the target runtime equipment.

The rendering of text in image form is one means of ensuring a higher level of *fidelity* in the presentation of the text, with somewhat less dependence on the architecture used. Issues of colour rendition may still be significant, but a greater certainty that the size, position and presentation of the text may be achieved. The rendering of text in image form comes, however, at a high cost when issues of subsequent amendments and changes to the text are required. This architectural trade-off is a very important consideration for the author when deciding how to deliver content.



Furthermore, the method of encoding text is a significant issue. HTML is widely used, and as widely understood. Despite its popularity, it has considerably flaws. These, and the use of HTML, SGML and XML are discussed in Chapter 3. SGML (Standard Generalised Markup Language) is a comprehensive markup language that is supremely flexible – as a generalised markup syntax it has great strengths in its ability to describe and markup almost anything. However, XML has become popular, as a subset of SGML, precisely because it defines a concrete, more manageable subset of SGML. XML had its origins as a transient data interchange mechanism, but has had considerable uptake in the humanities for all forms of text and multimedia encoding. While the markup process can be time consuming and expensive, the results in the humanities can be significant: a fully marked up document can yield diverse new opportunities for content analysis and textual criticism. The flexibility that SGML provides in consistent content recognition has seen its realisation in many disciplines: the Text Encoding Initiative (TEI) for literary encoding, MusicML and other standards for music encoding, and many other new directions in content markup.

On the Internet, JavaScript, proprietary browser Plug-ins, and JavaScript Applets are used to achieve specific enhanced text effects, rollovers (in which the text changes based on the movement of the mouse in proximity to the text), 3D effects for text and animation of text. These effects carry aesthetic value, but are used at the risk of a range of compatibility issues which may frustrate or negate the effect itself. Examples of such compatibility issues include:

- Older versions of browsers, or browsers that do not support or do not have a plug-in installed
- Browsers that have an earlier version of the plug-in
- Users who for security reasons disable specific Browser functions (for example Applets).

These techniques can have considerable aesthetic merit. The author should at least consider the degree to which their use will affect their target audience, and whether it is an issue. For example, a website target specifically to a youth audience might make use of embedded Flash for visual effect and dynamic features. It might be a reasonable expectation that the audience in this case that the author is attempting to reach can be expected to have the relevant configuration that will enable access to the site content. Alternatively, a site which is attempting to communicate information to a broader public base (for instance, an informational website) would need to consider carefully moving beyond a core set of features that can reliably operate on most browsers.

Language Issues

The ethnocentric nature of web publishing has been much discussed. While there has been some progress toward effective multilingual handling of text within word processing systems and on websites, there remain significant complexities in seamless language representation of text. Firstly, those dealing with historical languages may find that language capabilities of Word Processing systems, or language character sets for Internet Browsers simply do not have encoding capability for ancient characters. Secondly, Computer Systems often have a limited range of language sets defined: for instance, Microsoft Windows allows 512 language codes – yet the site (http://www.sil.org/ethnologue/) catalogues catalogue of 6,700 languages across 228 countries. A further problem is *overloading*: multiple different language attributes (sort order, printing code pages, fonts, time zones, etc) may be attributed to a single language code which might reflect several linguistic groups – *for example* Italian can be divided into two discrete groups: Italian – Italy and Italian – Switzerland.

In the website context, the international standard ISO 639-2 provides codes for languages & groups of languages. Inconsistent differentiation between groupings of languages and discrete languages can be a problem here also:

> "…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." (SIL p.9)

It is nevertheless widely used and is the basis for HTML/XML language definitions

In different text authoring environments, systems for language representation

- Sometimes involve software-specific plug-ins
- Can have versioning and obsolescence issue.

Because plug-ins can vary from one media to another, publishers of Multimedia E-books can face significant costs of converting their books from one technical platform to another.

UniCode

There is a general industry move to implement *Unicode.* Considerable international effort is underway to define standardised encoding character sets in each international language. This will provide a framework for consistent encoding and interchange of text content.

HTML

HTML is a widely used and simple method for encoding text. A loose derivative of SGML, and discussed in more detail in the next chapter, the popularity of the World Wide Web has given it immense support in the publishing of content.

Several languages have a large range of characters to be represented. The traditional ANSI character set allows only several hundred characters to be represented. As a result double-byte encoding techniques have evolved to allow representation of a larger range of characters. For instance, Websites in Japanese can be designed in one of several double-byte encoding standards, including Shift-JIS and EUC. The method of encoding is signalled to the browser using a **content header**.

For example:

```
text/html;CHARSET=x-euc-jp
```

signifies that the page being delivered by the website is:

a) marked up in HTML
b) language-encoded in Japanese EUC.

The specific extension of the website file (eg .htm) is not itself meaningful – web browsers and other runtime engines are explicitly expected to interpret the format of the web-served document from the content header (although Internet Explorer does do some interpretation based on a file suffix if lacking a content header).

Other common content headers are:

applicaton/pdf - a pdf document
application/msword – a Microsoft Word document
application/vnd-ms-excel – a Microsoft Excel document

application/exe – an application executable
image/jpeg – a jpeg image
image/gif – a gif image

This informs the browser of the character set to deploy for a given page. In some instances, this might involve the installation of specific plug-ins to recognise the character set.

CSS - CASCADING STYLE SHEETS

The use of style sheets can also considerably enhance the presentation of text on a website. The Style Sheet also 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). CSS define rule-based presentational instructions for HTML content markup. The HTML code using CSS is theoretically more portable, and site design changes easier to deploy. Once again, however, browser obsoleteness issues can present an issue, as older browser either do not support or imperfectly implement different Style Sheet features.

SGML AND XML

SGML, the Standard Generalised Markup Language, was developed in response to the difficulties of regenerating text content in multiple different formats and styles. WYSIWYG (what you see is what you get) is very intuitive for the content author, but suffers from limitations if the stylistic conventions are embedded **with** the content itself – especially when it come to delivering the content in different formats or media types. SGML sought to achieve a very important distinction between content structure and its specific presentation (ie the font styles, character sets to use, presentation of the structure, etc). Unfortunately, HTML has evolved a number of specific embedded presentation types that contradict the principle objective of SGML – for example the <font> tag allows definition of specific fonts, font sizes. This limited HTML to use in a WWW framework only. The objective of SGML was to provide a *standard* for encoding any form of content – where the rules for that encoding are defined separately from the content itself, and the markup syntax is strictly enforced. This structure is discussed in more detail in the next chapter. Because SGML can allow the content creator to develop their own markup scheme, it is very generalised but therefore also complex to implement in the first instance. XML (Extensible Markup Language) was developed after HTML to provide an information exchange vehicle for text which had specific markup conventions which were well understood (and similar to HTML) but was not tied specifically to a WWW presentational environment. XML is a much more consistent child of SGML, and has become a popular standard not only for information interchange but for long term archival representation of text and other content (eg MusicML).

XSL – Extensible Style Sheets

XSL (Extensible Style Sheets) are the equivalent of CSS for XML. They define the specific presentational format for content which has been marked up using XML. SGML and XML are discussed in more detail in the next Chapter.

PDF – Portable Data Format

The PDF (Portable Document Format) encoding method is a final presentational format that has the virtue of strong enforcement of the final look and feel. Because it has extensive internal encoding of information on all the objects within the document, controls can be enforced on how content is viewed and printed, with much greater control than can be achieved in HTML. PDF is therefore widely used for the distribution of published articles, prospectuses and other content where there is a desire to limit the extent to which the end-user can modify the target content or display it in a way different from that conceived by the content creator. As a proprietary representational format, it requires specific viewers and browser plug-ins for use, and specific software creation tools for content development. The ease with which HTML can be created shows the distinct trade-off between tight control of the final presentation and speed of content creation.

Text Interactivity

Text can be visually enlivened in a hypermedia environment for both aesthetic and functional reasons. Approaches for text enhancement can include:

- Use Images for text allowing rollovers and enhanced aesthetic presentation
- Use of text animation techniques using Javascript, Java Applets, Macromedia Flash, ActiveX and specific word processing plugins
- Use of text in Structured Indexes & Navigational aids
- Hyper linking the text
- Enhancing text discovery through Search engines.

Colour must not be neglected in the presentation of text. Colour is a subtle system for communication, and can add or detract from aesthetic value. Colour encoding of application page backgrounds can effectively communicate information. Similarly, colour encoding of important information to highlight its immediacy can be very valuable. However, colour is not culturally neutral: white/black can have specific connotations which any dictionary will reveal. It must be remembered that colour cannot always be seen – colour blindness to specific colours may mean that important information may be missed by some content users!

Colour is not the same on every system. For example the Web-safe colour palette consists of 216 colours that will not shift or dither on computers set to display 256 colours. These 216 colours will display consistently across all platforms and Web browsers. Many HTML editors will NOT pick colour-safe colours.

Rule-of-thumb for recognising Colour-safe codes in HTML

- Colour codes are 3 hex pairs representing red, green & blue.
- If colours-safe, all three pairs will be one of 00,33,66,99,AA,CC,FF
- IE: color="#00EEFF" is colour safe
  color="#31FF23" is NOT

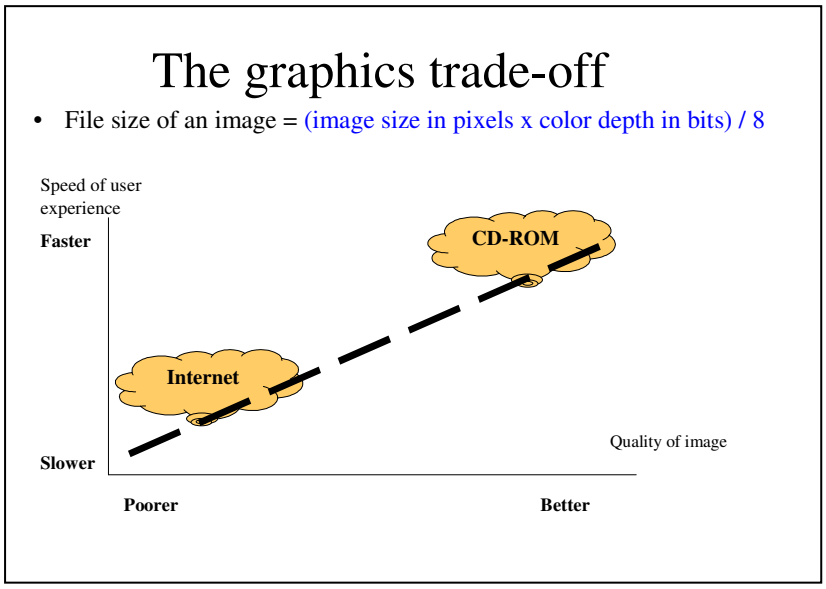
IMAGES

The representation of concepts in image form has been an effective communication technique dating to the most ancient times of human activity. In development content of any sort, images can serve many roles, including:

- To represent multiple languages
- For advanced text presentation
- To illustrate a concept or an idea
- To create a sense of mood or ambience
- To communicate information or instructions
- To project a corporate "image"
- As a website hypermedia "button"
- To provide Navigational direction
- As a metaphor for some meaning
- To illustrate an analogy
- To provide continuity between disparate elements

Or more simply: "a picture says a thousand words".

Digital images can be captured in a variety of ways. Bitmaps depict the graphic as an array of dots, called pixels. The specific resolution of the image cannot be enhanced, because a pixel has specific information in "dots". Vector images depict the graphic using lines and arcs that have a mathematical relationship. They are not distorted when enlarged, but may take more time to display.

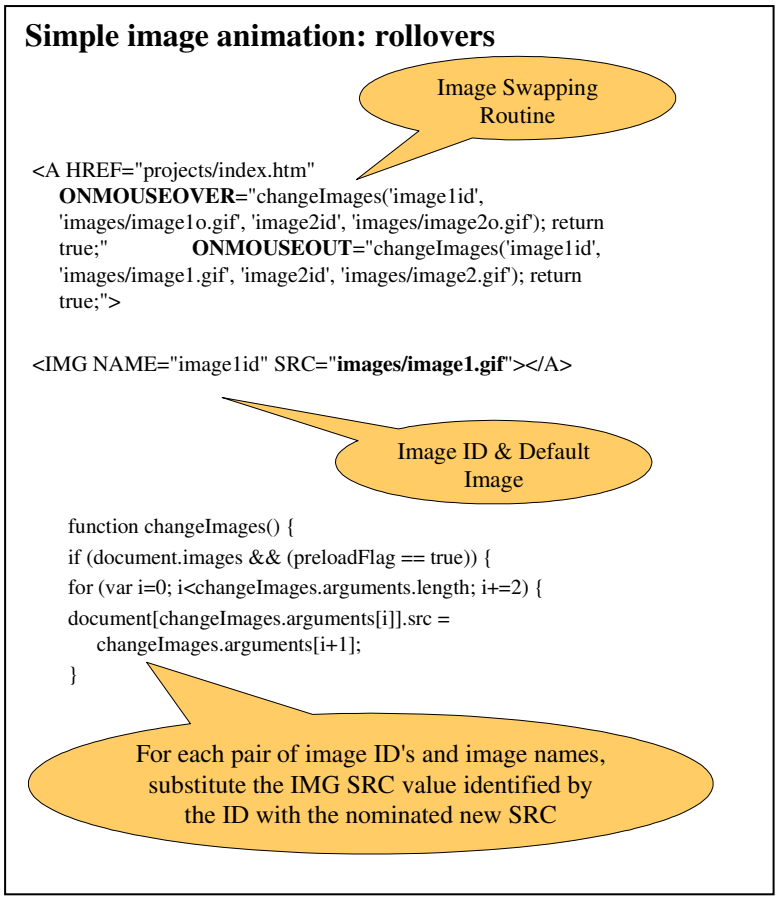


GIF and JPG are widely used compression techniques for image delivery on the WWW. JPG has better photographic representation, with a range of compression options. GIF, as a bitmap image encoding method, gives good clarity for buttons, navigation, background images, etc with minimal space usage – it is limited to 256 colours.

With images, and as we will see also with audio and video, there is a direct trade-off in between the speed of the user experience and the size/quality of the image:
Image capture can be achieved using Digital Scanners, Digital Cameras, Analogue and Digital Video cameras and through use of Image libraries. Copyright considerations are significant and often ignored.

ANIMATION

Multimedia animation is simply the movement of an image or object on the computer screen. As in movies, the perception of movement is caused by the rapid display of several still images or frames. An example of animation is the use of simple roll-overs in websites, where the movement of the mouse over an image causes the image to alternate with a secondary image, giving an animation effect. Some image formats, such as GIF, support animation through the embedding of multiple images within one file with simple rules for the sequencing of the image. More complex animation generally requires the use of plug-ins of one sort or another, popular examples including: Java applets, ActiveX controls, QuickTime VR, and Macromedia Flash.



3-D animation is itself a graphical design speciality. Tools are available such as Strata Studio Pro, Softimage, LightWave 3D, and 3D Studio MAX.
3-D animation in games such as DOOM and Flight Simulator have proven the effectiveness of VR (Virtual Reality) simulations can be sufficiently efficient to run at the desktop level. Specialist VR hardware is being combined with visual animation techniques to create vivid animation sequences.

Animation techniques such as *Morphing* and *Tweening* simplify the design of animations. Morphing involves the visual transformation of one image to another, and is commonly embedded in many tools, such as Flash. The content authoring systems allow simple visual mapping of the transformation process. The process of *Tweening* involves the generation of specific frame sequences between two images to given the impressing of smooth transition. Animation of text is quite easily defined through Tweening. Like all visual effects, the use of animation needs to be target carefully for actual effect.

AUDIO

Authoring of Audio content can include both voice capture and the capture of Ambient and Musical effects. Digital Audio can be captured either directly in digital form, or translated from analogue input sources. To understand the process for digitizing analog sound, one approach is to visualize the sounds as a series of recurring waves. These are called a waveform. The waveform shows both the

sound's volume (the higher the wave the louder the sound) and its frequency or the pitch (the closer together the waves, the higher will be the pitch). Frequency is measured in hertz (Hz), and a waveform that repeats every second is equivalent to 1 hertz . A waveform that repeats 1,000 times a second is equivalent to 1,000 Hz or 1 kHz (Kilohertz). In order to convert analog sounds, to digital sounds, thousands of samples are taken of the sound waves and recorded in binary form as digital content. The rate or frequency at which these samples are taken is called the *sample rate.* The more samples taken the greater the storage required to store the audio in its raw form, but the better the quality of the audio. The digital audio data file is as yet still uncompressed.

Typical sample rates are 11.025, 22.05, and 44.1 kHz, and typical resolutions are 8 and 16 bits. In a Microsoft environment, audio capture formats such as AVI have been popular at the desktop. File sizes for high fidelity capture of audio can be very large, and unsuitable for effective distribution on CDROM, and particularly over broadcast media such as the WWW. This limitation has driven the development of effective Audio Compression techniques, called **codecs.** Dramatic advances in the development of codes have seen very effective voice and music encoding approaches that yield much smaller file sizes at a very high quality. Specialised voice and music codecs result in better results depending on the audio sources.

Where using audio in the multimedia authoring process, care must be taken for the suitability of the audio:

- Using sound can enhance most multimedia applications. Indeed, sounds are essential for certain applications, such as educational titles that teach foreign languages or music.
- Like text, overuse of sound effects can be detrimental to information delivery.

For some people, understanding a concept is easier when sound is used. Anticipating how the intended audience will respond to various sounds can guide the choice of audio effects. For example, adults may respond better than children to the use of classical music as a background effect. Audio can be essential in a context where disability groups are being addressed. For others, audio may be an unnecessary distraction for which options to opt-in or opt-out are essential.

Once again, it is essential to be conscious of the architectural limitations of the runtime engine used for delivery of the audio content. Playback Systems are notoriously variable, and:

- Not all computers are capable of playing sounds.
- If they have CD-ROM/DVD, they are likely to have audio capability
- Don't put essential information in audio if it can be avoided
- Give the user control: especially, let them turn off or lower non-essential sound

Storyboarding is a common approach to integration of audio and other multimedia elements in a final design. The process of story boarding involves the visualisation of each of the stages of audio delivery on paper with a narrative for the content designer indicating at which stages different multimedia elements are deployed.

On the WWW, audio has been immensely successful. There are many methods for integration of audio, but two popular approaches are:

Audio on Demand

An audio encoded file is available for either download or immediate initiation on the web at the discretion of the user. This is generally suitable for delivery of relatively small audio segments.

Streaming Audio

Content Streaming servers delivery the audio in a continuous stream which the user can join in much the same way as standard mass media broadcasting. The user does not generally have the option to download content independently.

Three commonly used formats for audio encoding are:

Real Audio Player

Has a free client
As proprietary audio authoring and delivery systems
Has excellent voice & music compression
Is a "defacto" industry standard – with wide adoption
Licensing for streaming can be very expensive

Microsoft Windows Media Player

Has a free client & and the server software is bundled with the NT operating system license
It is generally not as scalable

MP3

Has a free client
Is widely used for offline audio sharing and distribution
Is not generally used for streaming.

The potential presented by Peer-to-Peer sharing of content has seen the rapid uptake of systems for audio content sharing, a development which has severely stressed the Copyright regulations internationally.

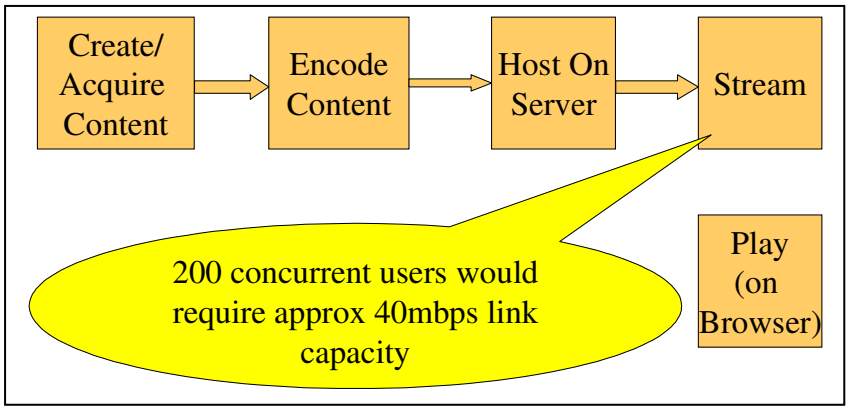
VIDEO

Video delivery has been immensely popular for the development of educational and self-paced learning systems. The more recent integration of video in online and CDROM reference publications such as Britannica and Encarta have shown the effectiveness and acceptance of video integration into multimedia. More recently, video streaming techniques are seeing a gradual uptake of media in online services such as newscasts over the WWW and the development of broadband magazines with integrated text, video and audio.

Once again, compression techniques have been essential for the effective dissemination of video content. Without effective compression, the massive size of digital video content makes distribution impossible, even on digital media such as the DVD. Digital Video Compression Video compression is a process that reduces the video file size while maintaining an acceptable image quality. There are two types of compression: *lossless* and *lossy*. Lossless Compression maintains the image fidelity and is used when there is a need to display the image exactly as originally created.

JPEG files use *lossy compression* and can achieve a compression ratio of 20:1. For example, if the pre-compressed file is 200 KB, the compressed file would be 10 KB. Lossy compression involves eliminating data in the image and can result in a compression ratio as large as 200:1, but at the cost of fidelity to the original image. Lossy compression is used for video because of the need for very high compression ratios over a narrow bandwidth. The loss in data may not be noticeable to the viewer in certain contexts, but can have a dramatic impact on the ability of a web server to efficiently deliver a page to users on a wide range of bandwidths. Compression techniques are sufficient to encode (and encrypt) a full movie at high quality on a single DVD (approximately 6GB), and reasonable results are now possible with compression of 2-3 hours of video into less that 1GB (or approximately 2 CD's), with some loss of quality. The file sizes are still substantial, and the compression required for WWW delivery of video is such that viewing for entertainment purposes over the internet is still impractical. The gradual uptake of broadband may change this in time but at present few organisations deploy significant video content over the internet. Media organisations, such as news, use video streaming and video on demand with very high compression and low requirement for aesthetics of the resulting video.

Media Streaming:



SYSTEMS THINKING

Multimedia delivery is still technology-platform bound, culturally bound, and has considerable trade-offs between quality and timeliness of content delivery. Video delivery through streaming is as yet in its infancy with only a negligible take-up compared to traditional broadcast media. However, the integration of image, video and text in the web browser has had profound implications for the adoption of multimedia technology in the general community. Obsolescence is a major issue, and copyright control presents a dilemma for content creators.

QUESTIONS

1. A student prepares a thesis in medieval history and literature. The thesis is prepared in Word using a plug-in for Ancient Greek (modern Greek is somewhat different). Discuss the issues the University may face in delivering the content online.

2. What issues will the University have to address if they wish to make this thesis available online? What formats might be considered?

3. Name an SGML initiative for the management of literary text resources?

4. Describe three types of obsolescence that are problems in the medium and long-term management of multimedia content.

FURTHER READING

AGOULMINE, N., DRAGAN D., GRINGEL T, HALL, J. ROSA E. AND TSCHICHHOLZ M. (2000): Trouble management for multimedia services in multi-provider environments. *Journal of Network and Systems Management* **8**(1):99-123.

AINSLEY, H., GHAOUI, C. and WHITELEY, K. An OO model to generate knowledge structures for authoring instructional hypermedia. *Proceedings of the 26th Euromicro Conference. EUROMICRO 2000. Informatics: Inventing the Future. IEEE Comput. Soc. Part vol.2, 2000*:35-42 vol.

AMATO, G., RABITTI, FAUSTO AND PASQUALE SAVINO (1998): Multimedia document search on the Web. *Computer Networks and ISDN Systems* **30**(1-7).

BERGERON, B.P. and BAILIN, M.T. (1996): Collaborative hypermedia development: considerations for academic publishing. *Educational Technology Review*(6):19-23.

BEZZI, M., DE POLI, G. and ROCCHESSO, D. Sound authoring tools for future multimedia systems. *Proceedings IEEE International Conference on Multimedia Computing and Systems. IEEE Comput. Soc. Part vol.2, 1999*:512-17 vol.

BIANCHI, N., ET AL. (1996): Multimedia document management: an anthropocentric approach. *Information Processing & Management* **32**(3):287-304.

BONARDI, A. and ROUSSEAUX, F. Composing interactive virtual operas. *Creativity and Cognition. Proceedings of the Third Creativity and Cognition Conference. ACM. 1999*:144-7.

CAUMANNS, J. (2000): Bottom-up generation of hypermedia documents. *Multimedia Tools & Applications* **12**(2-3):109-28.

COPELAND, D.R., CORBO, R.C., FALKENTHAL, S.A., FISHER, J.L. and SANDLER, M.N. (2000): Which Web development tool is right for you? *It Professional* **2**(2):20-7.

DAVIES, B., LIENHART, R. and YEO, B.L. (1999): The video document. *SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering* **3846**:22-34.

GUPTA, U. and CHAUDHURY, S. (1999): A multimedia data model. *Iete Journal of Research* **45**(5-6):283-91.

KELLY, B. (2000): Will it all fit together? The need for standards and the technical challenges. *Multimedia Information and Technology* **26**(1):67-72.

LIU, P.A.H., LIANG H (1999): Multimedia Document Systems From Authoring perspective. In *Internet and Multimedia: Systems and Applications*. FURHT, B. (ed), IEEE Press.

MENDES, M.E.X. and HALL, W. (1999): Hyper-authoring for education: a qualitative evaluation. *Computers & Education* **32**(1):51-64.

MORGAN, P.J. and SKINNER, H.S. (2000): Multimedia courseware production systems. *International Journal of Continuing Engineering Education* **10**(1-4):47-58.

Chapter 3: CONTENT MARKUP

What Is Content Markup?

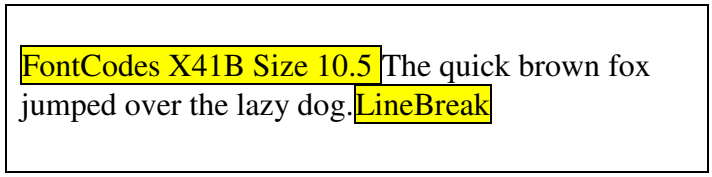
The invention of the earliest compilers implied the need for *encoding* systems to render data and programs in digital form. The encoding of text content was variously accomplished through standards such as BCDIC, EBCDIC and ASCII. Fundamental text forms of English character encoding date from such early encoding systems. *Parsing* is the process of traversing the structure of encoded data. *Classification* is the process of attributing semantic meaning to the markup or encoding system. Parsing is used to interpret and understand search queries, program scripts, and to provide the mechanisms to interpret and traverse text content.

Word processing systems have been one of the most successful text encoding applications. Word Processing systems add annotations to the text that you type 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. Quite early, it was realised that this "end result" is also a "dead end" in the electronic arena – propriety coding methods to specify fonts, layout, and structural elements of how the text should be printed were difficult to interchange between different word processing systems. The popularity of Word Processing systems in business exemplified by Wang in the 1980'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. In the US, businesses alone produce 92 billion documents, making the sharing of information is a vital process.

Markup is a set of formal rules for the separation of underlying content from information regarding its formatting and presentation. When we refer to "markup" this means *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. Markup may also indicate to the system which generates final output what the heading styles, page layout, margins and other information relating to the document structure. In this sense markup *licenses* certain inferences regarding the content, either:

> *constitutive* – that is, regarding the structure and order of the content; or
> *interpretive* – that is, additional narrative or information regarding the content which presents claims, interpretations or understandings about the text. (Sperberg-McQueen, Huifeldt et al. 2000)

Most early word processing systems used *Procedural Markup*. Procedural Markup refers to the embedding of the specific styles within the content itself. Thus the internal coding of the content might look something like:



Procedural Markup generally does separate the content elements from the presentation formatting. Procedural Markup codes may refer to specific fonts and styles that exist within only in content creators environment.

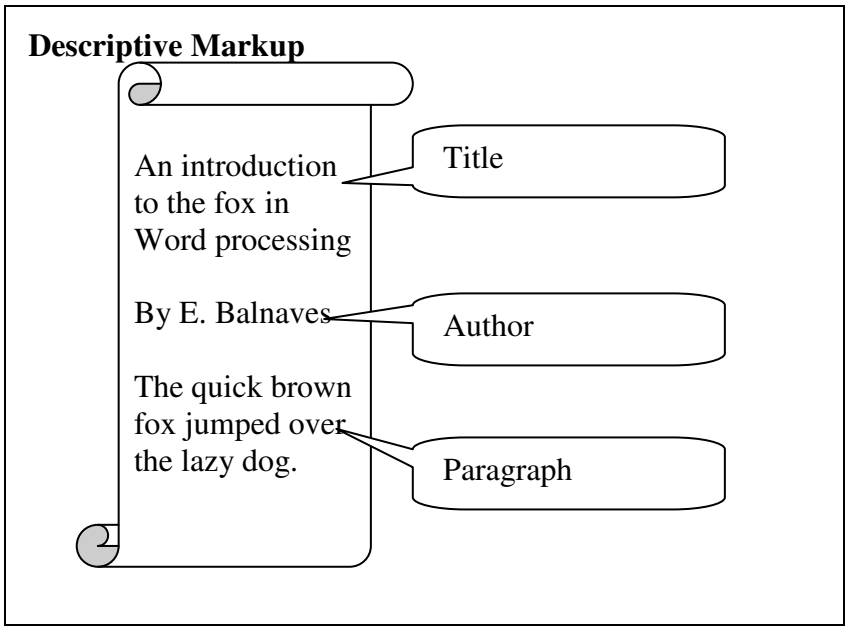
Word processing systems moved gradually from line-based entry methods to screen based display of content, and progressively hid the Procedural Markup coding. *WYSIWYG* (What You See is What You Get) exemplified the process of Procedure Markup, and has been an immensely intuitive method for end user word processing.

However, Procedural Markup has severe limitations in portability. 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. Furthermore, numerous different applications were exploring Procedural Markup techniques unique and proprietary to their word processing engines.

SGML

SGML, or Standard Generalized Markup Language, grew from the limitations that were implicit in Procedural Markup. A key objective of SGML was to move away from highly a specific markup syntax, often proprietary to the document system and to develop a universally understood set of standards for device-independent, system-independent representation of texts in electronic form. SGML was formalised in the International standard ISO 8879 in 1986.

SGML is a "meta-language" which has allows the definition of the rules by which markup is undertaken in a hierarchical manner. It is a *Descriptive Markup* language. That is, it describes the "structure" of the content, but not specific "styles". It is oriented toward separating "content" from "presentation" and embedding with the content not the specific style but structural rules for "content" – eg chapter headings, paragraphs, etc. The final presentation of content therefore takes the content and its structure, and applies a "style" sheet in order to produce the final result.



The specific markup tags are not defined in SGML, it simply defines rules for a language to describe those markup tags, called the *DTD (Document Type Definition).* Armed with a DTD to describe the encoding standards, a document can be encoded and exchanged and interpreted by anyone receiving that document. Formatting of the presentation can then flexibly be adjusted for any relevant output media: print, screen, etc. Once a *DTD* is defined describing the encoding method, the mark-up process encodes the document following the hierarchical standards described in the DTD – following the example above, the encoded result might look something like:

```
<TitlePage>
    <Title> An introduction to the fox in Word processing</Title>
    <Author> By E. Balnaves</Author>
</TitlePage>
<Paragraph> The quick brown fox jumped over the lazy dog.
</Paragraph>
```

The mark-up *elements* are the tags, such as "<TitlePage>". The mark-up content itself is known as the *Parsable Character Data (PCDATA)* and may itself be character encoded for language reasons (using for instance UniCode) and to differentiate embedded characters like "<" from Element tags.

Elements may themselves be characterised in more detail to describe their specific formatting rules. These formatting rules are called *Attributes.*

SGML is now widely used as an information exchange standard by organisations such as AAP - The Association of American Publishers and AFP - Agence France Press.

HTML

HTML can loosely be described as a child of SGML. As a scheme for content markup it has been immensely successful, and is the defacto standard for content creation on the World Wide Web. It is marked by its simplicity, robustness to errors 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, specific font descriptors such as <b>, <I> and <u> are all typically used in HTML as procedural markup codes. The embedding of such coding in HTML has limited the portability of web content to other environments. The W3C has “deprecated” such coding: indicating to browser developers that they should begin to phase out their use. Such a process, however, is likely to be very slow, given the widespread deployment of content which uses such markup encoding. HTML encoding which is fully procedural and honours a more standard SGML encoding structure is described by W3C at http://www.w3.org/TR/REC-html40/sgml/dtd.html.

Central to the separation of content from presentation in HTML are *cascading style sheets (CSS).* Cascading style sheets allow the creation of a separate style guide that may be consistently applied to all HTML pages. The HTML pages can, theoretically,

be cleansed of specific presentation markup tags, such as <font> <b>. While conceptually sound, the CSS has been plagued by browser incompatibilities, varying browser defaults, versional differences in implementation over time: all profound issues in software obsolescence. As a result, many HTML pages continue to use a hybrid set of CSS and procedural mark-up elements to achieve an acceptable level of presentational consistency.

XML

XML developed initially as an SGML-based encoding standard for information interchange between applications, primarily for transient data interchange. 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.

WYSIWYG has considerable end-user advantages for manipulation of content
It is, however, the antithesis of SGML philosophies. Using WYSIWYG content and form are brought together in a manner which is quite intuitive to the content author. In SGML they are maintained separately.

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*
- a structural definition, the DTD, describing hierarchically how *Elements* can be used.

DTD

The Document Type Definition (DTD) defines the strict hierarchical standards for construction of a particular XML content set - essentially the markup rules for a particular context. The following simple DTD defines a simple structure for a poem: each poem can have optionally a title, and one or more stanzas. A stanza comprises one or more lines. A line contains raw data (#PCDATA):

```
<!ELEMENT poem - (title?,stanza+)>
<!ELEMENT stanza - (line+)>
<!ELEMENT line - (#PCDATA)>

The "?" means optional
The "+" means may occur many times
#PCDATA means the actual content
```

Attributes can further characters exactly the format that an element can take:

```
<!ATTLIST poem
    id   ID   #implied
    status (draft | revised | published) >
```

Sample XML file

```
<?xml version="1.0">
<package>
<title>Norton Utilities</title>
<version>3.5</version>
<vendor>Symantec</vendor>
<platform />
<os />
<description>A hard disk utility program</description>
<copies>1</copies>
</package>
```

The DTD defines the structural rules for using ELEMENTS to mark up a document. The markup says nothing about the final look and feel - the presentation - of the content. This vital last stage of transformation can be performed using Extensible Style Sheets (XSL). The XSL is a style sheet defining the template for generating actual output (eg HTML) from an XML data file.

While XML developed as an encoding methodology for transient information interchange (for example E-commerce), it has had considerable success in other areas: MusicML, NewsML and many other examples. Unlike SGML, there has been a rich contribution of software tools for the manipulation of XML content from the development community. The proliferation of core support systems has reinforced the trend for XML to become a central standard for content markup generally.

XML SCHEMAS are also being introduced to attempt to more concretely define XML structure that are recognisable as a hierarchical structure definition than the more cryptic syntax of the DTD. XLINKS attempt to emulate the role of the hyperlink in HTML, but attempt to extend the concept by allowing hyper linking direct to intermediate locations within a target document.

XML MARK-UP CHARACTERISTICS

XML is case sensitive. The following element:

```
<Actor>John Wayne</Actor>
```

is not equivalent to

```
<ACTOR>John Wayne</ACTOR>
```

Unlike HTML, white space is not ignored.

```
<phrase>What you see    is what you get</phrase>
```

is not equivalent to

```
<phrase>What you see is what you get</phrase>
```

All XML elements must be closed.

```
<image url="picture.gif"/>
```

You cannot overlap elements. For example, the following code:

```
<actress>Judy<Surname>Davis</actress></Surname>
```

is improper syntax. The following is correct:

```
<actress>Judy<Surname>Davis</Surname></actress>
```

All attribute values must be in quotes:

```
<photograph url="judy.jpg" width="300px"/>
```

It is apparent that XML is more syntactically sensitive than HTML. However, there is also the risk of implementation-specific speciation of XML techniques as suppliers diverge in the technical implementations.

PCDATA

Like SGML, the target of the encoding process, the actual content, is placed in a *PCDATA* section surrounded by then *Elements* that describe the content. The content of the PCDATA must be encoded to eliminate the use of any special characters that are used for the mark-up elements. The encoding of the PCDATA section is intended to render character that would otherwise be confused with the element TAGS.

If you place a character like "<" inside an XML PCDATA section, a *parsing error* will result as this will be interpreted as part of a new element. The following XML would cause a parsing error:

```
<note>When errors occur < 100 times, no warning is needed</note>
```

To avoid this, you have to substitute the "<" character with an *entity reference*, as follows:

```
<note>When errors occur &lt; 100 times, no warning is needed</note>
```

The entity references used by XML are limited to those which can be used within an *element* tag:

<	<	Less than
>	>	greater than
&	&	ampersand
'	'	Apostrophe
"	"	quotation mark

CDATA

Everything inside a CDATA section is ignored by the XML parser. If the content contains a lot of "<" or "&" characters the XML element can be defined as a CDATA section and translation is undesirable a CDATA section can be used. Characteristically binary objects, images and other non-text content will require a CDATA section. A CDATA section starts with "**<![CDATA[**" and ends with "**]]>**":

For instance, the following Python script would normally have to be *entity reference* encoded because of the presence of the ">" character:

```
If myincome > 20000000:
        Print "Warning: data entry error"
```

It could however be rendered in a CDATA section as follows:

```
<script>
<![CDATA[
        If myincome > 20000000:
                Print "Warning: data entry error"
]]>

</script>
```

MERGING XML DATA SOURCES

Element names are not fixed and universal in XML: they are defined by the DTD. If several different XML sources are bought together, each with a different DTD, there is the possibility of naming conflicts. XML Namespaces provides a method to avoid element name conflicts.

For instance, the following two XML sources use a tag <product>.

XML source 1:

```
<product>jam</product><price>$6.40</price>
```

XML source 2:

```
<number>1</number>
<number>2</number>
<product>2</product>
```

These two documents have very different content definitions for the <product> element. XML *Namespaces* can be used to resolve this conflict. The XML namespace is a prefix added to each element that identifies the relevant DTD for the element. In the example above, conflicts could be avoided through the following Namespace encoding:

XML source 1:

<productentry xmlns:ns1=http://www.namespacesource1/>
<ns1:product>jam</ ns1:product>
< ns1:price>$6.50</ ns1:price>
</productentry>

XML source 2:

<calcthus xmlns:ns2=http://www.namespacesource2/>
<ns2:number>1</ns2:number>< ns2:number>2</ ns2:number>
< ns2:product>2</ ns2:product>
</caclthus>

The inclusion of 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 (.

Using XML – the Parser

XML is not generally intended to be read or interpreted in raw form, except during the markup stage. Its role as a standardised means of information exchange requires is realised through the use of a *parser*. The purpose of the parser is twofold: to validate the XML document against its DTD, and to facilitate the interpretation of the XML document. One approach to the development of XML parsing components has been the generation of a complete document tree which the parser exposes as a hierarchical document structure. The DOM (Document Object Model) approach can therefore reference any element in the XML document directly and immediately. This approach generally requires the full parsing of the XML document before any element can be referenced. Another approach, particularly for long XML documents, is a stream approach to the parsing: content is parsed serially and the XML contents can be explored in sequential hierarchical order. This approach is especially applicable to large XML documents. An example of a stream approach to XML document parsing is SAX (Simple API for XML). Both Python and ASP include modules for parsing and traversing an XML document.

XSL – Extensible Style Sheets

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 mark-up. Unlike CSS however, XSL is a template-oriented solution with some scripting-like constructs for sorting, selection and looping based on conditional constructs.

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. Simple XML editors will give a hierarchical view of XML data for simple data maintenance. However, XSL is essential for the final stage of rendering XML in a final presentational form.

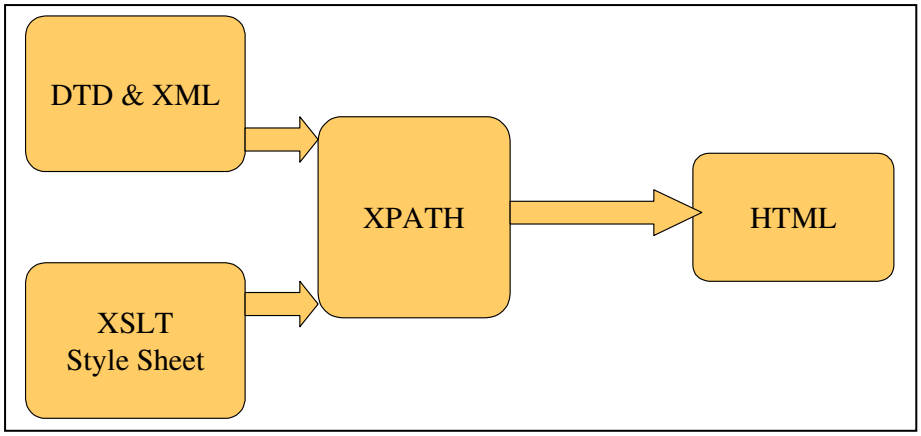
Content Style Sheets (CSS) could still be used to display XML. However, XSL provides considerably more powerful control in the final presentation of the content, because it supports not only the rule-based presentational information available to CSS, but also a scripting language for much more structured control of the presentation.

XSL is a World Wide Web Standard recommended by the World Wide Web Consortium. The elements of XSL are:

> XSLT - a language to transform XML to a final presentational format
> 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. One key function 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, footers, and may involve the sorting or re-arrangement of the source XML content.

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. This process of transformation is another example of a *parsing* process. The following illustrates the process for generating an HTML document using XSL:



Internet Explorer 5.0 (and later versions of IE) contains embedded XSL parsing features. It is possible with IE 5.0 to send XSL and let the browser render this in the format you have 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. That is, a Web server will transform any XML content and deliver to the web browser standard HTML.

SUMMARY

Content encoding and markup involves many choices. Some of those choices will determine the longevity and ephemeral nature of the content. Considerations include:

- the most effective markup approach for immediate communication of the content
- issues of future re-use of the content.

QUESTIONS

1. What is the role of XSL in content markup?
2. What is the difference between descriptive markup and procedural markup?
3. What are the similarities between HTML and XML? What are the key differences?
4. Contrast the strengths and weaknesses of SGML with XML.

FURTHER READING

ONLINE REFERENCE GUIDES TO CONTENT MARKUP

HTML

NCSA Beginner's Guide to HTML
http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

HTML: An Interactive Tutorial for Beginners -
http://www.davesite.com/webstation/html/

http://www.w3schools.com/ - a excellent tutorial site on HTML, CSS, XML and Scripting

http://www.w3c.org/ - the W3C consortium home page, with useful links to documentation on all aspects of HTML, CSS, XML.
especially

http://www.w3.org/MarkUp/ - Markup
http://www.w3.org/TR/REC-CSS1 - Cascading Style Sheets
http://www.w3.org/TR/REC-CSS2 - Cascading Style Sheets
http://www.w3.org/TR/ - Publications directory

XML

http://www.xml.com
O'Reilly portal for XML information

http://msdn.microsoft.com/xml/default.asp - The Microsoft XML Center with lots of examples and downloads – XML, DOM, SAX, XSLT

www.xml.org/registry/index.shtml - The OASIS repository
- a repository of XML vocabularies (standard Document Definitions and Schemas)

www.biztalk.org - Microsoft's BizTalk repository for interchange vocabularies (standard Document Definitions and Schemas)

www.rosettanet.org -RosettaNet with technical papers and XML vocabularies

CONTENT AND CONTENT SERVER STANDARDS

BORENSTEIN, N. and N. FREED (1992). RFC 1341: MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies, W3C. 2002.

CHAMBERLIN, D., J. CLARK, et al. (2001). XQuery 1.0: An XML Query Language - W3C Working Draft 07 June 2001 (http://www.w3.org/TR/xquery), W3C.

CROCKER, D. H. (1982). RFC 0822: ARPA Internet Text Messages, W3C. 13 July 2000.

FRANKS, J., P. Hallam-Baker, et al. (1999). RFC 2617: HTTP Authentication: Basic and Digest Access Authentication. 2001.

GETTYS, J., J. Mogul, et al. (1999). RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1. 2001.

GOLAND, Y. Y., E. Whitehead, et al. (1999). RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV:. 2001.

GUENTHER, R. (1997). RFC 2220: The Application/MARC Content-type. 1991.

JOURDAN, M., C. Roisin, et al. (2000). "A scalable toolkit for designing multimedia authoring environments." Multimedia Tools & Applications 12(2-3): 257-79.

KUNZE, J. (1999). RFC 2731: Encoding Dublin Core Metadata in HTML, IETF. 2001.

LASSILA, O. and R. Swick (1999). Resource Description Framework: (RDF) Model and Syntax Specification -W3C Recommendation 22 February 1999 (http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/), w3c.

LIE, H. W. and J. Saarela (1999). "Multipurpose Web publishing using HTML, XML, and CSS." Communications of the ACM 42(10): 95-101.

MATHEWS, B., D. Lee, et al. (1998). Vector Markup Language (VML): World Wide Web Consortium Note 13-May-1998, W3C: http://www.w3.org/TR/1998/NOTE-VML-19980513.

NEBEL, E. and L. Masinter (1995). RFC 1865 - Form-based File Upload in HTML. 2001.

NEIGUS, N. (1973). RFC 0542: File Transfer Protocol. 1992.

SOLLINS, K. and L. Masinter (1994). RFC 1737: Functional Requirements for Uniform Resource Names, ITEF Network Working Group. 2001.

SPERBERG-MCQUEEN, C. M., C. Huifeldt, et al. (2000). "Meaning and Interpretation of Markup." Markup Languages: Theory & Practice 2(3): 215-234.

Chapter 4 – Scripting

INTRODUCTION

Scripting languages are an accessible programming engine now embedded in many tools. They can be used to automate otherwise mundane and repetitive tasks. They appear as "macros", "scripts" and "interpreters". The evolution of programming languages has seen the exponential accretion of capabilities as innovations have seen the transformation of laborious automation activities into simple one-line statements. The focus in use of these technologies has branched beyond the activity itself to the evolution of Scripting technologies that integrate programming as one component in the creative exercise. The Scripting language, from the early spreadsheet macros and simplified interpretive programming languages like BASIC, has become integral to the authoring tools on the desktop. They are an essential adjunct for text parsing and for the development of dynamic website content.

Traditionally, programming languages have required several phasis of processing to render an operational (executable) result suitable for a particular computing environment. The Central Processing Unit of a computer has a finite set of instructions that perform relatively simple functions very rapidly. The transition of a programme from its human-composed form to a computer-operable programme involved a process of “compiling”, or translating from a high level “language” to a lower level of coding (a “macro” level) nearer to the instruction set understood by the computer itself. A next phase often involved the “linking” of disparate elements of the program to produce a final executable product. This final result might or might not be able to run in anything other than the operating system and CPU within which it was compiled.

Scripting languages are often interpretive. That is, there is no such intermediary process required to make the transition from composition of the programme through to its execution. Generally scripting languages focus on ease of comprehension and portability rather than strict enforcement of programming structure and conventions. As a result, they are often loosely “typed” – for example, the programme developer may not need to declare in advance what identifiers in the program are used as numbers and what elements are used for text.

In summary, scripting languages:

- Are generally interpretive
- Often have powerful text and list manipulation features
- Focus on ease of development and deployment rather than efficiency
- Will often be loosely "typed"
- Are usually very portable
- Are ubiquitous, and often embedded in applications:
    - Visual basic for applications
    - PHP
    - Python
    - Perl
    - Javascript
    - Microsoft Active Server Pages (ASP)

Scripting will be found in different application contexts according to their function:

(1) Applications with embedded scripting. These include macro languages such as Visual Basic for Application used to extend the capability of a user application.
(2) Server-side scripting languages. These include PHP, ASP pages using VBScript, Java Server-lets and CGI applications such as PERL and Python which operate entirely on the host as an application server, delivering to the user client the information pages required.
(3) Client-side scripting languages. These include JavaScript (embedded in many browsers) and VBScript (embedded in Microsoft Internet Explorer), and are used to distribute the processing between the client browser and the server, sifting some of the processing logic from the server to the client (for example, scripts to validate entries supplied in an online form before it is posted to the server).

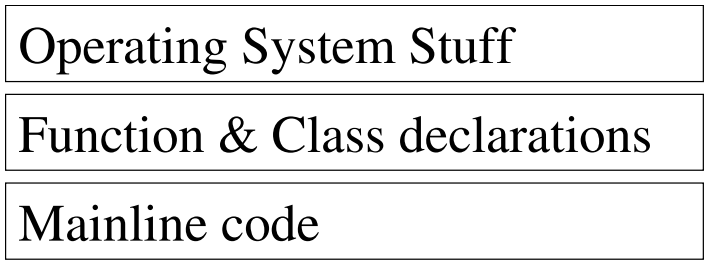
This section will explore the use of Python and Microsoft Active Server Pages (ASP) as scripting languages for server-side scripting of dynamic websites. Python is a useful exemplar of the role of programming languages in this context. It is a fully object oriented programming language which has many "plug-in" modules for internet-related scripting and programming. It can be used in both an interpretive and compiled context. It is loosely typed: that is, variable declarations are not required but can be assumed based on the context of their use. It has very powerful text and array manipulation functions. Microsoft ASP uses ActiveX scripting (and usually with VBScript), which, like its cousins Visual Basic and Visual Basic for Applications, is a feature-rich language that facilitates the rapid development of dynamic websites. While less structurally elegant that Python, the ease of integration in websites both at client-side (that is, in the client Browser) and on the web server makes it very popular.

SCRIPT STRUCTURE

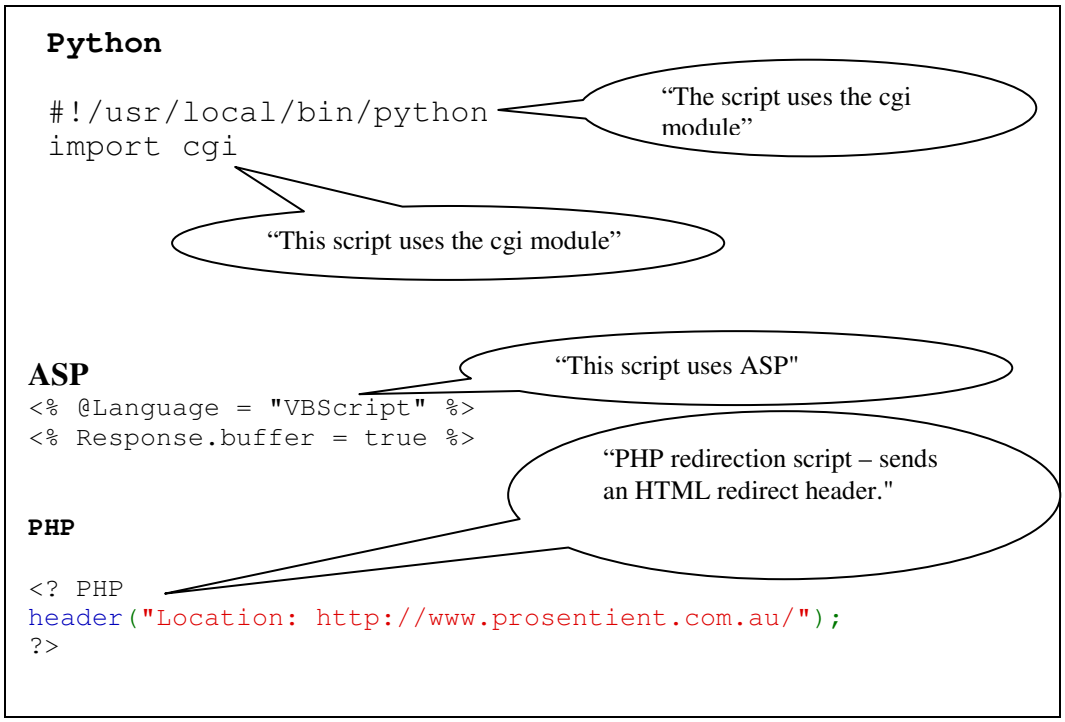
Python, named after the BBC comedy series rather than the reptile, was developed by Guido van Rossum in the open-source movement. It is relatively stable and highly portable, if not the most speed-efficient of languages. Concepts used in Python are common to other scripting languages. This section will focus specifically on the use of Python in a scripting context as a CGI application running on websites. (Lutz 1996) has a comprehensive manual on the use of Python as a programming language. The Python website (www.python.org) contains software downloads for most architectures, reference materials and many guides to programming and scripting in Python. The specific examples shown here are for Python scripting specifically in a Unix-style context (Unix, Linux, etc), but they are applicable to most architectures in which Python may be implemented.

Microsoft Active Server Pages(ASP) have extended the venerable old programming institution BASIC into the web site scripting. Well coded ASP should reflect the same structure as Python, although the explicit rules for structure are not as strong. ASP will be illustrated in the context of an Application Server.

The structure of a script is relatively simple. Three broad sections of code can be identified:



The **Operating Systems Stuff** identifies the script *as* a Python script, as distinct from other scripting languages. It also identifies any modules and plug-ins used for this specific script. A typical example of this element of the code is:



Python is inherently modular in its design. Its own logical programming constructs are themselves divided into modules, and this has encouraged the addition of other modules to be used within Python. The scripting language also encourages the modularisation of your code – that is, the dividing of your software design into separate logical elements each performing a specific function. Such modularisation can facilitate the re-use of code in many different contexts – this concept will be explored later in this section. PHP and ASP are natively written for web service delivery and have many scripting constructs that make it easy to generate web pages.

Functions and Class declarations are blocks of code that are used frequently within your CGI script. These logically divide your code into manageable components and activities. Functions are for small procedural elements that are used frequently (rather than repeating the same code many times). Classes are for code that can be

re-used across many applications. Modules are for large discrete elements of code that go together to build a larger application. Python, PHP and ASP support functions.

The **Mainline Code** is the heart of your script: where the actual content generation or logic will be performed.

Getting a Python CGI script to display a web page within a Web Server is quite easy – simply using the *print* command to display HTML content in the browser. There are two requirements: a) you must supply the web server with content headers that tell it that you are generating HTML code, and possible the language context you are generating; b) you must generate the tags and the content to be used. You should, of course, always generate well-formed HTML code. In the case of ASP it is not essential to supply content headers, although these may be useful if you need to specify language-specific header information for your web page. ASP code may be *inline* with the HTML, or may generate the HTML using an output class called *response*. Notice below that the ASP code blocks are surrounded by <% %>. The web page will act on the programming scripts between these code blocks at the time the script is called.

PYTHON SCRIPT TO DISPLAY A WEB PAGE

```
#!/usr/local/bin/python



print "Content-type: text/html"
print
#
# My web page
#
print """<HTML><HEAD>\n
<TITLE>Sample web page</TITLE>\n
</HEAD>
<BODY>
<H1>Here is my Heading</H1>
<P>This is my page content</P>\n
</body>\n
</html>"""
```

HTML Output commands. Note that ASP automatically generates a basic Content Header

ASP SCRIPT TO DISPLAY A WEB PAGE

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<html><head>
<TITLE>Sample web page</TITLE>
</HEAD>
<BODY>
<H1>Here is my Heading</H1>
<%
Response.Write "<P>This is my page con
%>
</body>
</html>
```

Note the PHP line terminator is a semicolon.

PHP SCRIPT TO DISPLAY A WEB P

```
<?PHP
Print "<html><head>";
Print "<TITLE>Sample web page</TITLE>";
Print "</HEAD>";
Print "<BODY>;
Print "<H1>Here is my Heading</H1>";
print "<P>This is my page content</P>";
print "</body></html>";

PHP?>
```

GOTCHA'S

Every language and scripting environment has its own syntactical oddities. Here are some common "gotchas' in each language. Of course, being "gotchas" you will only know they have "got you" when the "get you"!

- PHP: variable names are *case sensitive* – that is $userID is *not* the same as $userid
- PHP: command lines are terminated by a semicolon (";")
- PHP: if you are using the session object to keep session state, the first line in each script should be **session_start().** The gotcha?
    - Session_start (no brackets) is *not* the same as session_start**()**
    - ie you **must** use the function brackets – in this case you will not get an error but your sessions won't work
- PYTHON: the logic of flow control is determined by indenting of the if/while statements (see below for more details)
- ASP: string handling is very resource intensive in ASP and can impact the scalability of your application
- PHP & ASP: the intrinsic functions for Session handling are bound to a specific web server: if your website usage grows dramatically this can affect the scalability of your application.
- Debugging and IE browsers – you may not see your script error unless you turn of "friendly errors" in IE browser configuration (Tools/Internet Options/Advanced tab/uncheck Show friendly HTTP errors)!!!
- General scripting: beware spaces at the end of fields and variables when doing comparisons: "Balnaves " does not equal "Balnaves". Use the Trim functions to strip such characters
- Beware case when doing comparisons and searching – be conscious of the environment in which you are searching and use case insensitive search expressions and comparisons.
- Unix: page file names are case sensitive. If you are getting 404 "page not found" errors, is it because your page is "MyPage.php" not "mypage.php"?
- All programming languages have their syntactical sensitivities. You have to work with them.

  For instance, PHP and Python differentiate dictionaries from function calls by using square and round brackets. $_GET might appear to behave like a function but it is a predefined dictionary, similarly $_SESSION, so:
    - $_SESSION["loggedon"] *not* $_SESSION["loggedon"]

  But PHP functions are called with round brackets *eg:*

  location("http://www.prosentient.com.au")

COMMENTS

Comments within scripts and programmes are an essential device for documenting the logical and function of your scripts. In both Python and ASP comments can be placed on a line to themselves or at the end of a line of code:

ASP COMMENT: prefix your line with a single quote (')

Eg

```
' this is the main control loop
```

PYTHON COMMENT: prefix your line with a hash symbol (#)

Eg

```
#this is the main control loop
```

PHP COMMENT

// - single line prefix to a comment

Eg //this is a comment

/* */ - these surround a multi-line comment block

Eg:
```
/*
        Comment lines
        Comment lines
*/
```

VARIABLES

All programming languages have some means to store, manipulate and act on complex data according to a set of rules. Persistent data is generally derived from file systems or databases. Transient data used to calculate, translate or transiently manipulate content is generally held in *variables*. In both ASP and Python, *variables* can be created without pre-definition. However, care should be taken at all times:

- To name variables in a descriptive manner – there are no effective limits on the length of your variable names, so they should be named in a manner which facilitates the understanding of your script
- To use the variable exactly in the same spelling throughout. Because ASP and Python can create variables on the fly, a miss-spelt variable will not necessarily generate any error.

Examples of ASP and PHP variables are:

- Name = "Mr Bloggs"
- Address = "98 Erewhon St"
- Children = 6
- FinancialStatus = "D"
- MailingLabel = Name + /n + Address (Python)
- MailingLabel = Name & chr(13) & chr(10) & Address (ASP)

Examples of PHP variables are:

```
$income = 59000;
$surname = "Bananos";
$fullname = $surname . $firstname;  (php concatenation)
```

Notice some variables are numeric – they are not contained in quotes. Some variables are "text" – and are indicated as such by the fact that they are surrounded by quotations. A useful Python construct illustrated in the example above is the assignment of long text variables entered through a useful triple-quoting convention:

```
MyPage = """<html><head><Title>My Title</Title>
          </head><body>
          This is my web page
          </body></html>"""
```

Some special characters can be used in string variables:

\n – In Python this indicates that a new line should be output (the "carriage return, line feed"). Note that this is *not* an HTML <br> tag. It is, however, useful in formatting your HTML code visually attractively.

chr(13) & chr(10) – to create a new line in ASP.

These are called "variables" for the simple reason that their contents may change: For instance, the following will adjust the variable Children by increasing its value by 1:

```
Children = Children + 1  (ASP)
$children = $children + 1;  (PHP)
```

Beware of case sensitivity:

```
$children = $Children + 1;
```

is different in PHP from:

```
$children = $children + 1;
```

But the difference will cause no syntax errors in your code. In ASP you can force explicit declaration of variables by including the line:

```
Option explicit
```

as the first line of your code.

FLOW CONTROL AND PROGRAM STRUCTURE

Flow control is the essential element in scripting that allows your script to do more than static output of content. Flow control allows you to implement a logical set of selection rules, sequence of operations and repetition of operations that allow the complex implementation of an application or service. Scripting flow control allows you to perform repeating actions (for instance, printing every item in a list) and allows you to conditionally determine actions that may be based on the values stored in variables. Scripting languages such as Python enforce indentation to visually enhance the understanding of flow control – it also uses this indentation to enable it to interpret your program structure. The indentation of a script is an essential courtesy of the programmer to facilitate correct interpretation of your script. While PHP and ASP do not require indentation, but the same conventions for indentation should always be applied for easy of script interpretation.

The **if** statement is the simplest flow control statement, and follows a natural logical test structure – based on the logical outcome of a condition (true or false, yes or no) perform one set of actions or another:

PYTHON	**ASP**	**PHP**
If *condition:*	**If** *condition* **Then**	**if (*condition){***
Action 1	*Action 1*	*Action 1*
Action 2	*Action 2*	*Action 2*
Action 3	*Action 3*	*Action 3}*
Else:	**Else**	**else {**
Action 4	*Action 4*	*Action 4*
Action 5	*Action 5*	*Action 5*
Action 6	*Action 6*	*Action 6*
	End If	**}**

In the above examples the "actions 1,2,3" will happen only if the condition is true, otherwise the actions 4,5,6 will occur. The following **if** test illustrates the dangers of incorrect indentation in Python. We may wish to set a variable **AllowAccess** dependant on the value of another variable **UserType**.

```
If UserType == 'S':
        AccessMessage = 'Welcome'
        AllowAccess = 1
Else:
        AccessMessage = 'Get Lost'
AllowAccess = 0
```

In Python, the result of this **if** test will always be "AllowAccess = 0". It should have been coded as follows to achieve the desired result:

```
If UserType == 'S':
        AccessMessage = 'Welcome'
        AllowAccess = 1
Else:
```

```
AccessMessage = 'Get Lost'
AllowAccess = 0
```

The **if** test has a range of logical operators which are thoroughly described in the Python Tutorial and Reference manuals on from the Python website. Some common logical operators are:

PYTHON	**ASP**	**PHP**
= = means "equals"	== means "equals"	== means "equals"
>= means "greater or equal to"	>= means "greater or equal to"	>= means "greater or equal to"
<= means "less than or equal to"	<= means "less than or equal to"	<= means "less than or equal to"
!= means "not equal to"	<> means "not equal to"	!= means "not equal to"
& = and	And means "and"	And means "and"
| = or	Or means "or"	Or means "or"

Testing string equality has other considerations. When we compare a variable against a fixed string, we need to be conscious of the *case* of the string we are comparing – for example in Python:

```
If surname ==    'Balnaves':
```

would *only* match on the exact string 'Balnaves'- not 'balnaves' or 'BALNAVES' or indeed 'balNaves'. String handling expressions are available to assist in string handling and comparisons. The python **string** library has manu useful functions for string searching and comparison. The above example, might have been handled:

Python	**ASP**
import string if string.lower(surname) = = 'balnaves'	if lcase(surname) = "Balnaves" then

Additional spaces at the end of a variable may also affect such a comparison. 'Balnaves ' is not the same as 'Balnaves'. The above example could be adjusted to remove padding prior to the comparison:

Python

```
Import string
If string.rstrip(string.lower(surname)) ==   'balnaves'
```

ASP

```
If Trim(lcase(surname)) =   'balnaves' then
```

PHP

```
If (trim(strtolower($surname)) == 'balnaves') {action}
```

The following **if** logic might be used to check a variable called **strCountryName** for a particular value and set a value according to the result:

Python	ASP	PHP

If strCountryName == "Australia":
StrCurrency = "AUD"
If strCountryName = "England"
StrCurrency = "PND"
If strCountryName= "US"
StrCurrency = "USD"
	If strCountryName = "Australia" Then
	StrCurrency = "AUD"
	End If
	If strCountryName = "England" Then
	StrCurrency = "PND"
	End If
	If strCountryName= "US" Then
	StrCurrency = "USD"
	End If
		If {strCountryName = "Australia" }
		$strCurrency = "AUD";
		End If
		If {strCountryName = "England" }Then
		$strCurrency = "PND";
		End If
		If {strCountryName = "US"} Then
		$strCurrency = "USD";
		End If

How clumsy however would the above example be if there were 200 countries to deal with? There is a variant of the **if** statement in all these languages called the "select" or "case" statement that handles such a scenario well. We can improve our logic above in ASP and PHP (but alas not python) as follows:

PYTHON	ASP	PYTHON

☹	Select case strCountryName
	Case "Australia"
	StrCurrency = "AUD"
	Case "England"
	StrCurrency = "PND"
	Case "US"
	StrCurrency = "USD"
	End Select
		Switch ($countryname) {
		Case "Australia" :
		$strCurrency = "AUD";
		Case = "England":
		$strCurrency = "PND";
		Case "US":
		$strCurrency = "USD";
		}

Looping

Dealing with data file contents and an unknown number of rows in a database table requires an additional capability: to repeat the same scripting action a number of times. For example, if we have a database table called "users" which has an indefinite number of rows of data representing users of a web service. To list these users we cannot simply repeat a print statement for each user. The following example might be used to print three users in the table. The problem with the following ASP statements are evident – we simply don't know how many users there are in the table and the coding would in any case be wasteful to the extreme. They would fail with an error if less than three rows were in the table:

```
response.write rsUsers ("username") & "<br>"
rsusers.MoveNext
response.write rsUsers ("username") & "<br>"
```

☹

```
rsusers.MoveNext
response.write rsUsers("username") & "<br>"
rsUsers.MoveNext
```

A generic looping structure would display as many rows as are in the "users" table without this code repetition:

```
Do while not rsusers.eof
        response.write rsUsers ("username") & "<br>"
        rsUsers.movenext
Loop
```

Loops are useful to display each entry or selective entries in a database (for example to generate a report). Some common loop structures are:

	Python	**ASP**	**PHP**

The while condition performs a logical test (will I do this) *before* any action is taken. While the condition is true the statements are repeated. This is useful for stepping through a database table of information.	while condition:
	  statements
		While(condition)
		  statements
		loop
			while (condition) {
			  statements
			  }

The "do" condition always performs the set of statements at least once. Thereafter the condition is tested and the action repeated for as long as the condition is true	☹	do
		  statements
		loop while (condition)
			Do {
			  Statements}
			While (condition)
			
			}

The for statement is useful for a fixed number of iterations ("do this 10 times")	for var in sequence:
	  statements
		for var = 1 to 10
		  statements
		next
			for($i = 1; $i < 11; $i++)
			{
			statements;
			}

The **for** statement is generally used to repeat actions on a fixed list of items. The following example lists every author with a first letter *greater than* the letter **G.**

Python

```
#!/usr/local/bin/python
#sample4.py
import sys
import string
authorlist = ['deblanz','delany','gordon,h','wormald,g',
'albanathy,r','goward,g']
authorlist.sort()
for strAuthor in authorlist:
      if string.upper(strAuthor) > 'G':
         print strAuthor
```

ASP

```
set authorlist = server.createobject("Dictionary")
authorlist('deblanz') = 1
authorlist(,'delany') = 1
authorlist('gordon,h') = 1
authorlist('wormald,g') = 1
authorlist('albanathy,r') = 1
authorlist('goward,g') = 1
for each name in authorlist
    if ucase(name) > "G"
        print name
    end if
next
```

Note: Python list handling functions are considerably more powerful than ASP – the sort function is only one example of many integral features for advanced list handling.

The following example yields a simple list 0,1,2,3,4,5,6,7,8,9,

Python	**ASP**	**PHP**

for I in xrange(10):
print I
	for I = 1 to 10
	print I
	Next
		for($i = 1; $i < 11; $i++)
		{
		Print $i;
		}

The **while** statement will perform the set of statements indented underneath it as long as the condition remains true (that is, potentially indefinitely). The evaluation of the condition is undertake *prior* to each repetition of the statements. The **break** (in Python) and the **exit do** statement (in ASP) can be used to exit out of a loop abruptly .

TEXT MANIPULATION AND STRINGS

Scripting languages must deal effectively with text content of a variety of different forms. Scripting languages generally come equipped with a good complement of built-in text handling functions called "string" functions. A "string" is a text storage variable. Python has extensive string handling features in the **string** module. It also has sophisticated list handling facilities making manipulation of lists of items easy and flexible. These functions are similar in most languages.

The first challenge of string handling is *string assignment.* That is, storing (*assigning*) a set of text characters in a script *variable*. This variable can then be manipulated, changed and search. For instance, to assign the text string "Beethoven" to the php variable $strauthor do the following:

```
$strauthor = "Beethoven"
```

This variable can then be used to modify the original text string – for example by "concatenating" more caracters – eg:

```
$strauthor = $strauthor . ",L"
```

The result of this action is: "Beethoven, L"

In Python simple string concatenation can be performed using the '+' operator. In ASP the & operator is used. In PHP it is the "." character.

Python	**ASP**	**PHP**
StrAuthor = 'Beethoven' strauthor = strauthor + ', L'	StrAuthor = "Beethoven" strauthor = strauthor & ", L"	$strauthor="Beethoven" $strauthor = $strauthor **.** ",L";

Sub strings extraction is a simple and powerful text manipulation feature. In Python, text can indicated with square brackets and a *from* and *to* reference. The referencing in Python is relative to the first character, which is in position 0, and negative indexing means "from the end". Examples are:

```
Print strauthor[:4]    up to 4th index
Print strauthor[0:4]  from index 0 to index 3
Print strauthor[:-5]  ie 5 from the end
        All produce Beet
Print strauthor[len(strauthor)-4:]
Print strauthor[-4:]
        Produces oven
```

Python

In ASP the **mid**, **left** and **right** functions can be used for the same effect. **Note however that the character referencing starts from position 1:**

ASP

```
Response.write left(strauthor,4)    up to 4th index
Response.write mid(strauthor,1,4)  from index 1 to index 4
        All produce Beet
Response.write  right(strauthor,len(strauthor)-4)
Response.write  mid(strauthor,len(strauthor) – 4,4)
        Produces oven
```

In Python, you can repeat character strings easily: **Print " "*20** will generate 20 spaces. Similarly Print " "*20 will produce 20 HTML non-breaking spaces.

Other examples of string referencing are:

PYTHON	ASP	Result	Notes
"hope"[0]	mid("hello",1,1)	"h"	Indexing of a single character by absolute position
"hope"[1:3]	mid("hello",2,3)	"ope"	Slicing a range of characters
len("hope")	len("hope")	4	String length
"hope"[-1]	right("hope",1)	"e"	Relative from the end of the string
"hope" < "love"	"hope" < "love"	1	String comparison
"e" in "hello"	n/a	1	String search

Useful String manipulation functions include:

String manipulation	**Python** Refer to the string module of Python for more functions	**ASP** ASP inherits a rich vocabulary for string handling from BASIC	**PHP** Refer to the String Functions section of the PHP online manual for a rich set of string handling functions.
convert to lowercase convert to uppercase	string.upper(string) string.lower(string)	ucase(string) lcase(string	strtouppper(string) strtolower(string)
length of a string	string.len(string)	len(string)	strlen(string)
strip trailing spaces or all spaces	string.rstring(string) - string.strip(string)	rtrim(string) ltrim(string) trim(string	rtrim(string) ltrim(string) trim(string)
locate the index position of substring in string. *Remember, Python starts at index 0, ASP starts at index 1.*	string.find(string,substring,start,end)	instr(start,string,substring)	strops(string,substring)
replace all occurrences of the string *old* with the string *new* in the source string *string.*	string.replace(string,old,new)	replace(string,old,new)	str_replace(string, old, new) – *case sensitive* str_ireplace(string,old,new) *case insensitive*

These functions have nearly identical in Javascript.

DICTIONARIES AND LISTS

Text and string handling are pervasive in the Arts and Humanities. String handling functions are only the beginning of the useful tools at your hand in modern scripting languages. A tool in your scripting kitbag not to be missed is the Dictionary.

Python Lists and Dictionaries are collections of variables in a sequence. Python has very rich features for list and dictionary handling. In python a **list** is a simple sequence of variables in their natural order which creates a set of items that can be referenced by index number, for example:

```
authorlist = ['deblanz','delany','gordon,h','wormald,g', 'albanathy,r','goward,g']
```

Once again in Python the index starts from zero and the lists can be referenced to extract values from the list just as individual substrings can be extracted. For instance:

```
Print authorlist[0]
        Produces deblanz
```

Handy list functions that can be applied to the above list include:

Authorlist.sort() - sorts the list
Authorlist.append(value) - appends to the end of the list
Authorlist.count() - returns the total number of values in the list
Authorlist.remove(x) - removes the xth item from the list
Authorlist.reverse() - reverses the list order

A simple function that can create a list automatically is the String Split function:

Split(string, separator) - returns a list of words separated by separator

The following three sample scripts demonstrate aspects of list handling features in Python.

Python Sample script: printing a list in natural order

```
#!/usr/local/bin/python
import sys
import string
authorlist = ['deblanz','delany','gordon,h','wormald,g',
'albanathy,r','goward,g']
for strAuthor in authorlist:
        if string.upper(strAuthor) > 'G':
            print strAuthor
```

Output:

```
gordon,h
wormald,g
goward,g
```

Python Sample script: printing a list in sorted order

```
#!/usr/local/bin/python
#sample4.py
import sys
import string
authorlist = ['deblanz','delany','gordon,h','wormald,g',
'albanathy,r','goward,g']
authorlist.sort()
for strAuthor in authorlist:
        if string.upper(strAuthor) > 'G':
            print strAuthor
```

Output

```
gordon,h
goward,g
wormald,g
```

Notes

The above simple sort does not address issues of case sensitivity of the entries or leading spaces in items in the list.

Python Sample script: Generating an HTML Select list

```
print "<select name=" + chr(34) + "Country" + chr(34) + ">"
for strCountry in Countrylist:
   print "<option value=" + chr(34)
   print strCountry
   print chr(34)
   print ">"
   print strCountry
Print "</select>"
```

Python Dictionaries create a useful name-value pair, and referencing items in the list can thereafter by the name of the item as well as its relative position in the list. Such Name/Value pairs appears in many contexts:

- web sites use them for CGI interaction,
- scripting languages use them for enhanced list manipulation

ASP implements simple dictionaries using its Scripting.Dictionary class illustrated above.

Dictionary Variables are transient name/value pairs which can be valuable for translation of values and enhancement of the use of your code:

Use Dictionaries to map month number to month name
Python Sample Script

```
#!/usr/local/bin/python
MonthArray = {'Jan':1,'Feb':2,'Mar':3,'Apr':4,'May':5,'Jun':6,
'Jul':7,'Aug':8,'Sep':9,'Oct':10,'Nov':11,'Dec':12}
# Have a look at the contents
print MonthArray.keys()
print MonthArray.values()
# Notice now you can reference by the dictionary name to get a value:
print MonthArray['Feb']
print '\n'
#
# You can still step through the list, but notice you can reference by the
# name of the variable rather than just an index number in the list
#
for strMonth in MonthArray.keys():
      print strMonth
```

ASP Sample Script

```
Set MonthArray = Server.CreateObject("Dictionary")
MonthArray("Jan") = 1
MonthArray("Feb") = 2
MonthArray("Mar") = 3
MonthArray("Apr")= 4,
MonthArray("May")= 5
MonthArray("Jun")= 6
MonthArray("Jul")= 7
MonthArray("Aug")= 8
MonthArray("Sep")= 9
MonthArray("Oct")= 10
MonthArray("Nov")= 11
```

```
MonthArray("Dec")= 12
# Have a look at the contents
response.write MonthArray
# Notice now you can reference by the dictionary name to get a value:
response.write MonthArray("Feb")
response.write  chr(13) & chr(10)
# You can still step through the list, but notice you can reference by the
# name of the variable rather than just an index number in the list
for each strMonth in MonthArray
        print strMonth
```

PHP Sample

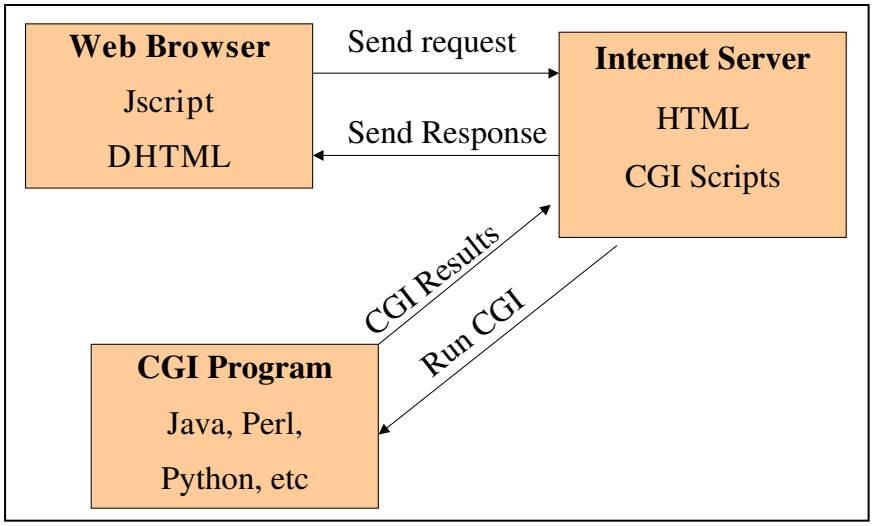
```
//PHP is terse but elegent:

$montharray = array("Jan" => "1", "Feb"=>"2", "Mar" => 3, "Apr"=>4, "May"=>5 ….)
print $montharray["Feb"];
```

PYTHON AS A CGI LANGUAGE

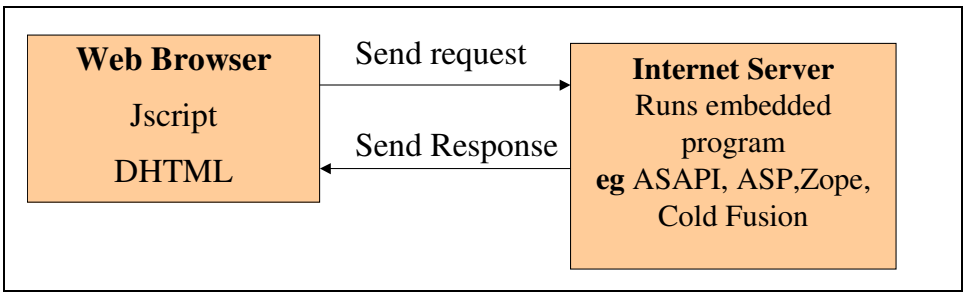
To add dynamic elements to a website, a scripting language of some kind must be deployed. There are several methods for implementing dynamic web content:

> CGI – Common Gateway Interface. The web server calls an external application passing and receiving values using a standard interface.



The CGI model will support any language that the host server can execute as long as that language can interchange information through the CGI interface. The application server model is generally more limited in the range of scripting languages supported. It does, however, have efficiency advantages: generally the Application Server script and its runtime environment will be "cached" or retained in memory – possibly only once for each client user. A CGI application may need to be fully and separately loaded for each client user. Failure of a CGI process will only generally affect the single user who experiences the failure.

ASP AS AN APPLICATION SERVER LANGUAGE



In ASP scripting on Microsoft's web server IIS, the web server has integrated server-based scripting that can be embedded within a given web page. Failure in an

Application Server application may result in instability or failure for other users if embedded as a tightly bound web server process..

In both models of operation, the conventional method for information interchange between pages is the HTML <FORM> **get** and **post** actions, and the addition values to the URL query string. Both essentially pass name-value pairs to your application and rely on your application generating the relevant HTML code in response.

QUERY STRINGS (URL MUNGING) AND FORM VALUES

URL query strings are one of the simplest methods for information interchange on the website. The Query String is the set of name value pairs that can be seen after the name of the url itself. Examine for instance the following URL:

http://www.prosentient.com.au/default.htm**?Product=CMS&Source=Australia**

This is otherwise known as "URL Munging" [Whitehead, 1998 #588]. This URL has two name-value pairs:

NAME	VALUE
Product	CMS
Source	Australia

The syntax of the name-value pair is also quite simple:

URL?name=value&name=value.

Special characters used in the value part of the URL must be *URL-encoded* – or translated to a format which allows your CGI application to differentiate between the use of special characters meaningful to URL – such as "/", "=", "&", and "?" which you can see in the URL above, and the content values you are sending. A similar encoding is applied to HTML form values - this is called *HTML encoding*. Fortunately both Python and ASP provide functions for the encoding and decoding of the values from URL data and HTML form data.

Python encoding and decoding of URL and form data

The **cgi** module has functions which make it easy to extract values that have been passed to a CGI application. The cgi.FieldStorage will return name-value pairs that have been received.

Python also has an elegant function for building a URL which is properly URL-encoded: urllib.urlencode.

The **urllib** module also has features for calling and returning the contents of a web page. These functions can be useful when building applications that need to extract content from other website. The following functions are particularly useful in retrieving website content:

GETURL(url) - returns the REAL address of a page (after redirect)

URLOPEN(url) - returns the content of the url as a FILE object
URLRETRIEVE(url,data) - retrieves the nominated url to the location specified in data

The following exercise illustrates the use of Python CGI functions to retrieve the first page from a web site search calling the Google search engine:

Sample script: retrieve search results from a Google search

```
#!/usr/local/bin/pythonimport cgi
import string
import sys
import urllib
# Note, the results are returned as a file variable.
print "Content-type: text/html"
print
sys.stderr = sys.stdout

strURL = 'http://www.google.com/search?q=' + \
  urllib.quote_plus('XML')
fileResult = urllib.urlopen(strURL)
print fileResult.read()
```

ASP encoding and decoding of URL and form data

The **Request** object of ASP provides an elegant interface to URL and HTML data objects:

Request.QueryString("QueryName") - will return the contents of the query string already parsed
Request.Form("FormName") - will similarly return the contents of an HTML form field already parsed.
Server.URLEncode("QueryValue") - will encode a specific value in URL-encoded form.
Server.HTMLEncode("FormValue") - will HTMLEncode the contents of the string.

PHP URL handling

The $_GET["queryname"] dictionary serves the purpose of URL fetching and decoding a particular query string name/value pair. Similarly $_POST["htmlformvariablename"] contains a specific variable. These are dictionary variables, not functions. The **for** function can be used to loop through the dictionary array in PHP as follows:

```
foreach ( $_POST as $key => $value ) {
        print $key . " " . "=" . " " . $value;
        print "<BR/>";
}
```

The urlencode("QueryValue") will encode a specific value in URL-encoded form

Eg – to create a url encoded link in your web page:

```
print '<a href="myprog.php?name=', urlencode($username), '">';
```

PERISTENCE AND COOKIES

A web server, using standard HTTP protocol, is a state-less engine. That is, it does not implicitly recognise the difference between any two web interactions, and cannot recognise whether the actions are the result of a the activities of a single user or sourced from a range of users. In an interactive, database-driven environment in which you need to manage the information delivered to users this is clearly a problem. This issue of "session persistence" is addressed in HTTP in either of two ways:

1. Use of Cookies, or name-value pairs that are always sent to the server by the client side browser with each interaction they have with your site.
2. Use of Query Strings, where *every* internal URL link contains information as to the session state of the user.

Cookies are a robust method for retaining session state. Cookies can be either transient (they last as long as single browser session is connected to your website) or persistent (they are stored locally on your system and are re-stored when you next visit this same website). They can thus persist on the users system so that when the user returns they are recognised as the same user. A difficulty in use of Cookies is the ability of users to turn off Cookie support or selective accept/reject cookies. Generally, however, users are now happy to accept a minimal signature when they visit a site, especially session cookies.

Query Strings are more robust in a cross-browser sense than Cookies, and cannot be "turned off" by the user. However, if a user navigates outside your site and later returns without the appended query string, the effect will be to lose the session state for that user. It is therefore essential to have tight and disciplined navigational control when using Query Strings to track information on a visitor to a website.

Cookies in Python are implemented using the **Cookie** module, or can be parsed manually. ASP offers cookies in two forms – session-based values using a system generated cookie value, or the response.cookie function for creation of your own cookies. PHP has the "setcookie" function to create permanent cookies and session_start() with $_SESSION to simplify state-based management similar to the ASP session (see a PHP example of session handling at the end of this chapter).

Sample script creating a persistent cookie called "user"

Python Scripting Sample

```
#!/usr/local/bin/python
import os, cgi, Cookie
print
print "<html><body>"
c = Cookie.Cookie()
try:
   c.load(os.environ["HTTP_COOKIE"])
except KeyError:
    pass
form = cgi.FieldStorage()
try:
     user = form["user"].value
except KeyError:
    try:
       user = c["user"].value
     except KeyError:
        user = "nobody"
c["user"] = user
print """
<form action="/cgi-bin/test.py" method="POST">
<input type="text" name="user" value="%s">
</form>
"""  % cgi.escape(user)
print """</body></html>"""
```

ASP Scripting Sample

```
Response.write "<html><body>"
user = request.form("user")
if user = "" then
     user = Trim(Request.Cookies("user"))
end if
response.write " <form action=""testlogin.asp"" method=""POST""> "
response.write "<input type=""text"" name=""user"" value=""<% =user %>"">"
response.write "</form> "
response.write "</body></html>"
```

PHP:

```
Print  "<html><body>";
$user = $_POST ["user"]
if ($user = "") {
     $user = $_COOKIE["user"];
}
print ' <form action="testlogin.asp" method="POST"> ';
print '<input type="text" name="user" value=" .$user .">";
print "</form> ";
print "</body></html>";
```

First try to retrieve the "user" value from a previous for posting

If not form value for user exists, fetch the value from the cookie

FILES AND DATABASES

The content source that scripting draws on may reside in operating system files or within a database. The file structure of the Operating system remains a popular vehicle for the management of multimedia content, and the storage of individual multimedia content objects on the file system has the advantages of leveraging the inherent capabilities of most current operating systems in indexing and managing files, as well as simplifying the delivery of these files to authoring and publishing systems.

A key difficulty of retaining multimedia content on the file system is the management of meta-data regarding that file. Crucial meta information regarding text could include title, version, copyright, authorship, and distribution details.

Hybrid systems which store meta-referential information in a database framework, the original multimedia content on the operating system file architecture are still very common. However, databases which attempt to organise all aspects of multimedia content are becoming prevalent.

Scripting languages can generally access both content in the file system framework, and content which is accessible through standard database interface architectures (JDBC, ODBC and many others).

Files have to be explicitly **opened**, then read from or written to, then **closed**.

OPENING FILES

Within Python, access to operating system files is achieved through the **sys** module, which contains environment specific function handling for a standard set of file operation. Like most scripting languages, Python requires that a file be explicitly opened before it can be accessed:

File = open(file name, file mode)

In the above example, file mode values can be:

- 'r' is used for "read" access – that is, data on the file cannot be changed
- 'w' for write access – that is, information on the file can be read by the script and also modified
- 'a' append – that is, information can be appended to the end of the current file.

For example:

1. File = open('config.txt','r') - allows the user to read the content of the file 'config.txt' . This file must already exist!

2. File = open('onthefly.html','w' – allows the user to CREATE/OVERWRITE information on the file 'onthefly.html'. This

file will be overwritten if it exists. The file will be created if it does not exist
3. open('userhistory.log','a') – is useful for creating log files. If the file does not already exist it will be created, otherwise the user is able to append information to the end of any existing content.

In ASP, the FileScriptingObject provides access to the file system of the relevant Windows system. Similar to the Pythons **sys** module, this class supports three modes for accessing OS file content access: read, create and append. The following line will open a file in the current application subdirectory:

ASP: Opening a file for Read access

```
Const ForReading = 1
Set fs = CreateObject("Scripting.FileSystemObject")
Set fsFile = fs.OpenTextFile(App.Path & _
       "\myfile.htm",ForReading)
```

ASP: Opening a file ready to append to the end of existing content (and create the file if it does not exist)

```
Const ForAppending = 8
Set fs = CreateObject("Scripting.FileSystemObject")
Set fsFile = fs.OpenTextFile(App.Path & _
       "\myfile.htm",ForAppending,True)
```

ASP: Creating a new file/overwriting an existing file

```
Const ForAppending = 8
Set fs = CreateObject("Scripting.FileSystemObject")
Set fsFile = fs.CreateTextFile(App.Path & _
       "\myfile.htm")
```

In PHP we have the **fopen** function:

```
$fp = fopen("mylocalfile.txt", "r");
```

PHP offers 6 file modes:

'r' - Open for reading only
'r+' - Open for reading and writing
'w' - Open for writing only automatically creating the file if it does not exist
'w+' – As "w" but empty the file if it already has data
'a' - Open for writing only; place the file pointer at the end of the file. Handy for log files!!!!.
'a+' - Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.

READING/WRITING DATA TO FILES

Having opened the file, access to the content of that file is trivial. A set of functions are provided which allow the scripting programmer to read information from the file a line at a time (for text data) or as a single file read (for multimedia content of other sorts).

Python:

Filedata = file.read() - reads the entire content of the file into a file variable
Filedata = file.readline() - reads one line at a time (up to the line break)

ASP:

AtEndOfStream – have I reached the end of file?
FileData = fsFile.ReadAll - reads the entire content of the file into a file variable
FileData = fsFile.ReadLine - reads one line at a time (up to the line break)

PHP:

Feof – have I reached the end of file?
Fgets – read a line at a time
Fread – read a chunck of data or the whole file at once

```
// fgets example
$fp = fopen("myconfigfile.txt", "r");
$configline = "";
while(!feof($fp))
{
$ configline.= fgets($fp, 4096);
if {$configline == "accesslevel" )
                        {$accesslevel = $configline; }
}

// fread example

<?php

$fp = fopen("test.txt", "r");
$data = fread($fp, filesize($fp));

echo $data;

...
?>
```

The following is an simple file loop to read and display to screen every line in a file – assuming the content is standard text-based content with lines delineated by a "Carriage Return, Line Feed".

Sample: Show every line of a file as a web page

Python

```
#!/usr/local/bin/python
import sys
print "Content-type: text/html"
print
print "<html><body>"
file=open('sample6.dat','r')
data = file.readline()
print "
while data:
        # remove the newline char
        if data[-1] == '\n':  data = data[:-1]
        print "'%s'" % data
        data = file.readline()
file.close
print"</body></html>"
```

ASP

```
Const ForReading = 1
Set fs = CreateObject("Scripting.FileSystemObject")
Set fsFile = fs.OpenTextFile(App.Path & _
 "\sample6.dat",ForReading,True)
Response.write "<html><body>"
Do While Not fs.AtEndOfStream
      strText = fs.ReadLine
      Response.Write fs.ReadLine & "<br>"
Loop
Response.write "</body></html>"
```

Note

The above example would allow the parsing and modification of text a line at a time. If no modification of the file contents is required, a Read() and ReadAll() would be more efficient.

Additional handy file functions in the Python **os** module are:

os.IsFile(filename) - returns true if the file exists
os.IsDir(path) - returns true for a directory
os.listdir(path) - returns a list of the files in a path

Similarly, the FileSystemObject in ASP supports the following methods:

fs.CreateFolder(*path)*
fs.FolderExists(*path)*
fs.FileExists(*path)*
fs.GetFileName(*path)* - returns a file at a time from the nominated path, for example

```
strFileName = fs.GetFileName("d:\inetpub\wwwroot\*")
```

```
Do While strFileName <> ""
        Response.write strFileName & "<br>"
        strFileName = fs.GetFileName("d:\inetpub\wwwroot\*")
Loop
```

DATABASES

To begin with, here is some common industry terminology associated with database handling:

RELATIONAL DATABASE – a relational database stores information permanently in TABLES of related data.

TABLES (in common parlance a "file") contain data for a specific set of data – for example a simple bibliographic database might have the following tables:

USERS – a table of user logins and passwords
TITLES – a table of books available
USERLOANS – a table which has shows which USERS have which TITLES on loan (this table would therefore only need the keys for USERS and the KEYS for TITLES)

COLUMNS – are the "fields" of a TABLE. For instance the USERS table above might have the following COLUMNS:

UsernameID – the *primary key* to the file
Surname
FirstName
Password

ROWS – are the individual "records" of the TABLE – otherwise known as a TUPLE

Eg for our USERS table a single row might contain

111, "Balnaves", "Edmund", "reallygoodpassword"

Most commercial database packages are Relational Databases, and scripting languages all provide methods to access the data stored in Relational Databases. Most use a standardised query language called SQL. Scripting languages diverge considerably where we need to access a database, and even to some extend according to the type of database (Microsoft SQL Server, mysql, etc). Access to content on a database will depend on the use of database specific access functions, which can vary from database to database.

Nevertheless, database access functions will generally follow a similar logical access structure:

Connect to your database server (this usually requires a host name,database name, user and password)
Send an SQL command to the database
If the request is a SELECT to retrieve records, act on the recordset:
Do While not EndofFile
Act on the request results
Get the next entry
End do
Close your recordset
Close your database connection

The database request is usually constructed using SQL (Structured Query Language), a standardised syntactical method for interrogating databases. The common database access methods are:

PHP: MYSQL and POSTGRESQL are popular databases used with PHP. PHP has built-in connection functions for many database systems. The syntax for each of these varies. The examples in this chapter are for MYSQL. An optional module called "pear" can be enabled in PHP which provides a generic point of access to multiple databases. The advantage of using pear is that the same programming logic can be used to some degree independently of the underlying database. This abstraction works only so far however, as you will discover that the SQL syntax between database systems can also vary in subtle ways – for example Microsoft SQL has a LIKE operator for partial string searching, while POSTGRESQL as ILIKE:

Select * from MyTable where surname LIKE 'Smith%'
VS
Select * from MyTable where surname ILIKE 'Smith%'

ASP: Microsoft supplies the ADODB (Active Data Object, Database connect)object which provides an abstract interface to database object through ASP.

PYTHON: Specific modules are available for ODBC, JDBC and other database connectivity methods.

Given the variability of database usage not only by scripting language but by particular database used (Microsoft SQL Server, PostgreSQL, MySQL, Oracle, etc) you will need to seek out resources that describe the particular access syntax for each database method. An example using PHP and MYSQL appears in the script at the end of this chapter.

Functions

Once your scripting activities achieve a moderate level of complexity, the use of functions becomes a valuable timesaver. Functions are universal to most scripting languages, and are a means of bringing commonly called blocks of code together

which can be called by a name. This is especially useful where the same scripting code is used innumerably in your script. A further advantage of functions is the ability to organise the scripting in a manner that is easier to read and interpret by others looking at your scripts.

Scripting and programming languages have developed based on the cumulative contribution of innumerable programmers. The use of functions in the micro-level (that is, within one script), and at the broader level the use of object oriented programming, are focussed on the effective modularisation, componentisation and re-use of existing scripting work. Where the same code has been repeated throughout your code, every reference must be updated whenever a change to this coding logic is required. The introduction of these structures within code is known as structured programming: the organisation of coding into logical units that efficiently describe the logic of your script with the minimum of code duplication and the maximum of comprehensibility.

The format of a function in Python is simple:

```
def functionname:
    code to run
    return value
```

Once again, indentation plays a key role in understanding the contents of a function. The following illustrates the use of a function to avoid repetition of code that is used several times:

Sample Function Scripts

Python

```
Def function ShowHTMLHeader:
        print "Content-type: text/html"
        print
        print "<html><body>"
Def function ShowMainMenu:
        print  """<a href="option1.htm">Option 1</a><br>"""
        print """<a href="option2.htm">Option 1</a><br>"""
Def function ShowHTMLFooter:
        print "<br>&copy; Edmund Balnaves<br></body></html>"

/*  start of the main routine */
requestform = cgi.FieldStorage()
If form('NextPage') == 'Menu':
    ShowHTMLHeader
    ShowMainMenu
    ShowHTMLFooter
Else
    ShowHTMLHeader
    ShowSubMenu
    ShowHTMLFooter
```

ASP

```
Function ShowHTMLHeader:
        the code here  puts out the web page HTML headers
End Function
Function HTMLHeader
        Response.write "<html><body>"
End Function
Function ShowMainMenu:
        Response.write "<a href=""option1.htm"">Option 1</a><br>"
        Response.write "<a href=""option2.htm"">Option 1</a><br>"
End Function
Function ShowHTMLFooter:
        Response.Write "<br>&copy; Edmund Balnaves<br></body></html>"
End Function
/*  start of the main routine */
If request.form('NextPage') = "Menu" then
    ShowHTMLHeader
    ShowMainMenu
    ShowHTMLFooter
Else
    ShowHTMLHeader
    ShowSubMenu
    ShowHTMLFooter
End if
```

Notes:

The main routine is brief and calls functions to do most of the work. Note the re-use of ShowHTMLHeader to put out HTML header code which is same irrespective of the type of page tested for in the *if* test.

Python Function example: show the current date

Build a function "today()"

```
#!/usr/local/bin/python
#sample7.py
import sys
import string
import time
def today():
        timeseconds = time.time()
        timearray = time.gmtime(timeseconds)
        timetoday =  str(timearray [2])+ '/' + \
                str(timearray [1]) + '/' + \
                str(timearray [0])
        return timetoday
```

Note: this function may be used in many different contexts within a script in order to display today's date.

PUTTING IT ALL TOGETHER

The following PHP web pages give an annotated example of a session-based login system which:

- Displays a login form to the user
- Posts to a page which checks if the user and password are valid against the database
- If the user is invalid, redirects back to the login form
- If the user is valid, sets a session cookie that can be checked at the start of all other pages.

This illustrates all the major elements of web-based database connectivity.

INIT.PHP

```
<?PHP
// Create the test table and insert a test username/password
//
//  connect to the database
$connection = mysql_connect("localhost",
      "dbuser","whatsit");

mysql_select_db("my_system",$connection);
// let me know if the connection fails
  if (!$connection) {
    print("Connection Failed.");
    exit;
  }

// try to read the userid and password
$sql = " CREATE TABLE my_system.myusers (" .
           "  username VARCHAR(240) NOT NULL," .
           "    password VARCHAR(240) NOT NULL," .
           "    PRIMARY KEY(username)";

$myresult = mysql_query($sql, $connection);
print $sql;
$sql = "insert my_system.myusers (username,password) values
('test','mypassword'); ";
$myresult = mysql_query($sql, $connection);
print $sql;
?>
```

System thinking: what about users who have forgotten their password – may be need an email column.

MYLOGIN.PHP

```
<?PHP
//
//
// Show the login page (the script must generate all the relevant HTML)
// Make sure that we start cookies working on the client browser
// otherwise we can't keep session state!
session_start();
echo "<html><head><title>my login page</title></head>";
echo "<body><form action=mylogincheck.php method=post>";
// note the use of single quotes used because the html
// has embedded double quotes!!!!
echo 'Userid <input type="text" name="userid" value=""><br>';
echo 'Password <input type=password name="passwd" value=""><br>';
echo '<input type="submit" name="submit" value="login">';
echo '</form></body></html>';
?>
```

System thinking: what about users who have forgotten their password:
System thinking: what about navigation, graphical design?
System thinking: what about version history (ie change history to this script)
System thinking: what about a copyright statement in the script?

MYLOGINCHECK.PHP

```
<?PHP
session_start();

// notice that this script outputs no PHP it just does a simple logic check to see if the
// user and password supplied are found.
//
//  connect to the database


$connection = mysql_connect("localhost",
     "dbuser","whatsit");

$userid = $_POST["userid"];
$password = $_POST["passwd"];
$mydb = mysql_select_db("my_system",$connection)  or die (mysql_error());


// try to read the userid and password
$sql = "select * from myusers where username = '" .$userid . "' and password = '" .
$password . "'";

$result = mysql_query($sql, $connection)  or die (mysql_error());
$myrow = mysql_fetch_row($result);

// check for database error
if ($myrow =="") {
          // if not found (ie row is empty), redirect back to mylogin.php
          $_SESSION["loggedon"] = "";
          session_write_close();
          header("Location: mylogin.php");}
else {
          // if found, set session variable and redirect to "myhomepage.php"
          $_SESSION["loggedon"] = $userid;
          $_SESSION["on"] = "yes";
          session_write_close();
          header("Location: myhomepage.php" );
          exit;
}
?>
```

System thinking: this script does *is* case sensitive – is this good for the username
System thinking: this script does not trim spaces – un-necessary mismatches may occur

MYHOEMPAGE.PHP

```
<?PHP
// Here again we wish to show a full HTML page, but first we must check the
// Session variable to see if the user is logged in!  We need to do this at the top of
// every page.
session_start() ;
if (trim($_SESSION["loggedon"] ) == "") {
        header("Location: mylogin.php");

}
//
//
// Show the login page (note you must generate all the relevant HTML)
echo "<html><head><title>my home page (a little bit secure)</title></head>";
echo "This is session ".session_id() . "<br>";


echo "<body>here I am and here are all the users on the system:<br>";

//  connect to the database
$connection = mysql_connect("localhost",
     "dbuser","whatsit");

$userid = $_POST["userid"];
$password = $_POST["passwd"];
$mydb = mysql_select_db("my_system",$connection)  or die (mysql_error());



// let me know if the connection fails
  if (!$connection) {
    print("Connection Failed.");
    exit;
  }

// try to read the userid and password
$sql = "select * from myusers order by username";

$result = mysql_query($sql, $connection)  or die (mysql_error());
echo "<table><tr><th>user</th><th>passwd</th></tr>";
while ($myrow = mysql_fetch_row($result)) {

                printf("<tr><td>%s </td><td>%s</td></tr>\n",
                $myrow[0], $myrow[1]);

}
echo '</table></body></html>';
?>
```

System thinking: what about using an "include" file in PHP for the login check for each page rather than just repeating the code – haven't I told you how: Google it!
System thinking: what is the difference between echo and print in PHP – find out!

SUMMARY

This chapter has explored Scripting Languages and Python specifically as an example. Scripting languages allow the transformation of static content to dynamic information presentation. They are also essential tools for content transformation. Scripting languages are now embedded in many applications and provide a powerful framework for the enhancement of digital creations. Some examples of data manipulation and script organisation through functions are given.

QUESTIONS

1. Describe the differences between a Scripting language and a Programming language

2. What is the role of a *function* in scripting?

3. How is the *cgi* module useful in web-site scripting?

FURTHER READING

http://www.python.org - for Python software, scripting examples and documentation

http://www.php.net – for PHP programmers, PHP downloads samples and documentation

http://www.w3schools.com/php/ - an excellent online tutorial in PHP

http://www.w3schools.com/asp/ - an excellent online tutorial in ASP

LUTZ, M. (1996): *Programming Python*. California, O'Reilly.

http://aspn.activestate.com/ASPN/Python/ - cookbooks and code samples

http://python.sourceforge.net/devel-docs/index.html - A consolidated documentation resources for python, with many examples of scripts

…and when in doubt, there are very many websites with programming assistance for ASP, PHP and Python: "google it!"

Chapter 5: Content Delivery Systems

CONTENT SERVERS

The convergence of technology across different disciplines has yielded many vital changes over the last few decades. Few of them have been as profound as the convergence of Networking and Personal Computing technology. Both have their early beginnings in the 1960's. The technological innovation represented by TCP/IP as an immensely flexible networking framework was not realised until the birth of the Personal Computer in the 1980's.

The PC made available to the wider community technology that was largely reserved for business and government before that. This bought a blossoming of useful "category killer" applications such as spreadsheets, word processing and small database systems. Born was a whole generation of easily portable and accessible, and readily usable tools. Programming itself left the domain of specialists and became a community phenomenon. We see, however, a fundamental change when the early technology of Bulletin Board Systems accelerated the sharing of information technology tools and information itself.

Bulletin Boards could be said to have had their heyday in the 1970's and 1980's, with the emergence of the internet as an effective method of interconnectivity that reduced massively the level of technical understanding needed to exchange information.

In the early 1990's the internet browser emerged as the "multimedia" platform for content delivery over the internet. The simplicity of the hyperlink paradigm and the role of the browser as a standardised content delivery platform have had an unparalleled effect in spreading technology use into the community.

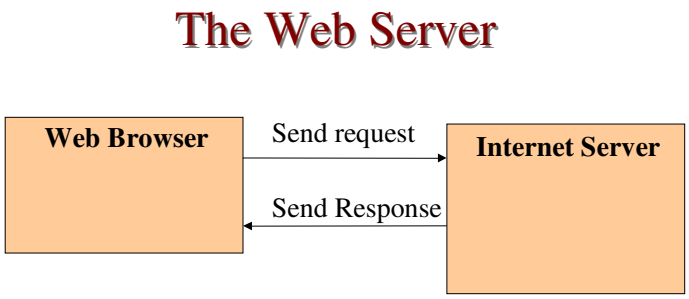
The World Wide Web now stands at a cross-roads as conflicting demands for free-flow and sharing of information come against commercial desires to know and understand clients and customers, and communal desires to limit the extent of discussion to social norms that are imposed on other media. In this context, we can see the resurgence of "Bulletin Board" style peer-to-peer architectures (in the manner of Napster and other systems) that defy efforts to bring community norms to play in an Internet framework.

The model for information delivery that was represented by the early browsers such as Mosaic has not profoundly changed, however much the utility and functionality of both the browser and server have changed dramatically.

The framework for information flow on the internet is HTTP, or HyperText Transmission Protocol. This technology is remarkably simple, and has formed a profoundly effective medium for static and dynamic information delivery. The same

platform has been extended to delivery XML and other markup approaches, as well as a wide variety of proprietary and standards-based multimedia objects.

Web applications are principally directed to sending (possibly dynamically generated) HTML documents to a user's Web browser while they interact with a Web site. CGI servers were an early approach to adding some dynamic content generation capability to websites. Internet-based Application Servers replace the traditional Web Server (which acts in many ways like a file server) with an application that may interact with a web browser, or with another client-based application (such as happens when the SOAP protocol is used). The Application Server might well act as an engine that processes data for another program to read and process on behalf of the user (for example, generating XML which is rendered by the client browser using XSL).



HTTP PROTOCOL

The protocol for communication between the browser and the server is called HTTP, or Hyper-Text Transmission Protocol. The client is responsible for initiating communication with the server, and it has a set of several basic commands, the most commonly used of which are:

GET url – retrieves the nominated web page from the server

HEAD url – retrieves the headers *only* from the server

POST url *followed by the contents of the post.*

These commands are sent in the format:

Command and Headers
Optional Content

Eg – a GET command:
GET home.html
Content-Type: text/html
Content-Length: 0

Eg – A POST command
POST search.html

&search=what+is+http+protocol&action=post

There is always a blank line between the Command/Headers and the Content. Both a GET and POST can contain additional content. For the GET command these are the query string parameters. For the POST command the content information contains the form field values being posted to the server. Other headers include cookies sent by the browser to the server.

Communication back to the client from the server follows a simple format of:

Server Headers
Server Content

Again, there is always a blank line between the Headers and the Content.

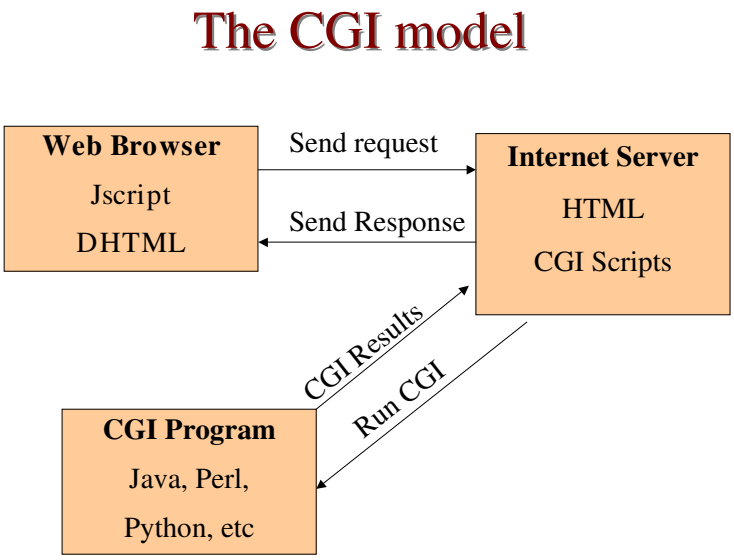
The headers are "hidden" meta-information about the nature of the content. All web servers deliver *at least* the most basic header line of the "**content-type"** which informs the client about the nature of the content being sent. Other headers indicated the currency and expiry of the page for purposes of caching (or to prevent caching), and cookies created by the server for the browser.

DYNAMIC WEB PAGES

The process of rendering information that is delivered to the client can be achieved through two approaches:

- Embedding scripting languages in the server itself (such as Microsoft's *Active Server Pages* and a range of other scripting approaches
- Calling scripting and application functions through a more "arms length" *CGI* program.

CGI or "Command Gateway Interface" is simply a standardised mechanism for calling functions from a server that execute program code - as distinct from simply delivery pre-formatted "static" HTML.



One of the functions that Web servers do *very well* is efficiently and quickly delivery static pages to end users. The embedding of dynamic content processing scripts and calls to external *cgi* functions are therefore, by their very nature, slower than the static delivery of web pages. This is an important consideration in the design of content delivery, because the slow delivery of content and easily impede the utility of the content server.

A popular language for the building of CGI functions has been Perl, a scripting language particularly powerful in its textual parsing capabilities. Perl is only one language that can be used for CGI scripting. A web server that supports CGI potentially gives a framework for calling any supported programming or scripting language. Popular CGI languages are Perl, Python and ASP (called as a CGI script rather than as a dynamic page).

CGI scripts can be slow, because most require a separate Operating System process to be invoked. Starting new processes on a Web server adds extra overhead. Soon after CGI came into being, other technologies quickly followed to solve this performance issue.

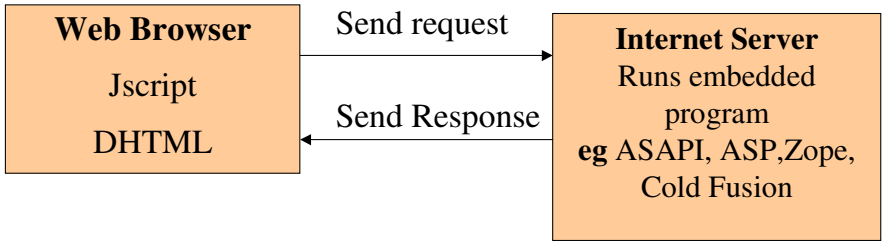
ASP (Microsoft Active Server Pages) technology represented a different approach to web server scripting. A VBScript interpreter was embedded in the Microsoft Internet Information Server runtime engine. This provided some improvements in efficiency at the risk of introducing vulnerabilities to the Web Server itself. Similar approaches have been taken with server-side Java implementations, such as Servlets and Java Server Pages, and more recently PHP (PHP HyperText Processor).

Even today, Perl has its dedicated developers, with new versions now highly optimised for web server efficiency.

This book will contrasts two scripting languages – Python and Microsoft's ASP. The use of *Python*, a programming and scripting language developed in the early 90's as an open source programming language. It has had considerable success and its capabilities are now very strong in web interaction. Microsoft ASP pages have a substantial following on Microsoft (and some Unix) web servers.

The application server with embedded scripting is the second approach to dynamic delivery of content.

The Application Server model



In the Application server model, we see the Web Server dynamically executing "programs" or "scripts" which process inbound requests and deliver responses. Most E-commerce applications follow and Application Server model, sometimes adding an additional "tier" of interaction as they move to separation of "business objects" from the database content server.

While it would be ideal to reach decisions on web content delivery independent of architectural considerations, these architectural issues can have profound implications on cost and the degree to which the systems can "scale" to ever larger numbers of users.

Content servers now include not only HTML web servers but also audio and video servers that can supply on-demand and streaming video. On-demand delivery integrates effectively with Multimedia reference sources: news, short replays of previously broadcast content, etc. Streaming video is the digital equivalent of traditional broadcasting and has the same advantages (efficiencies of scale, consistency of delivery, management) and the same disadvantages (it is a broadcast paradigm rather than an hypermedia paradigm of operation).

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 discovery and management. 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. This 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 area, 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.

Dublin Core Metadata Initiative (DCMI).

The Dublin Core Metadata Initiative was an organisation formed 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 used in many web pages: "Title",

"Creator", "Subject", "Description", "Publisher", "Date", "Contributor","Type", "Rights", "Coverage", "Source", "Language", "Identifier", "Format".

RSS (Rich Site Summary).

RSS is a widely used XML specification for website structural description. RSS was originally developed by Netscape 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. The following is an example of an RSS specification:

```
<?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 2.4: RSS Example

Additional elements include <image> and <lastBuildDate>. 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..

2.3.6 RDF (Resource Description Framework).

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 and 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 et al. 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 take-up, and has seen rival approaches to semantic description such as Shoe (Heflin and Hender 2000).

IEEE/LOM.

The IEEE Learning Object Metadata has gained some popularity in the Information Technology community for the encoding of curriculum resources (Steinacker, Ghavam 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" (raw media data) to "4" (collections of documents).(IEEE 2002). There have been recent efforts to align the IEEE LOM with an XML specification and RDF specification.

SCORM (Sharable Content Object Reference Model).

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 also represents a detailed 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.

OWL and Semantic Web.

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 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.

Content Reuse = Multiple Ontologies

The drive to automatic classification is in part related to the difficulty of effective semantic description of content. Semantic description is never at the forefront of the author's priorities, and this issues are potentially exacerbated by the inaccessibility of complex ontologies such as the Semantic Web. Increasingly complex ontologies have emerged, such as DAML+OIL - a merging of two other onotologies comprising DAML-ONT (an early result of the DARPA Agent Mark-up Language (DAML) programme3) and OIL (the Ontology Inference Layer)(Fensel, van Harmelen et al. 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 and 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 and Pleumann 2002), p1.

There is little doubt that the drive for 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 important stakeholders. The Semantic Web has yet to see significant organisational or industry backing - "the semantic web is still a vision." (Lu, Dong et al. 2002) Lack of satisfaction with progress with the Semantic Web has seen some emergent criticism of the slow progress (Ogbuji 2003), the development of alternative strategies (Heflin and Hender 2000). Content reuse must therefore continue to deal with multiple ontologies for metadata content exchange:

SOAP and Web Services

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. 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..

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 mark-up, 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.

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, & 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" which enable the discovery and integration of business functions (for use by applications) and accessible through the internet(Chung, Lin et al. 2003; Geng, Gopal et al. 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 is:

- Modularise and encapsulate 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 which fully describes their service
- Exhibit "plug-and-play" characteristics at the client level. (Geng, Gopal 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.

CONTENT MANAGEMENT

The management of complex Internet sites has seen the evolution of a class of tools that aid in the site maintenance and publishing activity. These “Content Management Systems” have proliferated in the last few years, reflecting the heavy demands for editorial management of a website. As with many systems that have grown out of an immediate need within one industry segment, they have often lacked the maturity of software that has seen several years of industry development. Nevertheless, if properly conceived, this class of software offers some new opportunities for the long-term effective management and reuse of multimedia content. This paper proposes a theoretical model that should act as a standard for continued evolution of this class of software, and as a framework for evaluation of existing software offerings.

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 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.

Content Management Systems as they have evolved in the Internet environment have sought to meet the immediate requirements of website publishing (Celentano, Pozzi et al. 1992) in order to facilitate:

- currency of information; that is, the need for a framework through which website content can be easily updated;
- technical elements of publishing; including the mechanics of delivering changes to the website (FTP, WebDAV, dynamic delivery, etc);
- design standards; the editorial control of information content and presentation, and the management of “corporate” design standards.

Websites typically go through several generations of design. The underlying content, however, may be similar between these generations of graphical design. 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. Content re-use 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. Evident here is a tension between the separation of content and style for purposes of effective management and content re-use, and the cohesion of content and style to enhance effective editing of the content. The paradigm of WYSIWYG (What You See Is What You Get) is highly intuitive for the content creator, but stands opposed to the structural need to decompose content elements and the stylistic elements for purposes of long-term management.

More recently the abstraction of site structural information as an integral element of site control has been explored (Fraternali and Paolo 2000). A content management model, to achieve the goal of both content re-use and content preservation, will necessarily provide visual presentation of the structure of information represented in a site.

Early software implementations of Content Management Systems focussed on the primary generation and maintenance of website content, and their origins lie in simple content delivery mechanisms for web publishing (Fraternali 1999; Yeh, Chang et al. 2000). 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. This area of industry discussion is very topical, and is best followed through current lists such as cms-list (Barrett). The explosive growth of Content Management Systems highlights rather than diminishes the potential for some theoretical convergence in the area of content re-use, as website publishers struggle with issues of currency, navigation and editorial consistency on their sites.

Obsolescence And Technology Architecture

The widespread availability of multimedia content authoring systems allow the content creator 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 speciation. 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 World 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, content storage formats are evident. This is classic entropy as it affects our information resources.

There is a fundamental architectural dissonance between the creative community and the technical community who are responsible for developing Content Management

Systems. Clearly there are at least some in the arts and humanities who are trying to reach an audience through time as well as space, and who want their artistic and research creations to last through centuries rather than decades or months. Clearly, the practical life of software innovations and digital technology spans a decade at most, commonly several years only, and at the worst barely a few months.

Nevertheless, there is an inevitable move toward using new media technologies for delivery of digital content in many different forms, if only because the audience the creators and innovators are trying to reach are those using that very media.

Yet while the archival management of the book and other printed media is well understood, similar archival strategies in the new media are in their infancy. There are daunting challenges facing those who attempt to preserve in the long term digital creations of contemporary times. There is an expectation of those expending creative effort in the new media that their efforts will be available in the long term. The cost burden of long term persistence of content in current formats has traditionally been borne by Libraries. However, few libraries have the resources to sustain the research and development required to address archival issues of Information Systems. 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).

In the long term, addressing the issue of obsolescence requires continuous transformation of multimedia content to ensure it is available through current delivery vehicles. A central role for Content Management Systems will be the efficient management of this transformation process over time.

Ephemera and Identification

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, Pennock 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 journal publications. Lawrence analyses the problem broadly as follows:

"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, Pennock 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.

The identification, access and archival characteristics of a traditional journal citation and a URL citation are very different:

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

Figure 1: Content identification

The URL has considerable value in the short term. There are experiments in archiving sites, such as the Pandora project and the National Library of Australia (Phillips 1998). However, this archive 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 – that is, which is *essentially* dynamic, the issues of attempting to preserve content in its final generated form become meaningless.

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 elements. The organisation of the content must lend itself to the effective replication of content for archival purposes and the efficient reuse of content.

Multimedia database and Content Identification

The multimedia publisher commonly employs a variety of different tools for content creation (Brett and Nash 1999; Michalas 2000). The process of creation, therefore, can involve the complex interaction of different content creation systems. Work by (Cheung and Chanson 1999) shows the added complexity of reaching and identifying multimedia resources over heterogenous networks. They propose three different models: configuration, user control and presentation. Complexities of the software management of multimedia content are explored by (Vazirgiannis, Kostalas et al. 1999) and (Agoulmine, Dragan et al. 2000). Multiple methods of publication extend the concern of publishers into the area of product support as well as the production and distribution of their content(Reynolds and Kaur 2000).

The once-off publication of multimedia content might, of its nature, not require an inordinate investment in the long-term management of the content itself. The

regeneration of multimedia content presents a challenge in every stage of the publication process.

The Content Management Systems must therefore facilitate the effective management of the heterogenous elements of complex multimedia elements, some stored locally and some sourced externally. The multimedia content database supporting a Content Management System must support the demands of:

a) content organisation in an manner which maximises content identification and dissemination;
b) management of connectivity information for external content items and content repositories;
c) elemental organisation of content, content structure and meta-information relating to content in a manner which enhances content re-use.

Content Indexing and Classification

Library Science has long had effective 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. 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 (*eg* 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.

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. Classification schemes evolved as a response to the escalating difficulty of managing and organising large 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. The 1990's saw a movement from a communication model of Information retrieval to a behavioural model of information seeking(Kuhlthau 1993), with detailed analysis of this process by McCreadie and Rice (Kuhlthau 1999; McCreadie and Rice 1999)and more recently applied in the multimedia context(Collins 1999). However, the analysis of Internet-based search behaviour has tended to remain isolated from the discipline of Information Research Behaviour, with recent papers such as (Wang 2000) focussing on a communication model of information search behaviour on the Internet. Research constructs described by (Mann 1993) in his “methods of searching” model have been extended to the analysis of discipline-specific “material mastery” in the behavioural elements of digital library usage (Covi 1999). Over large multimedia content databases, the challenge of effective indexing of content is still being explored (Amato, Rabitti et al. 1998). Content indexing of video and sound archives present particular problems (Dagtas, Al-Khatib et al. 2000). The behavioural elements in the process of information retrieval present a basic challenge not only in information retrieval, but also in the maximisation of the re-use of content.

Disciplines of cataloguing are rarely seen on the Internet. Some descriptive information that leads to the physical equivalent of the shelf location, the 'URL', is used: the most common example is the use of *meta-tags* in HTML to describe the page itself. In many ways the digital information era recalls that chaotic era prior to the adoption of universal schemes for cataloguing and classification. The "Dublin Core" was an early effort to define some standards in this area.

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 independent of the intellectual rigour surrounding classification systems. Meta-data classification of content needs also to address the classification of content at the elemental level, not just of content in is final form, and in a manner that enhances the distributed content identification and dissemination of the content.

The intersection of the problems of information retrieval with the management of digital collections were highlighted by (Wood 1998). The common use 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 database to enhance the process of retrieval.

Central to the role of a Content Management System, therefore, must be the flexible capture of the content itself, and the effective and consistent meta-description of the content at the elemental level. Presentation of the content structure should reflect the behavioural aspects of the content creators in information visualisation.

A MODEL FOR CONTENT MANAGEMENT

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. As (Fraternali 1999) has indicated, Content Management in a web-site context can architecturally be described as a hybrid between hypermedia and an information system.

In particular, a systematic model for content management must address:

- a model for content capture, separation and mark-up;
- an effective storage model for multimedia content and for unique content identification;
- a system for integration with authoring systems and for building associations between content stored within the Content Management System and content located externally;
- a workflow management process for the editorial control of the publishing process;
- an information retrieval and modelling interface to facilitate content reuse and expose the content structure;
- a final content delivery mechanism to deploy content through heterogenous runtime environments.

- A system for content version control and distributed editing.

Industry efforts in building Content Management Systems characteristically focus on discrete functional challenges they are addressing: original publishing, content update, efficient content delivery, etc. This paper and the research focus in this area at Sydney University is in the elaboration of a model that can form the conceptual basis for further work in this area, as well as specific issues in multimedia content generation and reuse. *Table 1* outlines this model.

Table 1: A model for Content Management

Layer 4: **Content Publishing/Runtime delivery Layer** Generative systems integrated with runtime engines and presentational formats. Content mapping systems (eg XAS) Web servers, Mail servers, etc
Layer 3: **Content Information Retrieval and Modelling Layer** Search engines (eg Z39.50) Clustering techniques Structured Indexing techniques Content and Site visualisation and modelling
Layer 2: **Content Storage, Identification and Reuse Layer** Unique identification and version control of content elements Management of the structural elements as well as the content elements Processes for Elemental decomposition of Content Rule-based content-relationships Structural content-relationships Management of the content cross-referential structures
Layer 1: **Content Creation, Capture & Separation Layer** Rule system for separation and markup of content and structure (eg XML) Content capture systems and Authoring systems

Content Creation, Capture & Separation Layer

Content capture is a non-trivial process in a heterogeneous media 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 mark-up systems. With content reuse as an objective, the first prerequisite for effective content reuse is the decomposition of content into elemental content items.

In qualitative data analysis tools such as NUD*IST, the elemental decomposition of content is fundamental to the effective analysis process. In this case, the software treats this as a text unit of a line, paragraph or document. By this means, the qualitative analysis of this content (the coding) can be annotated at the relevant elemental level. Subsequent dissection and reporting of the coding hierarchy is possible. The original document is maintained intact in this framework, while a level of "meta" information is built around the relevant portions of the document. The software provides a framework focussed on interpretation of the original text through layers of coding abstraction. The software maintains the integrity of the original document and exposes for research purposes the layers of interpretation built upon the source documents.

In addition to the research narrative surrounding an original document, content publishers have a commercial requirement to track the copyright clearance levels of content they use. Similarly, from an archivist's point of view, the management of digital content presents dilemma of multiple copyright ownership (Jewitt 2000), and the fact much relevant audio content is not in the public domain. Aspects of the social and copyright issues of multimedia publishing of academic journals explored by (Cox 1998) are equally applicable to the generation and regeneration of multimedia content in an Internet framework.

The management of online educational curricula is perhaps the leading area in which the challenges of content re-use are being actively engaged. There are manifest advantages to the effective reuse of curriculum 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 issues of architectural standardisation implied by such a goal are discussed by Colbert (Colbert, Peltason et al. 1997; Candler and Andrews 1999) and many others.

SGML has played a key role in the definition of a model for content mark-up in a manner that effectively describes the separation of content, structure and specific presentation of text. The rapid uptake of XML demonstrates the desire for an effective means for the description of complex textual forms inadequately addressed by simple text stores. In literature, the Text Encoding Initiative, an SGML standard for content mark-up of Literary works serves as a framework for consistent digital encoding of large bodies of literary works. Non-textual content standards such as MusicML show the popularity of XML as a vehicle for mark-up and content identification in other media.

The rapid obsolescence of information, both in content formats and runtime engines makes the regeneration of this content with revised or updated material an early exigency, particularly in the management of Website Content (Reynolds and Kaur 2000 need a better source) and in the Education arena (Cheung and Chanson 1999).

The abstraction of content therefore serves the additional role of attempting to retain underlying content value in a manner which:
engages the dilemma of long-term archival retention of the content
provides a framework for re-generation of the same content in new or other media delivery architectures.

Content Storage and Identification Layer

Content mark-up goes only partway toward the construction of systems for subsequent content re-use. Architectural considerations are of profound importance in the management of content. While database vendors are increasingly providing database tools which integrate the management of multimedia audio and video content, the efficiencies of the traditional OS file system often make it the popular choice for storage of audio, video and image content itself, while meta-data referencing the content is retained in database tables.

In this context, SGML as a standard is only part of an overall solution. A long-term tension between the communities building SGML-like architectures such as XML, and the database communities supporting large, high performance databases, has not lent to a happy resolution of problems in the area of very large databases of digital content. If anything, the problems of obsolescence are exacerbated rather than aided by current trends.

The hierarchical network model for content mark-up represented by SMGL needs to be mapped to a database framework which:

- Uniquely identifies elemental components of the content for purposes of syndication and re-use;
- For purposes of effective editorial control tracks the relationships between content as it has been used in various different forms.

A key area for further research is the effective database integration of heterogenous multimedia sources in a manner which not only addresses the storage issues of that content but the consistent content identification issues. To address 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. This is a key prerequisite to realising archival value of content in the long term. This is an issue of standards not only in the content storage but also at the national level for the consistent meta-identification of the content in a way that allows archival content repositories to be established. The Book Deposit scheme at National Libraries has been an effective mechanism for ensuring the preservation of key published content in print form within the national collection. The complexities of content identification and the technological incompatibilities at the content capture layer make it all the more important for the development of meta-identification standards that go beyond the limited Dublin Core.

Content Information Retrieval/Modelling Layer

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 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. In the management of multimedia resources, the integral relationship between the technological platform for the information delivery and the digital content present considerable problems.

The information retrieval layer of a Content Management System must realise several goals:

a) the effective discovery of content that may be distributed across different content stores, in different languages and storage architectures

b) the modelling of content toward effective content re-use,

c) the cost-economical persistence of access to content.

While standards such as Z39.50 have yielded good results in the content dissemination and information retrieval of academic collections they require considerable development if they are to address the ephemeral nature of content in the multimedia environment.

Content Publishing and Reuse

The diversity of authoring tools that feed into the multimedia production process compounds the difficulty of content reuse (Agoulmine, Dragan 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 and Kaur 2000), with little exploration of the issues of content re-use. The requirements of content reuse within Multimedia Content Systems are explored by (Liu 1999) from the perspective of document management systems, whilst the high cost of multimedia content publication(Kelly 2000) make content management vital. Nevertheless, the integration of workflow systems will need to go beyond the domain of the publication generation and into the arena of rights administration. Similarly, the complexity of the heterogonous tool-set involved in the content creation makes the computer automation of processes involved in the generation of content all the more important. While the concepts underlying workflow management of business flows are well defined, they tend to be poorly defined in the editorial management of content databases, although it is clear that many concepts relating to workflow management apply; in particular:

* availability of graphical tools
* routing capability
* queue management of tasks
* management of events
* task and process management(Perez and Rojas 2000).

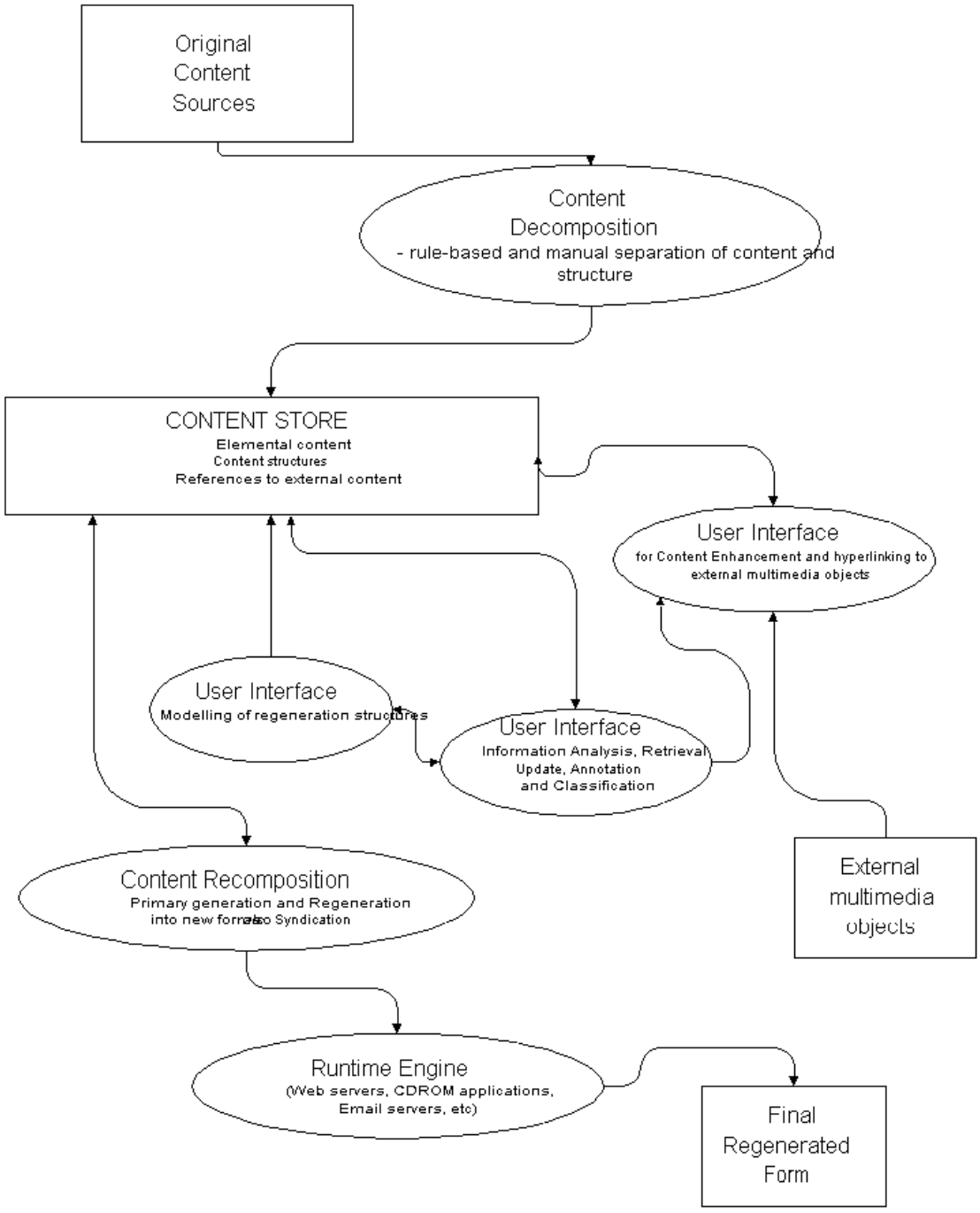


Figure 2: The process flow for content re-use

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.

Online Educational delivery continues to be a source of considerable experimentation in the construction of educational objects that can easily be deployed in a variety of educational delivery environments, and to a range of target audiences. 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.

Website content delivery platforms can broadly be characterised in two models of operation:

- a generative process that produces a "static" result on the media
- a dynamic generative process that delivers the final presentation of the content on demand.

Performance and architectural issues are the driving factors behind these two models for content delivery. A traditional website content serving static pages or with minimal dynamic generation of content still represents the most speed-efficient runtime engine (Mendes and 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 is essentially one of the timing for content recomposition. The final, delivered, form 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.

Figure 2 describes the process flow for the content regeneration cycle, through the stages of original content capture, of content enhancement through inter-linking with external resources, the stages of content identification, information modelling and retrieval, and finally the regenerative process for content re-use in various runtime contexts. 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.

SYSTEMS THINKING

Management of complex and dynamic content websites has seen the evolution of a class of software to meet the management dilemma for effective control of these websites: the "Content Management Systems". It is the proposition of this paper that a well-described theoretical model for this class of software will enrich the potential for these systems to meet increasing demands for content reuse, archiving and version control. To achieve this, a cross-disciplinary model bringing in concepts of database management, information retrieval and some elements of workflow control is proposed. Such a model can serve as a basis to guide further development of systems in this area and as a benchmark for judging the functional completeness of products.

QUESTIONS

What is the role of the Content Management System?

What is the difference between a Static Web Server and an Application Server?

Describe a workflow model for website publishing.

Describe the purpose of RSS for content delivery

What is the function of the CGI in website information delivery.

What is the place of the Digital Library in information delivery?

FURTHER READING

Advanced Distributed Learning. (2001). Sharable Content Object Reference Model Version 1.1 (http://www.adlnet.org/).

Agoulmine, N., D. Dragan, et al. (2000). "Trouble management for multimedia services in multi-provider environments." Journal of Network and Systems Management 8(1): 99-123.

Amato, G., F. Rabitti, et al. (1998). "Multimedia document search on the Web." Computer Networks and ISDN Systems 30(1-7).

Anderson, K. M. (2001). Data scalability in open hypermedia systems. Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots, February 21 - 25, 1999, Darmstadt Germany, IEEE.

Balnaves E (2004). "Regeneration and Reuse of Content in the World Wide Web: a Model-based Approach" Doctoral Thesis.

Barrett, C. (2001). Camworld: Content Management Systems. 2002.

Brett, P. A. and M. Nash (1999). "Multimedia language learning courseware: a design solution to the production of a series of CD-ROMs." Computers & Education 32(1): 19-33.

Candler, C. S. and M. D. Andrews (1999). "Avoiding the great train wreck: Standardizing the architecture for online curricula." Academic Medicine 74(10): 1091-1095.

Celentano, A., S. Pozzi, et al. (1992). A multiple presentation document management system. Proceedings of the 10th annual international conference on Systems documentation.

Cheung, S. C. and S. Chanson (1999). "A model-based authorware for the construction of distributed multimedia systems." Information and Software Technology 41(11-12): 715-727.

Chung, J.-Y., K.-J. Lin, et al. (2003). "Web Services Computing: Advancing Software Interoperability." Computer 36(10): 36-37.

Colbert, M., C. Peltason, et al. (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. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques,.

Collins, P. (1999). "Document and workflow management '99." Management Services 44(1): 23-26.

Covi, L. M. (1999). "Material mastery: situating digital library use in University research practices." Information Processing & Management 35(3): 293-316.

Cox, J. E. (1998). "The Changing Economic Model of Scholarly Publishing: Uncertainty, Complexity, and Multimedia Serials." Library Acquisitions: Practice & Theory 22(2): 161-166.

Dagtas, S., W. Al-Khatib, et al. (2000). "Models for motion-based video indexing and retrieval." IEEE Transactions on Image Processing 9(1): 88-101.

DCMI (2001). Dublin Core Metadata Initiative (www.dublincore.org).

Duffy, K. (2000). "Content management comes of age: streamlining the information creation and delivery process." Information Management & Technology 33(4): 183-91.

Ekman, R. H. (2000). "Can Libraries of Digital Materials Last Forever?" Change 32(2): 23-35.

Fensel, D., F. van Harmelen, et al. (2001). "OIL: an ontology infrastructure for the Semantic Web." IEEE Intelligent Systems 16(2): 38-45.

Fraternali, P. (1999). "Tools and approaches for developing data-intensive Web applications: a survey." ACM Computer Survey 31(3): 227 - 263.

Fraternali, P. and P. Paolo (2000). "Model-Driven Development of Web Applications: The Autoweb System." ACM Transactions on Information Systems 18(4): 323-341.

Geng, X., R. D. Gopal, et al. (2003). "Scaling Web Services with Capacity Provision Networks." Computer 36(11): 64-72.

Haustein, S. and J. Pleumann (2002). Is Participation in the Semantic Web Too Difficult?. The Semantic Web - First International Semantic Web Conference, Springer, Heidelberg.

Heflin, J. and J. Hender (2000). Semantic Interoperability on the Web. Proceedings of Extreme Markup Languages 2000, Graphic Communications Association.

IEEE (2002). Draft Standard for Learning Object Metadata, IEEE.

Jewitt, C. (2000). "Digital chaos and professional standards." Multimedia Information and Technology 26(2): 148 - 150.

Kelly, B. (2000). "Will it all fit together? The need for standards and the technical challenges." Multimedia Information and Technology 26(1): 67-72.

Kuhlthau, C. C. (1993). Seeking a Meaning: A process Approach to Library and Information Services. NJ, Norwood.
Kuhlthau, C. C. (1999). "Role of experience in the information search process of an early career information worker: Perceptions of uncertainty, complexity, construction, and sources." Journal of the American Society for Information Science 50(5): 399-412.
Kunze, J. (1999). RFC 2731: Encoding Dublin Core Metadata in HTML, IETF. 2001.
Lancaster, F. W. (1968). "Interaction between requesters and a large mechanized retrieval system." Information Storage and Retrieval 4(2): 239-252.
Lassila, O. and R. Swick (1999). Resource Description Framework: (RDF) Model and Syntax Specification -W3C Recommendation 22 February 1999 (http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/), w3c.
Lawrence, S., D. M. Pennock, et al. (2001). "Persistence of Web References In Scientific Research." Computer 34(2): 26-31.
Libby, D. (1999). RSS 0.91 Spec, Revision 3 (http://mynetscape.com/publish/formats/rss-spec-0.91.html), Netscape.
Liu, P. a. H., Liang H (1999). Multimedia Document Systems From Authoring perspective. Internet and Multimedia: Systems and Applications. B. Forth, IEEE Press.
Lu, S., M. Dong, et al. (2002). "The semantic web: Opportunities and challenges for next-generation web applications." International Journal of Information Research 7(4).
Mann, T. (1993). Library Research Models: a guide to classification, cataloguing and computers. NY, OUP.
McCreadie, M. and R. E. Rice (1999). "Trends in analyzing access to information. Part I: cross-disciplinary conceptualizations of access." Information Processing & Management 35(1): 45-76.
Mendes, M. A. S. and V. A. F. Almeida (1998). Analyzing the impact of dynamic pages on the performance of Web servers. 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, CMG.
Michalas, V. e. a. (2000). "A comparison of multimedia application development platforms towards the object web." Computer Standards & Interfaces 22(1): 13-26.
Ogbuji, U. (2003). The past, present and future of web services, webservices.org. 2003.
Perez, M. and T. Rojas (2000). "Evaluation of Workflow-type software products: A case study." Information and Software Technology 42(7): 489-503.
Phillips, M. E. (1998). "Tomorrow's incunabula: preservation of Internet publications." Lasie 29: 5-10.
Reynolds, J. and A. Kaur (2000). Content Management.
Santini, S. J., R. (1998). Beyond Query by Example. Proceedings, ACM Multimedia 98, Bristol, England, Sept 12-16 1998, Bristol, ACM.
Smith, W. (1998). "Lost in cyberspace: preservation challenges of Australian Internet resources." Lasie 29: 6-25.
Steinacker, A., A. Ghavam, et al. (2001). "Metadata Standards for Web-based Resources." IEEE Multimedia 8(3): 70-76.
Vazirgiannis, M., I. Kostalas, et al. (1999). "Specifying and authoring multimedia scenarios." IEEE Multimedia 6(3): 24-37.

Wang, P. H., William B. and Tenopir, Carol (2000). "Users' interaction with World Wide Web resources: an exploratory study using a holistic approach." Information Processing & Management 36(2): 229-251.

Wood, M. E. J., Campbell, N.W. and Thomas, B.T (1998). Iterative Refinement of Relevance Feedback in Content-based Digital Image Retrieval. Proceedings, ACM Multimedia 98, Bristol, England, Sept 12-16 1998, Bristol, ACM.

Yeh, J.-H., J.-Y. Chang, et al. (2000). "Content and knowledge management in a digital library and museum,." Journal of the American Society for Information Science 51(4): 371-379.

CONTENT AND CONTENT SERVER STANDARDS

BORENSTEIN, N. and N. FREED (1992). RFC 1341: MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies, W3C. 2002.

CHAMBERLIN, D., J. CLARK, et al. (2001). XQuery 1.0: An XML Query Language - W3C Working Draft 07 June 2001 (http://www.w3.org/TR/xquery), W3C.

CROCKER, D. H. (1982). RFC 0822: ARPA Internet Text Messages, W3C. 13 July 2000.

FRANKS, J., P. Hallam-Baker, et al. (1999). RFC 2617: HTTP Authentication: Basic and Digest Access Authentication. 2001.

GETTYS, J., J. Mogul, et al. (1999). RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1. 2001.

GOLAND, Y. Y., E. Whitehead, et al. (1999). RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV:. 2001.

GUENTHER, R. (1997). RFC 2220: The Application/MARC Content-type. 1991.

JOURDAN, M., C. Roisin, et al. (2000). "A scalable toolkit for designing multimedia authoring environments." Multimedia Tools & Applications 12(2-3): 257-79.

KUNZE, J. (1999). RFC 2731: Encoding Dublin Core Metadata in HTML, IETF. 2001.

LASSILA, O. and R. Swick (1999). Resource Description Framework: (RDF) Model and Syntax Specification -W3C Recommendation 22 February 1999 (http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/), w3c.

LIE, H. W. and J. Saarela (1999). "Multipurpose Web publishing using HTML, XML, and CSS." Communications of the ACM 42(10): 95-101.

MATHEWS, B., D. Lee, et al. (1998). Vector Markup Language (VML): World Wide Web Consortium Note 13-May-1998, W3C: http://www.w3.org/TR/1998/NOTE-VML-19980513.

NEBEL, E. and L. Masinter (1995). RFC 1865 - Form-based File Upload in HTML. 2001.

NEIGUS, N. (1973). RFC 0542: File Transfer Protocol. 1992.

SOLLINS, K. and L. Masinter (1994). RFC 1737: Functional Requirements for Uniform Resource Names, ITEF Network Working Group. 2001.

SPERBERG-MCQUEEN, C. M., C. Huifeldt, et al. (2000). "Meaning and Interpretation of Markup." Markup Languages: Theory & Practice 2(3): 215-234.

Chapter 6: Information Retrieval Systems

We have explored the concepts around content delivery using the Internet as a platform for content delivery. Getting information *out* is, of course, only half of the problem.

The concepts of Information Retrieval have grown over the last two centuries to become the backbone of research activities of all sorts. The *Renaissance* is a term used to describe the rediscovery of much knowledge that was known widely to the ancient world but only rediscovered in Medieval Europe due to the diligent efforts of monastic libraries and the then more open intellectual milieu of Islam. This "rediscovery" of knowledge was also propelled by the invention of the printing press and the wide distribution of printed texts that was previously inconceivable. With the growth in publishing of all sorts, the increasing specialisation of the topics of publishing, and the natural human endeavour to organise and arrange was born the disciplines of cataloguing and classification, and thereafter of Information Retrieval systems.

Cataloguing has one fundamental purpose: to fully and accurately describe an item in a way that enables it subsequently to be found. The Library Catalogues is the progenitor of all other subsequent catalogues and database: it describes the author, title, publisher, classification, shelf location, sometime the table of contents, physical book information, book identifiers and so on. Strict disciplines of description (eg Anglo-American Cataloguing Rules) defined standards for description in a manner that allowed recognition in a consistent manner of an item: rules for handling multiple authors, lack of authors, consistent description of authors across multiple variants, etc.

Such disciplines of cataloguing are rarely seen on the Internet, but descriptive information that leads to the physical equivalent of the shelf location, the 'URL', are used: the most common example being the use of meta-tags in HTML to describe the page itself – for example the "Dublin Core".

Cataloguing therefore aims to give us a means of *locating* an item. It does not necessarily tell us anything *directly* about what the item is: its subject areas, its relationship with other items of the same subject. It was to bring the multitude of books together in some logical order that drove the invention of *Classification* schemes in the 19th Century, by Melvil Dewey and others.

> "It is hard to realize today that undergraduates at that time were taught much like high-school students today: they worked from lecture notes and textbooks and were not required to write research papers or otherwise use the library. No students lived on campus, and classes were held continuously from morning prayer until the afternoon, when the students went home. The Columbia College Library was open only a few hours a day and acquired most of its books by donation. Until the graduate programs began, its importance in the life of the College was minimal. Only a few decades earlier, it had been run by a combination librarian and janitor, and at other times it was a part-time assignment for some unlucky professor. "
> http://www.columbia.edu/~brennan/library/gl.history.html

Classification is an intellectual process: that is, it involves the (hopefully) careful judgement of an item to identify its relevant subject area(s). The logic of ensuring that all copies of a book or journal were shelved in a *single* physical location has meant that classification schemes invariably were forced to identify a single primary "classification" for an item. The fundamental principle of classification is to describe a "hierarchic" view of knowledge categories, and to describe within a given "tree" of knowledge each of the different branches by which that "tree" is classified.

Subject cross-referencing of classification schemes grew from the immediately evident problem few items to be classified could be completely described by *one* classification code. Thus grew the Library of Congress *Subject Headings* and other Library schemes for non-hierarchic descriptions of an item. In particular, an item could attract many subject headings whilst still limited by a single shelf location and classification.

It should be noted that the library catalogue and classification system was an innovation of fundamental importance in managing the growth of information and still has a fundamental role in effective information management.

The mid-60's availability of large computer systems to libraries made possible several key techniques that have driven innovations what was to become the new area of Information Retrieval.

The National Library of Medicine was at the forefront of innovation in the management of large volumes of information that could not be described and accessed in computerised catalogues. In the 1960's, the NLM turned to computers to assist in the daunting increase in the number of medical journals and books that it was indexing in its printed publication: Index Medicus. They built a computerized literature retrieval system, known as MEDLARS, launched in 1964. MEDLARS was the precursor of many of the online database systems now available, such as their own MEDLINE. .



(http://www.nlm.nih.gov/exhibition/phs_history/102.html)

Case Study: Dewey Decimal Classification (1883-1889)

In 1883 Melvil Dewey accepted the post of Chief Librarian at Columbia. This event can be considered the beginning of the Columbia University Libraries as we know it today. Columbia College was starting to build its graduate schools, and Prof John Burgess of the Faculty of Political Science had convinced President Barnard that the library needed better organization and management to meet the needs of graduate education.

It is hard to realize today that undergraduates at that time were taught much like high-school students today: they worked from lecture notes and textbooks and were not required to write research papers or otherwise use the library. No students lived on campus, and classes were held continuously from morning prayer until the afternoon, when the students went home. The Columbia College Library was open only a few hours a day and acquired most of its books by donation. Until the graduate programs began, its importance in the life of the College was minimal. Only a few decades earlier, it had been run by a combination librarian and janitor, and at other times it was a part-time assignment for some unlucky professor.

In the last decades of the 19th Century, many graduate and professional divisions were established at Columbia, and many new subjects, including science, social science and modern literature, were admitted to the liberal arts curriculum, many as electives. Most of Columbia's "schools" date from this period, and in 1896 the name of the corporation was changed to Columbia University in the City of New York. (The former undergraduate School of Arts, that is liberal arts, was then renamed Columbia College, causing a confusion that has plagued Columbians ever since.)

Melvil Dewey had formulated the decimal classification scheme for which he is most famous while working in the Amherst College Library during his junior year of college, 1872-73. He convinced the college to adopt it and stayed on as Assistant Librarian until 1876, when he published the first small edition of the system and reported on it at the first American librarians' conference in Philadelphia. After leaving Amherst, Dewey established a company in Boston called the Library Bureau to manufacture and distribute library supplies, and wrote and lectured on the subject of library organization.

The library problem Dewey found at Columbia in 1883 was the same thing he had seen at Amherst, though on a larger scale: a library growing sharply in size and use but with an inadequate way to locate and arrange the books. A few years earlier, the library had become so hard to use that Prof Burgess, while setting up the first graduate programs, got the Trustees' approval to start a separate library for history and political science. Burgess also convinced President Barnard to put up a new library building (on the Madison Avenue campus at 49th St) and to hire Dewey to reorganize. There are several surviving photographs of the interior of the new Library Building, with its balcony level and high peaked wooden ceiling. The books were shelved around the walls of two or more reading rooms. In some photographs, single letters can be seen marked on the walls; possibly these oriented readers to bookcase locations, much as range numbers are used in bookstacks.

With the decrease in the cost of storage and dissemination of information, the 1990's saw the widespread availability of online full-text information resources, containing not only the bibliographic reference to an item, but also the full text of that item. Such a revolution in information access has had consequences not only in the capabilities available to researchers, but also the very manner in which research is undertaken across all disinclines:

> "Computer-based searches lead individuals beyond traditional genre boundaries, as they easily cover a wide range of texts…" Ruhleder (1995) p.188

DIGITAL DATABASE OPPORTUNITIES

Libraries, still the quintessential information content suppliers, are attempting to build models for information retrieval in the context of multimedia content of complex variety and type. The Library of Congress, in building its platform for a Digital Library, has focussed not only on the technological delivery issues, but also the behavioural aspects of information retrieval. The research behaviour in a particular discipline can itself affect the methodology and approach to information retrieval.

The digital delivery of information has opened opportunities beyond that of information delivery. Indeed, the building of information delivery systems has created new opportunities who focus on *analytical systems* for text and content analysis.

The availability of easily networked, digital databases, presents many opportunities for research not possible with the traditional catalogue:

- Literary analysis using statistical techniques on the source text.
- Keyword and other content searches across the all descriptive information and the text itself.
- Grouping of like-documents together by inherent content rather than classification ("clustering" techniques).

Database capture of religious texts also offers new opportunities for the analysis of the full corpus of texts for a given period. In the last century one of the most comprehensive compilations of the works of the early Christian Fathers, the *Patralogia Latina* was published. Its availability in searchable full text form dramatically changes the ability to analyse this text. Such database systems can also allow the greater dissemination of such texts that would otherwise have been possible. *Project Gutenburg* is focussed on formulation of just such a database of full text content. Libraries such as the Vatican have also undertaken major digitisation programmes in order to provide scholarly access to the original manuscript versions of texts in their Libraries. The use of hypermedia to annotate the multicultural discourse surrounding historical events also transforms the communication of historical information. The *Endeavour Project* at the Australian National Library is a project which is publishing the journals of

Captain James Cook in the first Pacific voyage (1768-71). These journals record many indigenous Australian and Pacific islander encounters which have profoundly influenced European intellectual and cultural history. The purpose of this digital collection is to integrate the cultural discourse surrounding this encounter in one location (Turnbull, 2000).

Case Study: The Library of Congress Digital Library Project

Digital libraries are the logical extensions and augmentations of physical libraries in the electronic information society. As extensions, digital libraries amplify existing resources and services and, as augmentations, they enable new kinds of human problem solving and expression. High levels of attention and funding were first given to digital libraries in the early and mid 1990's, leading to a plethora of visions and projects invariably driven first by finding ways to apply the many technologies developed in the 1980's and second by desires to create new technologies for managing distributed information resources (see April 1995 issue of Communications of the ACM for descriptions of various DL projects and the May 1996 issue of IEEE Computer for descriptions of six prominent projects). In some cases, notably the efforts of national libraries or large academic libraries, efforts focused on extending access to existing collections through digitization and network access. At present, digital library projects have focused on developing digital collections and providing rather limited access services.

Marchionini (in press) has characterized DL research and development as falling into four categories: content, services, technology and culture. Research issues related to content include the integration of multimedia objects; data acquisition, including analog to digital conversion; metadata extraction and standardization; procedures for indexing, storage and retrieval; workflow processes and management and collection preservation and maintenance. Service research issues are strongly dependent on user interfaces and include search, filtering and browsing; reference and question answering and instruction. Technology research efforts are mainly related to high-speed networking, search engines, interface design, security and billing and interoperability across many DLs. The culture issues include intellectual property; insuring data quality, privacy and equity and organizational interfaces for various communities of practice. In addition to these research and development challenges, meta issues related to managing and evaluating DLs and their impact on people and organizations are also active areas of study.

The Library of Congress (LC) has long served as an exemplar for library practice and led many institutions in defining library automation needs. In 1995, LC initiated an ambitious National Digital Library Program that aims to digitize and make available five million objects from a variety of LC collections. Funded by corporate and foundation donations, the program will not only develop one of the premier digital libraries for American culture, but also provide an opportunity for researchers and practitioners to address many technical and intellectual challenges that DLs provide. The program is pushing the state of the art in digitizing large volumes of non-book and sometimes fragile objects, creating high quality packages for K-12 classrooms, breaking new ground in corporate/government partnerships, operationalizing sophisticated search services and creating innovative interfaces for public and sta• users (see www.loc.gov). In 1995, the LC NDL Program contracted with the University of Maryland's Human-Computer Interaction Laboratory to collaborate on the interface development for the NDL.

The focus here is on user interfaces and the work of researchers creating interfaces for information retrieval and data visualization is particularly applicable to DLs. Although there have been a number of innovative approaches to supporting search, the WWW environment has only recently begun to support the basic interaction techniques available in standalone or earlier client/server environments (e.g. multiple windows, mouse dynamics, stateful interaction, etc.). The extension of HTML (e.g. frames) and especially the development of Java are allowing designers to port interfaces to the web. **Marchionini (1998)**

RETRIEVAL TECHNIQUES

Most Internet and online Database search engines use *Boolean Logic* for the construction of search queries to retrieve what you are seeking. The construction of the search itself may be mediated by natural language query construction techniques to insulate the user from the complexities of understanding Boolean Logic. However, an understanding of basic Boolean Logic and enhance considerably your capability to research and retrieve precisely what you are looking for.

Searching is generally a process of Iterative Refinement as we get closer to the content that we are seeking. Some terminology helps to understand the process of search design. When we construct a search, our search query can be constructed with an emphasis either on:

> *Exhaustivity*: is it likely to bring out EVERYTHING related to my interest area.
>
> *Specificity*: is the query constructed to extract ONLY items that are relevant to my interest area.

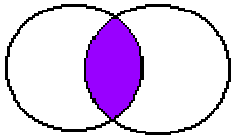
The emphasis placed on *Exhaustivity* and *Specificity* will have a direct effect on:

> *Recall*: the degree to which the search results contain everything that is available.
>
> *Precision:* The degree to which the search results actually do contain ONLY relevant information.

The construction of a complex query will generally involve the use of Boolean Logic, although the representation of this will vary. Boolean Logic is simply the mathematical representation of terms to achieve a search goal. Similar to simple algebra, Boolean Algebra follows useful rules that make interpretation reliable by search engines. The meaning of terms in Boolean Algebra is very similar to their natural language meaning, which makes its use quite easy when its operation is understood. Three key concepts are present almost universally in WWW and database search engines: AND, OR, AND NOT.

AND

If a person requests Apples *and* Oranges, their expectation is that they will receive BOTH Apples and Oranges and will not be satisfied with only one or the other. This is precisely the meaning of AND in Boolean logic.



APPLES AND ORANGES: the target content must have BOTH the terms APPLES and ORANGES.

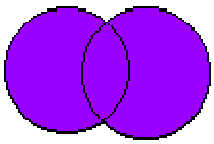
Naturally, the use of AND enhances the specificity of a search query. In WWW search engines, the use of AND is commonly indicated by ah **+** prefix. The default search uses OR logic:

> RAMPANT CAPITALISM – would search for any web page containing *either* "rampant" *or* "capitalism"
>
> +RAMPANT +CAPITALISM – would search for any web page containing *both* "rampant" *and* "capitalism" within the one web page.

Even when we refine the above example to include both "rampant" and "capitalism" we can see several problems regarding the specificity of our search. The term "rampant" may have been used in a very different context from the term "capitalism". We could get around this by using an *exact phrase* "Rampant Capitalism". However, what about "Capitalism is out of control" or even "Uncontrolled Capitalism"? A combination of terms is clearly required which involves the use of the **OR** operator.

OR

If we ask for Apples *or* Oranges, clearly we will be happy with *either*, but not with *none*. This is the precise meaning of *or* in Boolean Logic.



APPLES OR ORANGES: any one of the terms is acceptable - *but at least one reference to either APPLES or ORANGES must be present*

We can, therefore, enhance our previous search through a mixture of AND and OR logic as follows:

> CAPITALISM AND
> (RAMPANT OR "OUT OF CONTROL" OR UNCONTROLLED)

This would give a more exhaustive search. However, it would also retrieve a content item which contains: "Capitalism is the cure to rampant socialism." The **order** of operators is quite crucial, and may vary according to the search engine used: precedence is often clarified through use of brackets:

> Eg (A AND B) OR C

However "A AND B OR C" could be interpreted as "(A AND B) OR C" or as "A AND (B OR C)" with very different results, for example:

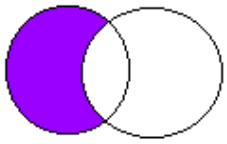
> Rampant or Uncontrolled and Capitalism ---> Would probably retrieve anything with *"Rampant"* in it irrespective of the term "Capitalism".
>
> (Rampant or Uncontrolled) and Capitalism --> Guarantees that our precedence of terms is correct.

Sub-searching is another way of creating logical "conjunctions" of concepts

Boolean Logic Basics. That is we perform an initial search to form a subset of documents or pages that result. We then perform a further search only on that result set. This is the functional equivalent of the AND operator.

AND NOT

Our previous example may well result in search results that are not relevant to our interest area at all.



APPLES AND NOT ORANGES: the document retrieved must contain a reference to APPLES only where there is no reference to ORANGES.

For example, we may specifically be interested in industrial pollution. One approach would be to exclude terms we do NOT wish to see – using AND NOT:

(Rampant or Uncontrolled) and Capitalism AND NOT Socialism

This structure is generally supported in WWW search engines with a "-" operator on terms.

PROXIMITY AND FIELD SEARCHING

Finally, many databases and some WWW search engines allow the enhancement of searching using Proximity searching – that is, we only want to retrieve a result where the terms are:

a) **both** in the target document; and,
b) are co-located within a certain distance (eg within the same sentence).

A similar capability is Field Searching. Not so commonly available in WWW search engines, Field Searching is generally commonplace in online database search engines. Field Searching allows the specification of a limitation on the specific fields for which the search is to be performed, for example *only* on the Title, or *only* within the Subject fields.

TRUNCATION AND STEMMING

Truncation and Stemming allow the searching of partial roots of a word. In our example above, we may wish to search on "RAMPANT" as well as "RAMPAGING". Truncation would allow a search on all root forms of the word, for example RAMP* to retrieve "RAMPAGE" "RAMPAGES", "RAMPAGING", etc. Stemming is a more structured system which will automatically resolve such related words based on a known dictionary of stems.

PLANNING SEARCH STRATEGIES

It is quite apparent from the search examples discussed above the difficulties of contriving effective searches without forethought of the inter-relationships of the concepts being searched. Generally, several stages are followed in the evolution of a search strategy:

- formulation of the research question or issue
- separation of the discrete concepts to be searched
- construction of effective keywords, terms and synonyms
- preparation of the search logic

This process may be iterated, and should always be informed by the information discovered in the process of searching. As a search is conducted, more appropriate terminology will often be revealed which should be recycled into the iterative search process. Serendipity - the discovery of relevant ideas incidentally in the process of searching - is one of the pleasures of the search process.

WWW SEARCH ENGINES

On the WWW, the way in which Content Authors commonly communicate information to WWW search engines is through Meta Data. Meta Data elements generally appear within the <head> element of web pages, and contain information on the author, title,content owner, subject and other descriptive information. For example, the following header is a simple *Dublin Core* implementation with information on the Page title, keywords, and ownership.

```
<TITLE>MontagePlus - </TITLE>
<META NAME="keywords" CONTENT="Montage Plus, projects, schools, teachers, students, learning, British
Council, Australia">
<META NAME="copyright" CONTENT="&copy; The British Council of Australia ">
<META NAME="author" CONTENT="British Council">
<META NAME="update" CONTENT="2000">
<META NAME="DC.Title" CONTENT="Montage Projects">
<META NAME="DC.Subject" CONTENT="montage plus, montage, projects, schools linkup,">
<META NAME="DC.Creator" CONTENT="British Council, The">
<META NAME="DC.Source" CONTENT="The British Council, Australia">
<META NAME="DC.Language" CONTENT="English">
<META NAME="DC.Publisher" CONTENT="The British Council">
<META NAME="DC.Coverage" CONTENT="Australia">
<META NAME="DC.Coverage" CONTENT="MontagePlus - Projects">
```

Some search engines on the WWW generally have *spidering* engines which *crawl* the web extracting meta data information in order to populate their search databases, and by this means populate their database of web pages. That is, they have a program which:

- Step 1: retrieves a given web page automatically
- Step 2: extracts ever hyperlink from the web page (*spidering)*

- Step 3: follows those hyperlinks to their targets, repeating the processes of Step 1 and 2 (*crawling).*

Characteristically search engines are able to map about 10% of website content in this manner. Other search engines are design to search the content of other search engines – these are known as "Meta Search Engines" (engines that search engines).

Other WWW search engines have a higher degree of moderation for registration of content, and are Subject or Category specific (eg Looksmart and others). The following is a brief review of commonly available search engines, illustrative of their capabilities:

Boolean Capabilities and Constraints (and, or, nesting)

Northern Light, AltaVista Advanced, HotBot, iWon, NBCi, Excite, MSN Search, Magellan, WebCrawler

Boolean Capabilities and Constraints (not)

Northern Light, HotBot, iWon, NBCi, Excite, MSN Search, Magellan, WebCrawler, AltaVista Advanced, Excite

Implied Boolean (+, -):

Northern Light, AltaVista Simple, HotBot, iWon, NBCi, Google, Excite, MSN Search, Lycos, Magellan, WebCrawler, Fast

Proximity

Phrase Search: Northern Light, AltaVista, HotBot, NBCi, iWon, Excite, MSN Search, Lycos, Magellan, WebCrawler, Fast
NEAR: AltaVista Advanced

Truncation & Stemming:

Truncation: AltaVista, Northern Light, HotBot, NBCi, iWon, MSN Search
Beginning Truncation: HotBot, NBCi, iWon
Automatic Truncation: Yahoo!
Single Character Truncation: Northern Light, iWon
Automatic Plural/Singular: Northern Light
Word Stemming: HotBot, NBCi, iWon, Anzwers, MSN Search

Field searching

title: AltaVista, Northern Light, Fast Advanced Search, HotBot, iWon, NBCi, Lycos Pro, MSN Web Search
intitle: Google
allintitle: Google
url: AltaVista, Northern Light, Fast Advanced Search, Lycos Pro
inurl: Google
allinurl: Google
link: AltaVista, Google, iWon, Fast Advanced Search, Lycos Pro
host: AltaVista
domain: HotBot, iWon, NBCi, MSN Web Search
site: Google
anchor: AltaVista, Fast Advanced Search
image: AltaVista
related: Google

others: AltaVista, Northern Light, HotBot

Z39.50

Z39.50 is a search technology that has evolved from the Library community. It considerably extends the search constructs evident in web search engines in providing:

- persistent searches
- searches based on formal classification ontologies

It is widely used within the library community to provide a common interface for searching across disparate, distributed, library catalogues.

TEXT DATA MINING

While digital libraries are the platform that seeks to provide a vehicle for effective and coherent organisation of multimedia information, existing textual and content resources provide a rich source for textual mining of data. There has been an dramatic increase in the range and quality of tools which are available for textual analysis of data, both from the language sense of analysis of morphemes and the audio elements of sound, and in the statistical, heuristic and clustering analysis of texts to extract new insights form the textual material. Just as genes have an embedded historical message for those exploring the genetic history of organisms, so also text, its structure and content, provide many clues and insights not only in the understanding of meaning, but also the history of that meaning. Text data mining is also discussed in Chapter 8.

INTELLIGENT AGENTS

An intelligent agent is a lightweight software component, usually operating autonomously, which acts in a "data gathering role" on behalf of a human user. That is, it uses complex rules for reasoning based on user criteria and feedback to search and gather information, and to monitor and alert on relevant information, on behalf of the user. The key differential element in Intelligent Agents is the ability to learn and evolve more complex methods of data gathering. Such agents may be characterised by:

Single Agent Learning: a single system which improves its performance through its own feedback cycles
Multi-agent Learning: a cluster of systems which improve their collective performance through collaborative information sharing

FEDERATED SEARCHING

The term "federated search", like many Information Technology labels, can encompass both broad concepts of cross database searching and specific marketing labels applied by search engine vendors. The terminology around federated searching is complicated by the recent patent granted to WebFeat for their approach to "federated searching'.

Broadly, federated searching could be taken to cover a variety of approaches have emerged over time to tackle the problem of searching heterogeneous document resource collections. Terms such as metasearching, broadcast searching, and more recently federated searching describe processes for launching searches across multiple document collections. The generally accepted approach to federated searching is the use of a single search interface that searches across multiple databases and yields a single integrated search result(Mah & Stranack, 2005). Meta-searching is exemplified by services such as Google and CiteSeer where the search engine extract extensive metadata, often by spidering other databases, collate and index this information, and manage searching and fulfilment of requests themselves. Search engines capable of federated searching include Ultraseek.

While it is tempting to look at the traditional OPAC and database query as a simple "search" and "result" cause and effect, there is a great deal more that happens behind the scenes. When looking at the integration of search facilities across multiple databases, it is worth examining in more detail the functional components of the "federated search".

Metadata collation

(1) Collection of the search metadata (for example, spidering all web pages in a single site or scanning a database table in order to structure the information gathered for categorisation and searching.

Search activity

(2) Query interface presentation (commonly either a single search box or a more advanced presentation of complex criteria for searches). This also entails the semantic interpretation of the search interface by the search participant and their formulation of a query.

(3) Acceptance of the search participant's query and semantic analysis, including potentially:

   a. Analysis of the database resources applicable to the search

   b. Search parsing (interpreting the search request and preparing this request for presentation to the search engine or engines from which results will be yielded.

(4) Search initiation:

   a. Database connectivity and search presentation either to an aggregated meta data resource or to various database resources

(5) Result collation & presentation (either as an integrated list of search results or presentation on a database-by-database basis).

Search metadata collection

(6) The collection of search and download history for purposes of billing, end user customisation and subsequent search engine tuning and enhancement.

The process of metadata collection, either in terms of the OPAC (a library Online Public Access Catalogue) contents, union list or aggregation of metadata from other systems can be the most important element of the search engine design. Cross-database searching may collate high-level metadata on the available database resources and possibly thematic information on their holdings and the semantic structure of their searches. Google, ultraseek and other database engines are grounded in the spidering of metadata from external resources. (Jascó, 2006)
Google Scholar takes this process further by indexing large volumes of original full text resources and collating information from other content aggregators such as IEEE. Another approach is through Open URL integration from database services through to library catalogues and full text provides (Kesselman, 2006). Pub Med such a service in the Health domain.

As with metadata collation, search presentation can vary according to the level of metadata aggregation, including:

- Presentation of search results fully in the context of each database provider
- Presentation of an integrated list of search results (possibly with duplicated entries removed)
- Presentation of clustered, integrated, search results
- Direct linking to full text resources from the search results.

CONVERGENCE AND SYSTEMS THINKING

The management of multimedia resources shows early signs of technological convergence in the management of information resources. Qualitative Data Analysis systems help to explore concepts and ideas derived from non-quantitative surveys, focus groups, interviews, brainstorming sessions and other free-from information sessions. Document Management systems are evolving in to "Knowledge" systems and business organizations use them to capture their "Knowledge Capital". Similarly, Information Retrieval systems become focal information databases that use analytical techniques to cluster and re-present information in a heuristically advanced search. All these are the early signs of convergence the architecting of information search and delivery platforms that are considerably more advanced than those currently at hand.

Search engines are present in many technologies, including WWW search engines, Online databases, Word Processing Packages, PC file find functions, Qualitative Data analysis packages and so on. There is considerable convergence in these technologies, and companies such as Microsoft have been at pains to embed a common Search Engine technology across their Operating System and Application systems.

Such changes can actually change the very behaviour of researchers themselves, both positively and negatively.

There are dangers: there is a behaviorally attractive motivator in researching and downloading information entirely within the digital domain. Clearly, where not even

a fraction of content is as yet delivered in digital form, the failure to continue drawing on *and reading* content as distinct from its synthesized form has implications both in the depth and diversity of thinking and research. Ruhleder (1995) discusses how this changes the very relationship of the literary critic to the text:

> "The text is a key resource for literary critics, and for all subjects within classics, including archaeology. It is an artefact with emotional meaning for classicists; they speak easily of a love for books…But to talk about 'the text' is deceptive; any scholar wiring with ancient Greek and Roman writings knows there is no such thing as a definitive text."
> •Ruhleder 1995, p.183

Open access movement & Copyright

A further tension lies in the subscription to and management of electronic documents. Electronic subscriptions are now commonly more than 50% of the holdings of journals in a library. The digital rights over these documents and ownership are different from those associated with the physical printed assets, and depending on the licensing terms the library may have rights to search and retrieve journal articles while they maintain a current subscription with the supplier. Should they cease the subscription rights to access these journals cease. This is a move away for the historic role of libraries as a long-term archive repository of these resources. The centralization of ownership of publishing houses and electronic journal repositories has seen a reaction against the progressive commercialization of the academic publish process and the growing limitations on digital rights to published resources. The "open access" movement is the most significant counter-trend to the electronic digital rights limitations of current journal publishing. In the case of "open access" either the author makes pre-publication copy available in an institutional repository, or in the case of over 2000 scholarly journals the journal publish makes the full journal freely availability on the public internet, permitting any users to read, download, copy, distribute, print, search, or link to the full texts of these articles without financial, legal, or technical barriers. Some commercial journal publishers make journals available for access after an "embargo" period of several months to a year.

The area of electronic collection building is therefore complicated by two polar extremes:

- The "open access" movement for free access to journal publishing output
- Increasing Digital Rights Management controls that limit distribution of electronic resources.

See Chapter 9 for a further discussion of copyright issues.

QUESTIONS

1. Describe the relationship between specificity and Exhaustivity. How do they relate to precision and recall?

2. What are the merits and problems in the use of Internet Search engines?

3. What is the purpose of federated searching?

FURTHER READING

BALNAVES, E. "Systematic Approaches to Long Term Digital Collection Management ". Literary and Linguistic Computing. Sept 16, 2005

BALNAVES, E (2007). Integrating Federated Searching in an Inter-Library Loans System. Information Online 2007.

BIEBER et al. (1997) Fourth Generation hypermedia: some missing links for the World Wide Web. *In* International Journal of Human-Computer Studies, Vol 47, 1997, pp.31-65.

COX, John (1997). The changing economic model of scholarly publishing: uncertainty, complexity and multimedia serials. *In* Library Acquisitions, Practice & Theory, vol 22, no 2, pp.161-166.

JASCÓ, P. (2006). Options for Presenting Search Results: Part 1 - Common Options. Online Information Review, 29(3), 311-319.

KESSELMAN, M. (2006). Google Scholartm and Libraries: Point/Counterpoint. Reference Services Review, 33(4), 380.

LIBRARY OF CONGRESS NETWORK DEVELOPMENT & MARC STANDARDS OFFICE. International Standard Maintenance Agency. Z39.50. http://lcweb.loc.gov/z3950/agency/

NATIONAL INFORMATION STANDARDS ORGANISATION. Z39.50 Resource Page. http://www.niso.org/z3950.html

MAH, C., & STRANACK, K. (2005). Dbwiz: Open Source Federated Searching for Libraries. Library Hi Tech, 23(4), 490-503.

MARCHIONINI, Gary, Plaisant , Catherine and Komlodi, Anita (1998) Interfaces and tools for the library of congress national digital library program, *Information Processing & Management*, Volume 34, Issue 5, September 1998, Pages 535-555

OAI. (2004). The Open Archives Initiative Protocol for Metadata Harvesting. Retrieved 18/8/2006, 2006, http://www.openarchives.org/OAI/openarchivesprotocol.html

RICE, R.E. AND MCCREADIE, M (1999). Trends in Analyzing access to information. *In* Information Processing and Management, vol 35, 1999, pp.45-76.

RUHLEDER, Karen (1995). "Pulling down" books vs. "pulling up" files: textual databanks and the changing culture of classical scholarship." *In* The Cultures of Computing, edited by Susan Leigh Star. Oxford, Blackwell, 1995.

SADEH. (2004). The Challenge of Metasearching. New Library World, 105 (1198/1199), 104-111.

WEISLOGEL,Judy. (1999) Science Digital Libraries Symposium II: a conference report, *Library Collections, Acquisitions, and Technical*

WWW SERCHING GUIDES

COHEN, Laura (2000). ***Searching the Internet: Recommended Sites and Search Techniques***. September 21, 2000. http://www.albany.edu/library/internet/search.html

COHEN, Laura (2000). ***Conducting Research on the Internet***. August 2000. http://www.albany.edu/library/internet/research.html

COHEN, Laura (2000). ***Quick Reference Guide to Search Engine Syntax***. August 3, 2000. http://www.albany.edu/library/internet/syntax.html

COHEN, Laura (2000). ***Second Generation Searching on the Web***. August 31, 2000. http://www.albany.edu/library/internet/second.html

COURTOIS, Martin P. and Michael W. Berry (2000). "Results Ranking in Web Search Engines." ***Online*** 23.3 (May 1999). Also available http://www.onlineinc.com/onlinemag/OL1999/courtois5.html

ELKORDY, Angela (2000). ***Web Searching, Sleuthing and Sifting***.February 18, 2000. http://www.thelearningsite.net/cyberlibrarian/searching/ismain.html

FLANAGAN, Debbie (2000). ***Web Search Strategies***. October 15, 2000. http://home.sprintmail.com/~debflanagan/main.html

GRESHAM, Keith (1998). "Surfing with a Purpose." ***Educom Review*** 33.5 (Sept./Oct. 1998): 22-29. Also available: http://www.educause.edu/ir/library/html/erm9851.html

HOCK, Randolph (2000). "Web Search Engines: (More) Features and Commands." ***Online*** 24.3 (May/June 2000): 17-24+.

KRIESEL, Ronald (2000) W. ***Suggested Internet Research Strategies.*** August 19, 2000. http://www.concentric.net/~Rkriesel/Search/Strategies.shtml

SULLIVAN, Danny (2000). ***Search Engine Watch***. http://searchenginewatch.com/

TURNBULL, I (2000). "The Endeavour Project". LASIE March 2000

Chapter 7: Qualitative Research Methods

Statistical and Qualitative research methods are part of the basic research toolkit. Quantitative methods are applied to the analysis of population data, controlled trials, surveys census taking, econometrics, ratings analysis and many other areas. Quantitative methods are equally pervasive in opinion surveys, focus groups, Delphi groups, Action research and other forms of measuring and modelling relationships where the information sources are narrative or loosely defined. This Chapter will explore aspects of Qualitative research as they affect the Arts and Humanities, with a particular focus on Audience Research.

There are many motivations to undertake research. They may vary from a desire to explain or understand existing phenomena. The research may be oriented to making money, or for personal satisfaction. It may seek to answer questions as yet unexplained. It may aim to draw conclusions from large population sets of data, or to generalise conclusions based on hypotheses. This may all be to add to the existing body of knowledge, or to improves our understanding of the world in which we live.

The Movie production industry illustrates the varied ways in which research can be applied. Movie Classifications are reached through focus groups and surveys. Media organisations undertake ratings surveys to measure interest and draw advertising revenue. Movie producers use focus groups pre-test their products in order to market-segment their pre-release advertising.

While such research methods are ubiquitous and effective tools, they are not value free. A profound example of the Ethical boundaries of research is illustrated in the *US FTC Report on the Marketing of Violent Entertainment to Children.* It illustrated the ways in which the Movie Production houses had used children under 17 in Focus Groups to evaluate R-rated movies, with the implication that such focus groups were for the purpose of targeting their productions to audiences strictly not entitled to view the productions (http://www.ftc.gov/opa/2000/09/youthviol.htm).

Quantitative research involves population sampling techniques give the capacity to analyse the ability generalise theories. There are many texts on the most effective approaches to Quantitative research. The mission of *Qualitative Research* is the discovery of new phenomena through careful in depth examination of the results of *non-quantitative* investigation. The scope can be anything from the detailed study of a single case to the textual analysis of large amounts of free-form survey data. Approaches to Qualitative Research include:

- Focus groups
- Case Studies
- Delphi Method
- Content Analysis
- Action Research

Qualitative research in the Arts has utility in evaluating attitudes and opinions, building techniques for textual interpretation, teasing out social trends and directions, and highlighting the subtle interactions and processes that underlie formal systems. *Action*

Research can involve the use of qualitative analysis techniques to explore the detailed system relationships in an organisation.

Case Studies

The in-depth analysis of a particular organisation, situation or environment can highlight possible cause/effect relationships that are not otherwise apparent. They represent Max Webers "typification" - the realisation of generalised models through the detailed understanding of specific cases. They are, of their very nature, open to interpretation, and subjective. A case study may involve re-interpretation of existing data in a new way. Edwin Black recently published a detailed scrutiny of the use of Information Technology by the Nazi Third Reich. IBM and the Holocaust presents a detailed Case study of the use of IBM Hollerinth Punched Card Technology to aid the identification of Jews in Nazi German through analysis of population survey returns. Black's work simultaneously illustrates the ethical issues of research itself (the profound misuse of Census information) and the advantages of the Case Study in highlighting a specific issue:

> "I assembled more than 20,000 pages of documentation from fifty archives…" Black(2000)

Focus groups

Quantitative analysis can be an effective tool for the analysis of specific opinions and issues. However, Focus Groups can be an efficient way of rapid gathering of many different opinions in a relatively short space of time. A selected panel of users discussing issues in an environment controlled by an interviewer, potentially involving a series of iterations on questions, can provide immediate feedback on questions being tested. With an experienced interview, follow-up questions can arise immediately to reveal aspects of a question that have not yet been considered, and in this way key issues can be identified quite early.

The risk of focus groups is the potential domination within a selected group of strong individuals, whose opinion tends to occupy the discussion space. Similarly, interviewer bias can subtly be communicated to the participants. Typically, the Focus Group is useful for gauging consumer reaction, evaluating consumer-purchasing decisions, and measuring the use of products and services. They can be an effective approach to measuring the potential target audience reaction to a proposed idea.

The question design for focus groups yields best results when the target group are taken trough several phases in development of their ideas leading to the central question of questions. Characteristically the focus group will go through four phases:

> Introductory questions – which introduce the broad interest area. Their main purpose is to stimulate the initial discussion among the participants.
>
> Transitional questions. The group should be led through more concrete questions, examples or case studies, which focus the discussion in the interest area.

Key Questions. The key focus group questions are introduced by the moderator when the group has reached a suitable level of discussion and engagement in the interest area. The key interest areas should be directly addressed. Feedback, discussion and the following of interesting aspects of the discussion are a key role of the moderator.

Final Questions. A final series of questions can be used to wrap up the discussion and give a sense of closure, as well as exploring ancillary topics of interest arising from the key questions of the focus group.

Results are gathered from four or more focus groups and these are compiled using a Qualitative Data analysis tool such N5.

The Delphi Method

The Delphi method is an approach to forecasting using expert panels. Like a focus group, discussion and panel sessions are used to elicit opinion and ideas regarding developments that may be on the horizon. This is an iterative process, that may see several groups exchange their ideas as they work to a consensus on key future trends, issues or research directions. In the nature of these panels, very strong facilitation is necessary to avoid an early convergence to consensus or the domination of one individual or theme. In the final round of a Delphi session, questions are often ranked in priority or probability. Such techniques are often a useful approach to formulating options in cases of high uncertainty. The work by Lindstone and Turnoff (1975) presents a comprehensive appraisal of the Delphi approach.

Content Analysis

In many cases researchers already have a rich resource of content available for textual mining. Content analysis looks at trends and occurrences and meanings in such texts. Word frequency, contextual analysis, semantic analysis of texts, clustering and other analysis methods now rely heavily on Information Systems. Software tools such as ATLAS*TI and NUD*IST are particularly strong in methods for content analysis using Grounded Theory. Other packages focus on thesaurus based and probabilistic analysis of texts: Semio Taxonomy and Intelligent Miner for texts being to examples. Hamlet is a software tool that focuses on various techniques for word frequency analysis. Linguistic analysts also have at command a range of software applications focused specifically on lexical analysis: such as *Interlinear Text Processor* and *Shoebox*.

Action Research

Finally, Action Research is an immensely popular method for situationally based research. Rather than attempting to compartmentalise the researcher and the subjects of the research, Action Research assumes the active engagement of the researcher in the problem and its resolution. It is focussed on applied research, and continual refinement. The research process is modelled on the system feedback cycle of:

Problem -> Solution -> Evaluation

Proponents often express dissatisfaction with other research methodologies. True Quality Assurance systems rely on action research for continuous process refinement,

Grounded Theory

A relativist stance on qualitative analysis will assert that the whole process is immersed in a subjective interpretation that will differ for each individual and will project the cultural and socio-political position of the researcher. Qualitative research methods are also challenged by those who deny that it meets standards of normal scientific enquiry: with the low sample rates the analysis and commentary associated with qualitative research may make it merely "journalistic" opinion. There are several approaches to Qualitative Data Analysis that attempt to address such criticism. Those coming from a tradition of Critical theory & hermeneutics positively affirm the interpretive and analytical process of the researcher. This chapter will focus on Grounded Theory and its implementation in *Quantitative Data Analysis Systems* (QDAS) such as NUD*IST as a middle way. Grounded Theory (from the Positivist tradition) establishes a process for the undertaking the research process which borrows the disciplines of the hypothetico-deductive research model. The research process will often include the formulation of a hypothesis. The collection of the source material for analysis may involve elements of random population sampling. Integral, however, to Grounded Theory is the construction of a framework of analysis that exposes each stage that the researcher has taken in the analysis. Characteristically, the researcher will follow the following stages:

Initial reading and analysis. In the first stages of qualitative data analysis the researcher will undertake an initial reading and "open coding" process. This involves attributing subjects and making annotations to the text as it is ready. Software products like ATLAS/TI and NUD*IST have integrated functions to enhance this markup process.

Progressively more complex coding. Periodic reviews with other researchers should be undertaken to begin forming a coherent terminological framework for coding based on results of the initial reading and coding process. At this stage the research is attempting to organise and classify the terms consistently so that the next stages of coding can be used more effectively to reveal the messages implicit in the research material.

Axial Coding is now used to build relationships between classifications and codes. The beginnings of a classification structure should begin to be evident.

> Some classification structures may be hierarchical. Axial coding may bring different classifications together to form new conceptual linkages. QDAS is integral to the effective research and documentation process. Software can considerably enhance the process of coding and researching the material

The coding process itself has risks. Chief among them is reification – that is, codes and classifications, having been established, begin to take on meanings of their own which may reflect the assumptions and background of the coder rather than any meaning or message implicit in the source. The benefit of the Grounded Theory approach is the degree to which the coding process is fully transparent and exposed to all researchers who wish to scrutinise the source coding.

MICROMODELING AND DATA MINING

The Soviets in the now defunct Union of Soviet Socialist Republics had a fascination with micro-economic modelling, to the degree that they attempted to build economic models which could predict the entire production requirements across the breadth of their economy. The soviets built huge matrices to describe their economic situation; however:

> "The fact that the Soviets themselves, having access to their statistical data, vehemently debated over their growth rates makes it clear that the "true" numbers never existed, and the corpse was missing in mystery of Soviet economic growth. Appropriate assumptions and interpretations were debated instead of substance since conventional growth accounting allowed a broad spectrum of arbitrary, but seemingly plausible interpretations for the Soviet economic model. In other words, one could imagine choosing from a menu of Soviet growth rates on the basis of one's own taste." *Kushnirsky (1998) p.59*

This has not dampened the economists desire to model all aspects of society and behaviour, as manifest in the pervasive presence of economic micro and macro forecasts in all aspects of the political horizon. Such a focus on complex micro-economic models has driven considerable software development of micro-modelling systems. This effort has made a considerable contribution to the domain of Microsimulation. The attraction, and advantage, of microsimulation is that it endeavours to incorporate into models theories of individual behaviour and to reflect the heterogeneity of information sources for a model. The process requires a strong underlying data set of information to which is applied behavioural and other mappings to create a system predictive modelling based on conditional probability analysis of the source data. Such quantitative models are the playground of Econometric modellers.

Statistical modelling techniques, married with complex data mining algorithms, provide valuable tools for those researching and analysing peoples behaviour and interest. This process, applied with appropriate ethical limitations on the preservation of the privacy of individuals, can be advantages to those who are attempting to provide the greatest clarity and effectiveness in communication to their target audience.

AUDIENCE RESEARCH

Fundamental to Politicians and Marketing Analysts has been the use of techniques for data mining and survey analysis to build an understanding of their constituents and

clients. Once again, the technological convergence of intelligence gathering systems (such as surveys and data mining tools) with econometric modelling tools has delivered powerful systems for interpreting the wants and desires of constituents and clients. These tools should not be neglected in building systems in the Arts and Humanities. When we discussed Systems Thinking, we explored the issues of feedback cycles and the importance of feedback as a lever for change. Audience Research and Qualitative Analysis tools are an important element of such an understanding.

INTERNET AUDIENCE RESEARCH

The Qualitative research methods discussed earlier in this Chapter are directly applicable to Internet Audience research. Online surveys should not be regarded as the only means of gathering information on users of a website. The toolset for Internet Audience analysis should include:

Data Mining website usage statistics
Audience Research and Measurement through

Data Mining Internet Usage Statistics

The key tool available to most web sites is log analysis of site access activity. Website logs are gathered as the user interacts with the website. Characteristically they retain the web page visited, the source URL for the website visit, sometimes session information to show continuity of access of the user through the site, time and date visited, and other user details that can be gleaned from the client browser interaction.

A key problem in the data mining of website logs derives from the very nature of web servers. Web servers are "stateless" in their basic form: that is, the protocol for user interaction, HTTP (Hyper Text Transport Protocol), blindly responds to each inbound web page request as a new and independent interaction. It knows nothing about a specific "customer", "session" or "client'. It simply sees a web page request and responds to it. This dilemma is addressed generally through one of three techniques:

1. The use of cookies. The web browser sends information with each interaction with the web site regarding itself.. The web server can request the browser to create and retain cookies, this establishing a continuity of information between interactions with the website. Cookies used in this way are generally called **Sessions**.

2. The use of URL query strings. URL query strings themselves can contain detailed information which themselves are retained in the usage log of the web server. With very consistent use of URL query strings, the functional equivalent of Cookie-based session states can be achieved.

3. Embedded JavaScript & Java Applet. A small JavaScript or java applet function triggers an interaction with a non-cached server which is counting site visits. This can be one of the most functionally effective ways of capturing and tracking user interaction. It is deployed at the risk of slowing user interaction,

and is the most vulnerable of the three techniques to security and proxy server limitations.

Whichever of these techniques is applied, their essential purpose is to track *visits* to a web site as distinct from *hits.* A website *hit* is generally regarded as a single interaction with an HTML page or website script. A visitor to a website may visit characteristically 4 pages on the site – these 4 pages viewed by the user would be counted as 4 hits. This would however be counted as 1 visit by the user. Hits are useful for tracking volume of usage and measuring the popularity of specific pages on a site. Visits are an effective measure of user interest.

Many software tools are available for log file analysis. Analog, for instance, is an open source log analysis product producing graphic log analysis results in a variety of different formats (http://www.statslab.cam.ac.uk/~sret1/analog/).

The session "cookie" depends on the user being willing to accept cookies, but is otherwise an effective means of tracking the activity of a user through a site. The second approach, through URL query strings, is vulnerable to loss of session state when the browser ends or when the user follows paths off the site, but has the advantage of requiring no client-side information storage such as a "cookie".

Cookies are now the generally accepted mechanism for identifying and tracking user movement through a site. The analysis of site usage of Internet Content is framed around two concepts:

> "Hit", or a discrete page activation by a browser client of any component of a page (including possibly images); and,
> "Visits", or a sequence of page movements through a site all belonging to the same person.
> "Requests", or a "whole" page request, as distinct from the various "hits" that might comprise a page and all its component parts.

By-and-large, the industry has moved toward measuring "Visits" and "Requests" as distinct from "Hits", except when measuring raw volume of access to a site. Web servers maintain activity logs which track the individual page movements on the server.

Web caching servers, however, can defeat the most effective tracking mechanism. The issue of web caching serves is less significant in dynamic websites, because a large proportion of the content is generated "on the fly" and may not be effectively cached.

Dynamic serving of content (where the content is regenerated specific to each separate request, and possibly personalised to the person making the request) can be used to avoid the difficulties of tracking user activity caused by caching servers. However, the negation of the caching process is at the expense of the speed of the user experience – a factor that will remain important while network bandwidths are generally poor for most users. Dynamic content delivery can also be more expensive in the resource overheads required of the Content Server. Not all information sites require or benefit for the end-user customisation and personalisation that is possible through cookie-based dynamic

websites. Indeed, the focus on privacy policies itself shows a limit to which end-users which to have their usage patterns known and analysed in too much detail.

However, the review of usage statistics of a website should not be the only tool deployed. Focus groups can provide a useful tool where the website population can be engaged in such a process.

Online voting and survey systems

Online voting systems remain a nascent technology. While their use is popular on websites, the actual merits of the information gathered is questionable. There are several approaches to online voting, with very different cost profiles in implementation:

Controlled environment voting uses a fully secured infrastructure (physical access, networking control) for the population engaged in the vote. This is a high-cost approach to vote collection. It does, however, give full control on the authentication of participants and the validation of their vote. Such voting systems are becoming popular in Corporate AGM's and other time and location constrained voting events.

Secure Server Approaches in an online, public access environment are becoming popular as a means of controlling aspects of the voting process. Public Key Encryption combined with some form of User Authentication can go some way toward addressing the issues of user identification in order to prevent duplicate or fraudulent voting. Such approaches are discussed further in Chapter 9.

Online polling systems are relatively easy to implement, but fail many of the basic standards for design of survey instruments. A key weakness of online survey delivery is the inability to apply normal selection techniques for those participating in the survey. The difficulty of auditing repeat submissions also diminishes their value as a method of information gathering. Taken purely as an indicative barometer of opinion they have some value. When such surveys become a factor in decision making, such structural weaknesses are very significant. Few countries have implemented online voting systems because of the difficulty of host-positive validation of the identity of the voter in order to prevent false votes and the prevention of duplicate votes. Controlled environment approaches are being used for electoral voting – these use a variety of pre-voting identity checking and physical logging of votes taken. A further issue of electronic voting for electoral purposes is the difficulty of auditing voting results so that "scrutineers" from both sides can be certain that no electronic tampering has occurred.

Privacy

The collection and use of private information has become the subject of active legislative engagement both at the national and international level. The *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data* illustrates key issues that all content providers will necessarily have to address in designing systems which collect information on individuals. Key issues relating to privacy treatment of individual information include rights of access to review and correct information, obligatory notification of breaches of security and the

limitations and retained for the minimum necessary time. Sanctions for breaches of such regulations are being extended to companies, and are becoming an element in international trade restrictions. In this context, a *Site Privacy* policy will soon cease to be an optional extra to online web-based information delivery and for online survey collection. Website users are becoming progressively more critical of sites that do not clearly identify the uses of information supplied. Furthermore, such privacy policies will also be constrained by national and international legislative requirements.

CASE STUDY 1: QUALITATIVE DATA ANALYSIS

The following is a detailed analysis included with permission of the British Council Australia:

a) an example of qualitative analysis using focus groups
b) statistical analysis of website information

SUITE 18
20-24 GIBBS STREET
(PO BOX 214)
MIRANDA NSW 2228
AUSTRALIA

		International:
PHONE:	(02) 9525 3200	**61 - 2 - 9525 3200**
FAX:	(02) 9525 3656	**61 - 2 - 9525 3656**

MARKET ATTITUDE RESEARCH SERVICES PTY LTD
INCORPORATED IN NSW
ACN 058 339 245

20 DECEMBER 2002

EXECUTIVE SUMMARY REPORT:
OUTCOMES FROM FOCUS GROUP EVALUATION OF THE RECENTLY DEVELOPED BRITISH COUNCIL,
AUSTRALIA INTERNET WEB-SITE - DECEMBER 2002

The British Council recently re-developed and launched its internet web-site covering issues relevant to the British Council, Australia and international presence (sourced from the UK) of the British Council.

In late 2002, Market Attitude Research Services Pty Ltd (MARS) was asked to evaluate this recent re-development of the British Council site. This evaluation (conducted during December 2002) involved MARS conducting three (3) focus group discussions with young, tertiary educated professionals involved in science, business, education, or the Arts. Several of the participants were also alumni of UK universities. Each focus group comprised a mix of these different young adult target audiences. In early 2001, MARS conducted a similar project with Australian partner stakeholders of the British Council.

The three focus groups conducted during December 2002 involved:

- two (2) focus groups conducted in special focus group rooms where **broadband internet connection** was available; and
- one (1) focus group conducted at the MARS office where **standard telephone line internet access** was only available.

The purpose of using different styles of internet access was to assess the useability of the British Council web-site under both broadband and standard telephone line access.

The two broadband focus groups were conducted with 12 young, tertiary educated professionals drawn from science, arts, alumni and business executive backgrounds. The standard telephone line access focus group comprised five participants. The newly developed British Council web-site was intensively investigated, examined and reviewed during the focus group sessions (each of which lasted 1 1/2 hours).

Key findings which emerged from the testing evaluation are presented below…

YOUNG ADULT, TERTIARY EDUCATED PROFESSIONALS ASSESSMENT OF THE RECENTLY DEVELOPED BRITISH COUNCIL AUSTRALIA WEB-SITE: DECEMBER 2002

↓

KEY FINDING ONE: The recently developed British Council internet web-site performed successfully, and was rated by young adult, tertiary educated professionals as better than the earlier site.

A majority of the participants had previously in 2001/2002 visited the British Council web-site (before its recent current development) and **rated the new site much more favourably than the earlier site.** Illustrative comments are expressed below on this outcome…

"This new British Council site is really up to date and I love the graphics. It is better than the previous British Council site I visited".

(Recently graduated student from an Australian university looking for postgraduate opportunities in the UK)

"This new British Council web-site has much better information and content compared to what I saw earlier this year when I visited the site".

(Young business professional, 25-30 years, alumni of a UK university)

"Compared to the previous British Council site this new site is more pleasing. It is more dynamic, has pleasing colours and changing areas and text which is pleasing to the eye. It also has cute music and graphics in the Studying in the UK section. There are a lot of things on the site which draw your interest".

(Typical comment expressed by young tertiary educated professionals and alumni in arts, science and business using broadband access to the site)

Based on the expressed comments the conclusion to draw is that participants rated the new site very favourably. **Key successful features were that the site was regarded as "modern and up to date", it was "attractive due to the way text, colour, images and photographs were mixed together without being aggressive or too animated" (as some sites are), the site contained "quite a deal of interesting information which could be explored", and the text was clearly presented in a conversational style which was easy to read.** Illustrative comments on these outcomes are shown below…

"This British Council site is aesthetically pleasing. It is a fantastic design and has simple, easily understood, conversational style"

(Australian student with a non-English speaking background applying for university entry in the UK)

"I am positive about this site. I was able to find what I wanted, and the information was easy to understand".

(Undergraduate student applying for transfer to a UK university)

"I am very satisfied. It is an interesting site and there is lots to look at. There is also a good Calendar of Events".

(Science journalist, young tertiary educated professional)

"I enjoyed the bright, attractive colours being used in the site. There were also very good moving graphics being used".

(Young tertiary educated accountant professional, aged 25-30 years)

"The new British Council site was asthetic and tactile. It worked well and was quick. It gave information you were trying to seek and it was easy to use. It also seems to be regularly updated. It seems different from a lot of other sites. It was soft to look at, and not harsh like some sites. The site has soft pastel colours and spacing within the texts which make it easy to read".

(Typical comment expressed by young tertiary educated professionals and alumni in arts, science and business using broadband access to the site)

"I am satisfied. The site had a lot of information and an excellent level of detail. However, if you are at home using a computer on a telephone line it is not the sort of site you go for fun. You would visit the site to seek information about the UK and study opportunities".

(Typical comment expressed by young tertiary educated professionals and alumni in science and business using standard telephone line access)

Another interesting feature enjoyed by users were how animations and images changed on the Home Page and in other areas where the content listing changed order. However, for users accessing the site via standard telephone lines, it was slower at the site to see and experience these animations and images (and this problem reduced enjoyment of the site).

A new design feature of "drop down style navigation" was employed in the site and this feature was highly rated as a very useful feature to help move through the site. An illustrative comment expressed on this feature is shown below…

"This is a very good feature to help find information"

(Student applying for entry to a UK university)

"I am very satisfied. I really liked the drop down navigation. You can reach a page very quickly which you are trying to reach"
(Typical comment expressed by young tertiary educated professionals and alumni in arts, science and business using broadband access to the site)

Finally, when given several questions which required the site to be explored to find answers to the questions many of the participants were quickly able to navigate through the site and find the answer (although intuitive guessing in some cases allowed this outcome to happen quickly, however, if an intuitive guess was not used it was difficult in some cases for participants to quickly find the answer).

KEY FINDING TWO: Although the recently developed British Council web-site was highly rated, suggestions were made for some design enhancement.

A key overall conclusion was that the new British Council site was highly rated by young tertiary educated professionals. However, several design issues were found which will require further attention. These design issues included:

- **the inability to return back to the Home Page directly once into the site (this was annoying to the participants because they had to keep clicking pages to get back to the Home Page)…**

  "It would be very useful to have a direct route to allow users to access quickly other parts of the site, especially to quickly come back from the detailed areas of the British Council site back to the Home Page".
  (Australian young tertiary educated professional with a non-English speaking background, Australian alumni of a UK university)

  "Going back to the Home Page is difficult. It is not clear what you have to do"
  (Young tertiary educated professional, aged 25-30 years, using standard telephone access to the British Council web-site)

  "There is a need to provide a separate quick hotlink back to the Home Page. The Home Page is slow to click back to reach. It would be difficult for a less experienced user of the site to visit the site for a casual browse".
  (Alumni of a UK university, young tertiary educated business professional, 25-30 years)

  "You have to keep clicking and clicking to get back to the Home Page. It is annoying, you cannot click on the "Home" button to get back to the Home Page".
  (Typical comment expressed by young tertiary educated professionals and alumni in arts, science and business using broadband access to the site)

- **the Home Page was actually around 1 3/4 pages in length which meant that scrolling was required to fully read the Home Page (and this annoyed the participants)…**

  "The Front Page is not actually one page. It is actually up to two pages and you have to scroll through it. This is a poor design"
  (Typical comment by male and female young, tertiary educated professionals using standard telephone access to the site)

  "The Home Page is more than one page and you have to scroll. This is annoying".
  (Typical comment expressed by male and female young, tertiary educated professionals using broadband access to the site)

- **the end of the Home Page contained useful and interesting information (such as publicising new events) and reading about this was easily missed if the reader did not scroll** (and MARS observations of user behaviour identified that some people did not scroll through to the end of the Home Page).

KEY FINDING THREE: Images created by the recently developed British Council web-site suggested to the young, tertiary educated professionals that the UK is a vibrant, modern, contemporary country.

The young, tertiary educated participants perceived that the newly developed British Council web-site projected the UK as being contemporary, modern and vibrant…

"The colours, graphics, changes in site design are satisfying. The section "Destination UK" is wonderful!"
(Student considering applying for university study in the UK)

"The site does show a vibrant image of England and the UK. The site is colourful, has white space separating the text blocks, and a good use of animation and images".
(25-30 years, research scientist, alumni of UK university)

"Yes! It does show a modern England. The graphics are modern and colourful"
(25-30 years, professional in the Arts)

"The British Council site is showing a modern image of the UK. It really is colourful and up to date. The presentation is not flat. It is lively and this encourages you to visit and study in the UK"
(Typical comment expressed by young tertiary educated professionals and alumni in arts, science, and business using broadband access to the site)

Conclusions and Recommendations

The conclusions to draw from the evaluation are that the re-developed British Council web-site is regarded as an enjoyable and attractive site to visit and experience, and the language used and style of communication is easy, conversational, friendly and informal. The young tertiary educated professionals enjoyed these features of the site and perceived that the UK was modern and contemporary through images projected by the site.

There is an enormous range of information on the site and this aspect of the British Council web-site is highly liked, and the new drop down navigation feature allows easy and fast navigation from the Home Page to specific areas. However, it was very difficult for site users to return back to the Home Page, and to be certain they were actually at the Home Page section if they were moving back from quite a number of pages within the site.

The main issue for the British Council to address is to undertake further design changes to enhance the internet web-site in the following ways:

- to develop a technology solution which allows users of the internet web-site to move directly back to the Home Page without requiring multiple clicking on the "Back Page" item and requiring them to go through several pages to reach the Home Page;
- to, if possible, to make the Home Page one page rather than 1-2 pages which currently require the reader to scroll through the "page" to find information (particularly useful information about events and activities the British Council is presenting in Australia). In other words, it would be useful to limit the Home Page to one page and more aggressively "market" the activities of the British Council in Australia, and other countries where the British Council has a presence; and
- to ensure that the animations and images contained in the site do not become too difficult to use, enjoy and experience by people who only have standard
- telephone line access to the site.

CASE STUDY 2: QUANTITATIVE ANALYSIS OF A WEBSITE

The follow analysis is an example of a detailed quantitative analysis of a website based on web traffic and other quantitative data available.

Site Profile

The Australian presence of the British Council comprises two domains:

britishcouncil.org.au – the principle gateway to the Australian office, including the principle education gateway management by the office. Principle gateways in this site are **arts, education,** and **alumni.**

mindsetuk.com – representing the science funding events and resources gateway.

Statistical resources used

This report was compiled from the following sources:

Financial Data - supplied by the BCA office comprising the staff costs for support of the websites.

Website usage data - using detailed statistical analysis of the raw usage logs for 2002 in order to measure website traffic and usage. The arts section of the site is now publishing from UK servers and is not represented in statistics show here after October of this year.

The full analysis reports can be viewed at:

http://www.britishcouncil.org.au/stats/special.htm and

http://www.britishcouncil.org.au/stats/special2.htm.

References are also made to prior year reports available at:

http://www.britishocuncil.org.au/stats/bcusage.htm

Content management activity – activity statistics measured in the Content Management System used to manage these sites: Inter-Publish. The figures in this report do not include additional editing/publishing activity undertaken in ObTree for maintenance of the home page and the arts section of the site.

Content Management Statistics

The following is the content profile of the two primary domains (other than Montage) managed by the British Council Australia as at end December 2002:

	Web Pages	Images	% Content Changed in 2002
www.britishcouncil.org.au	4337	11631 (gif) 673 (jpg)	73%
www.mindsetuk.com	655	105(gif) 65 (jpg)	100%
Total	4992		

Inter-Publish maintains an audit trail of changes made to websites. As an additional measure of the value of investment in the websites the following is an indication of activity

Month	Documents changed in this month at least once	Total edits this month
Jan 2002	285	499
Feb	3723 (site makeover)	3996
Mar	209	299
Apr	493	651
May	272	390
Jun	252	331

Jul	162	696
Aug	271	2303
Sep	324	1801
Oct	536	2832
Nov	553	3243
Dec	498	1433
Total	**7578**	**18474**

Note: these figures represent actual editing work (as distinct from automatically assembled and syndicated content).

On 7578 content edits representing on 18474 content items are performed for these sites. This represents a content turnover of approximately 14.5% for the site on average each month. This performance represents a substantial achievement in terms of the information currency of the website, given the high information-value of pages on the websites.

In addition to the manual editing of web pages, a substantially larger number of pages are frequently updated as a result of scheduled publishing and syndicating activities of the Content Management System Inter-Publish.

Taken over the full year of 2002 over 73% of the site content was modified.

These figures are indicative that the office maintains I high level of currency in the websites that they manage.

Site Usage Statistics

During the period 1 January 2002 to 21 December 2002 the site had 3,887,432 hits representing 127,883 distinct visits to the site.

Number of hits	3,887,432
Number of requests	502,483
Number of visits	127,833
Average number of requests per visit	3.93
Average visit duration	00:10:05

This contrasts with 2001:

Number of hits	3,208,452
Number of requests	298,329
Number of visits	86,777
Average number of requests per visit	3.44
Average visit duration	01:00:03

The most popular sub-site is Education, but all sites have experienced steady growth in usage. Unique site visits have increased by 48% on 2001, although the duration of visits has considerably reduced. The logging levels and statistical analysis tools are identical over the period, so this would appear to be a trend in user activity: more users, visiting about the on average more pages in a briefer amount of time.

The following are usage statistics for each principle subsite:

Arts
Visits: 10,271
Average Visits per day: 28.99

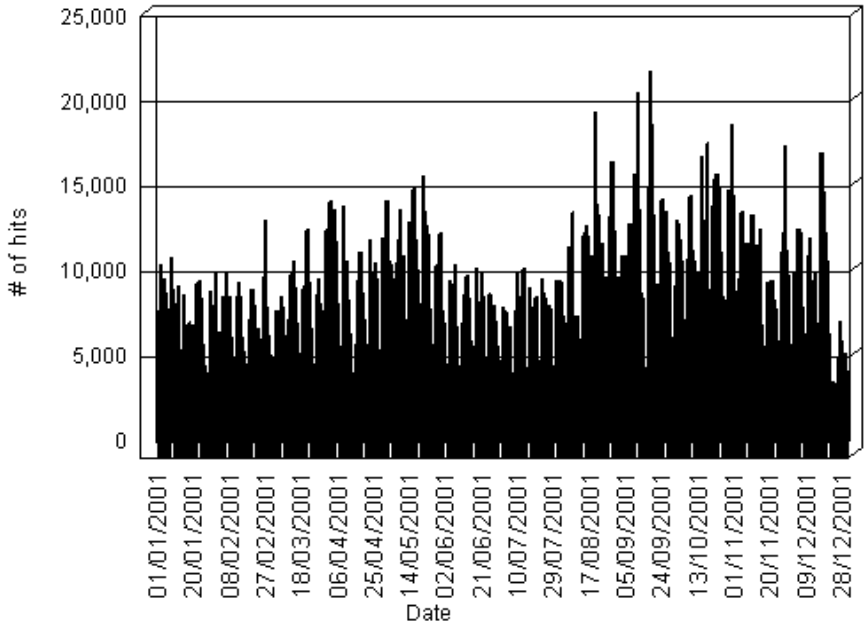
Alumni
Visits: 9,054
Average per day: 25.55

Education
Visits: 65,038
Average per day: 183.56

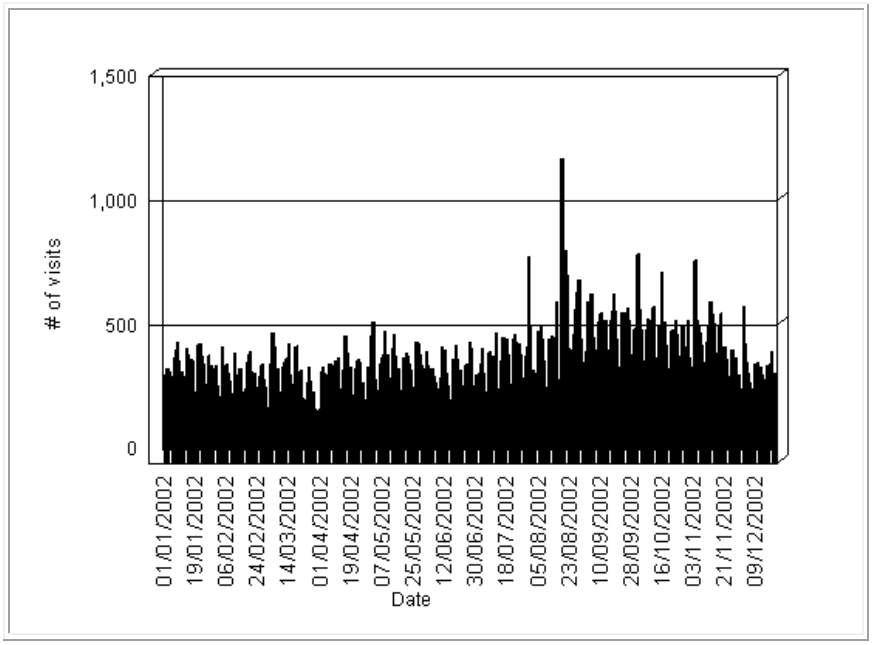
Science
Visits: 13,702
Average per day: 68.07

Site Trends

There is seasonality evident in site usage, and in particular, the demand increases significantly in the second half of the year, from July through to November, declining in December. This probably correlating to the usage of the education site, and to planning for the forthcoming year. The following graphs show site usage trends over the last two years.



Visits for 2001

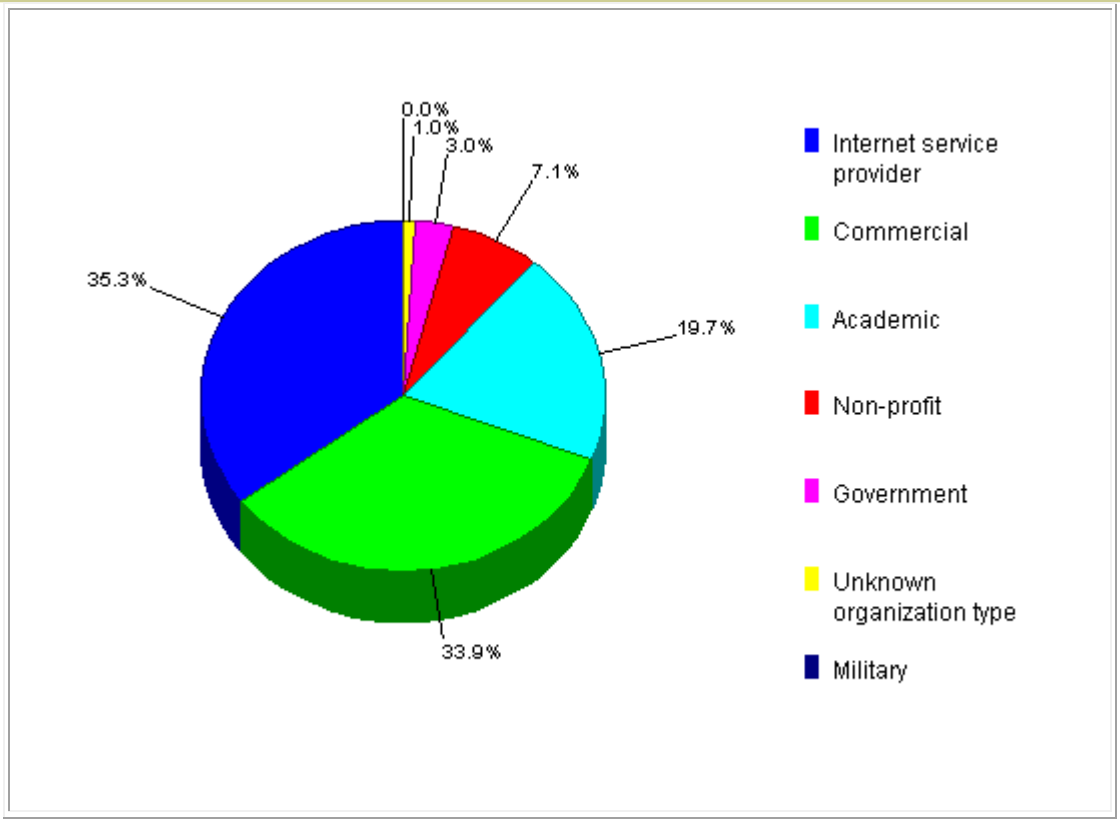


Visits for 2002

Organisations

Over one third of visits are from Educational institutions, and these institutions feature highly in the top organisational sources.

International requests by organization type



	Organization name	Number of requests	% of Int'l requests
1.	OPTUSNET.COM.AU	23,291	10.59%
2.	BIGPOND.NET.AU	16,313	7.41%
3.	nexnet.net.au	13,857	6.30%
4.	TMNS.NET.AU	11,910	5.41%
5.	IPRIMUS.NET.AU	9,400	4.27%
6.	USYD.EDU.AU	4,189	1.90%
7.	DAV.NET.AU	3,750	1.70%
8.	POL.CO.UK	3,342	1.52%
9.	CONNECT.COM.AU	3,036	1.38%
10.	IHUG.COM.AU	2,927	1.33%
11.	UNIMELB.EDU.AU	2,902	1.32%
12.	comindico.com.au	2,750	1.25%
13.	UNSW.EDU.AU	2,743	1.25%

Content Usage

The education site remains consistently the most popular part of the site. The scholarship search introduced mid-year has grown to be a popular resource on the site. 50 of the top 100 most used pages of the site are in the Education section of the site.

In other sections of the site, the following are the most significant entry and activity points of the site:

Arts : the Home Page and events pages appear to be the most visited parts of the arts site.

Mindset: the Home Page, events and /ukscience pages (also a news/events page) appear to be the most popular pages on the Mindset site. The funding search in Mindset does not appear to be as popular as the equivalent scholarship search in education.

Alumni: the Alumni site has only recently been launched. Focal points of usage again relate to events and the member contact/search page.

The first visit pages of the site once again highlight the importance of the education section of the site. It is clear that many users are bookmarking directly to the parts of the education site, especially scholarships. Other key entry points are:

/about/faq.htm
/about/faq.htm
/whatson/ (events)
/ukscience/ (science news and events)
/links/ (science links)

… and of course the home pages of Alumni, MindsetUK and the BCA site itself.

Top 10% of first requests

Lists the top first pages that users see when coming to your site. The data only reflects links from other sites or URLs entered by a user.

	First request document title	# of visits	% of visits
1.	/	33,198	25.97%
2.	/education/	8,078	6.32%
3.	/robots.txt	5,872	4.59%
4.	/stylesheets/british_council.css	2,207	1.73%
5.	/education/scholarship.htm	2,152	1.68%
6.	/alumni/	1,998	1.56%
7.	/education/Newteaching.htm	1,814	1.42%
8.	/about/faq.htm	1,426	1.12%
9.	/education/scholarships.idq	1,329	1.04%
10.	/education/News.htm	1,322	1.03%
11.	/education/InfoWebSites.htm	1,314	1.03%
12.	/whatson/	1,307	1.02%
13.	/education/InfoLearnExams.htm	1,037	0.81%
14.	/contact/faq.htm	963	0.75%
15.	/sites/	943	0.74%
16.	/education/schol_law_New.htm	865	0.68%
17.	/education/Newaccountancy.htm	842	0.66%
18.	/links/	837	0.65%
19.	/default.ida	798	0.62%

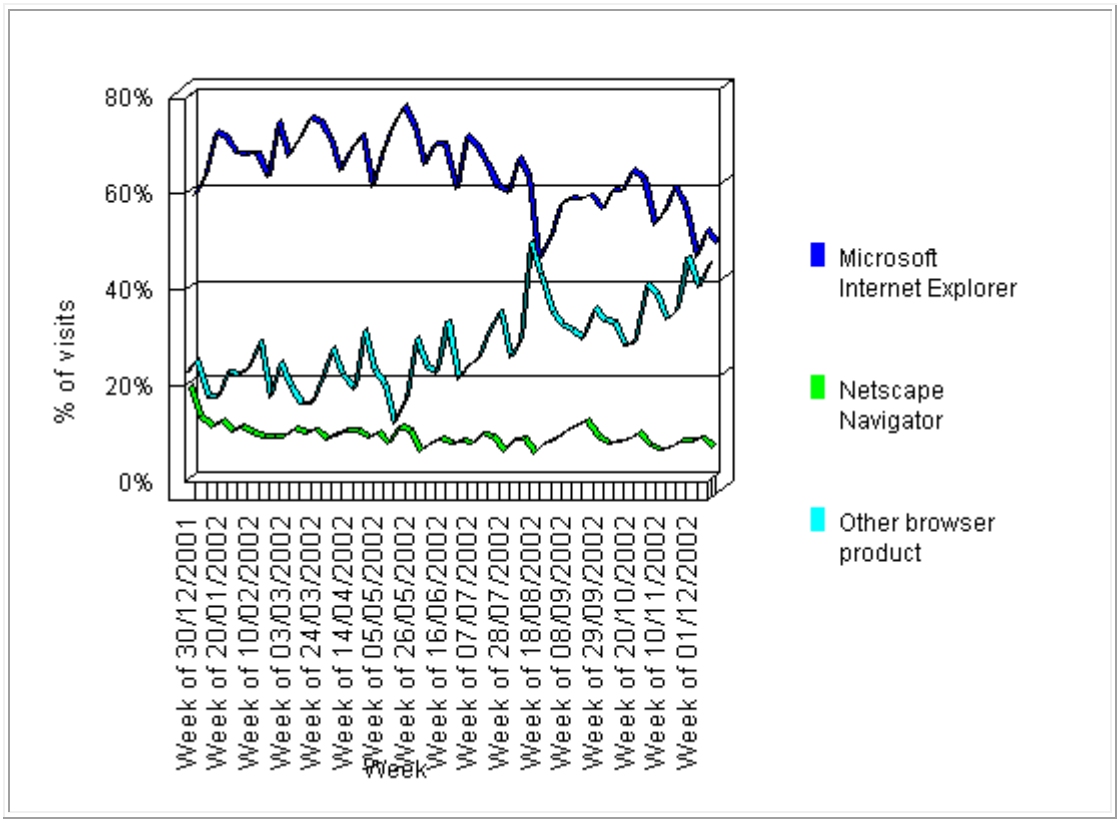
20.	/education/Newoptometry.htm	741	0.58%
21.	/education/Newpharmacy.htm	731	0.57%
22.	/about/	716	0.56%
23.	/education/Newdentistry.htm	711	0.56%
24.	/education/Schol_mba_New.htm	704	0.55%
25.	/education/summerschool.htm	688	0.54%
26.	/education/InfoTeaching.htm	674	0.53%
27.	/education/NewcaterHosp.htm	665	0.52%
28.	/arts/flash/	654	0.51%
29.	/education/UCASApplication.htm	642	0.50%
30.	/education/AusDifferences.htm	590	0.46%

Browser platform

There has been a steady decline of Netscape as browser of choice. In a trend apparent in all major sites, Netscape has declined from a 20% share to a 5% share of the browser presence. Within Internet Explorer, version 5 and above predominate, with earlier versions delining to less than 1%. Windows platforms similarly represent 95% of the site accesses.

Browser market share

Shows the percentage of visits to your site using Netscape and Microsoft browsers over the analysis period.



Financial Analysis

The total recurrent costs for managing these websites is £37,437, representing:

Networking costs and Server Administration: 41%
Web Server Content Management: 59% or £22,241 per annum

The costs of running the site, expressed in terms of hits, visits and uses is summarised below:

British Council Sites

	Hits	Requests	Unique visits
Totals	3,887,432	502,483	127,833
Costs	< £0.01	£0.07	£0.29

Expressed as a cost per content page produced these costs are:

	Pages Edited/Changed	Cost per Page
Cost per page Edited	18474	£1.20
Cost per page Published	7478	£2.93

This expresses both the effectiveness of the office in content creation and the efficiencies gained through effective use of a Content Management System.

Conclusion

This report presents a detailed financial, content and usage analysis of the British Council Australia websites. The following are the key findings of the report:

1. Website usage has grown steadily, now reaching 3,887,432 (up 22% on 2001). Note however, future measures of this growth will depend on availability of logs from the new UK server, with arts and home page hits now captured off a separate server.

2. Visitors are viewing more pages but staying for a briefer time. This may be a phenomenon apparent more widely than at the British Council alone.

3. There is distinct seasonality in Education sub-site, with heavier demand in the latter half of the year, especially July through November. This seasonality is not as apparent in Arts, Alumni and Science sub-sites.

4. The Education second of the website is the section in highest demand, illustrated not only by the popularity of the pages but also the clear book-marking of sub-pages in this site as first entry points to the British Council Site.

5. The education scholarship search has been highly popular and successful.

6. The event content in Arts, Alumni and science appear to be the principle points of attraction in these sites. There may be some opportunity to gain synergy between education pages and related arts events/pages.

7. Cross-browser compatibility issues are on the decline with Internet Explorer representing 95% of browser usage of the site.

8. The cost of delivery of web content is < £0.01 per page served and about £0.29 per unique visit.

9. The cost of content management is £1.20 per page edited (excluding fixed costs), and .

There is a high level of currency of information on the site, with a turnover of content of approximately 14.5% per month and an overall turnover of 73% of the site content in the full year.

SYSTEMS THINKING

Web caching, and unreliable session technologies mean that audience research of access to a site is far from a "science". Audience analysis is best supported by a mix of quantitative and qualitative methods for researching audience opinion and preferences. Online polling is still an emergent technology that requires better universal methods for effective identification of the voter but can provide an effective technology for registered users of existing services. Electronic electoral voting systems are emerging, mostly in a controlled environment setting.

QUESTIONS

1. Contrast the uses of Qualitative Data Analysis and Quantitative Data Analysis in the Audience Analysis.
2. Describe the stages for moderating a Focus Group.
3. What is the role of the Privacy Policy on a website?
4. Discuss issues associated with collection of electronic voles in a General Election

FURTHER READING

BASKERVILLE RL, W.-H.A.T. (1996): A critical perspective on action research as a method for information systems research. *Journal of Information Technology* **11**(3):235-46.

BLACK, Edwin. (2000). IBM and the Holocaust: The Strategic Alliance between Nazi Germany and America's Most Powerful Corporation. NY, Little Brown.

BOVE, V.M., JR., DAKSS, J., CHALOM, E. and AGAMANOLIS, S. (2000): Hyperlinked television research at the MIT Media Laboratory. *IBM Systems Journal* **39**(3-4):470-8.

CHATZOGLOU, P.D. (1997): Use of methodologies: An empirical analysis of their impact on the economics of the development process. *European Journal of Information Systems* **6**(4):256-270.

DENZON, Norman K and Lincoln, Yvonna S. Handbook of Qualitative Research. Sage. 2000

KELLE, Udo. Computer-Aided Qualitative Data Analysis: theory, methods and Practice. London: Sage, 1995.

KUSHNIRSKY (1998), Fyodor I. Changing Methodology of Measuring Growth and Inflation: The Case of Ukraine. *Comparative Economic Studies*, Winter 1998 v40 i4 p59(1)

LINSTONE, H & TUROFF, M. The Delphi Method: techniques and applications. Reading, Mass: Addison-Wesley Publications, 1975

MYERS, M.D. (1997a): Critical ethnography in information systems.:276-300.

MYERS, M.D. (1997b): Qualitative research in information systems. *MIS Quarterly* **21**(2):241-2.

PALVIA, S. and HUNTER, M.G. (1996): Information systems development: A conceptual model and a comparison of methods used in Singapore, USA and Europe. *Journal of Global Information Management* **4**(3):5-16.

Chapter 8: In the Field

This chapter will explores application of Arts Informatics technology in the context of the specific disciplines of Archaeology, Religion, Music, Art, History and Philosophy. Common to all is exploration of Information Systems in a way that enhances and at times transforms the fieldwork and creative activities in each discipline.

LANGUAGE AND TEXT

We explored in Chapter 2 and 3 various aspects of text representation. An issue fundamental to such markup systems is the nature of language itself. Language can comprise a heterogonous mix of Dialects, Language Groups, local variants of language, and variances between written language forms and local spoken variances. Information Systems have dealt with this issue in limited ways. Most information systems are limited in their language capabilities to the limitations of the Operating System itself. Operating systems have in many cases defined a limited number of language codes, and have used a single language code to describe several aspects of the language – for instance in Microsoft Windows the *langid* may be used to describe OS sort orders, code pages, spelling and other aspects of the language treatment. It is called *Overloading*. Where there is a limited language range (Windows allows 512 languages) it may not be possible to reflect different cultural usage as much as linguistic difference – such as:

Italian Italy Italian Switzerland

Internal language definitions on computers do not necessarily equate to pure linguistic definitions.

Internet Browser rendering of languages is generally based on the International Standard ISO 639-2, which provides codes for languages & groups of languages. It too has inconsistent differentiation between groupings of languages and discrete languages

> "…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." (SIL p.9)

Is nevertheless widely used and is the basis for most HTML/XML language definitions. ISO 639-2 3-character code theoretically supports 17,500 codes, but implementation as a "mnemonic" in practice supports only several hundred. This is a problem in the context of least 6,700 language variants known internationally (http://www.sil.org/ethnologue/). The proper rendering of languages in an Information System will also characteristically depend on the availability of a language *plug-in* which provides the specific information for rendering the language characters in a particular browser and operating system environment. Often historical languages (ancient Chinese, Latin and ancient Greek) will have symbols that will be incorrectly rendered or perhaps cannot be rendered at all.

There are, however, information systems which facilitate the understanding and interpretation of languages. Linguistic: applications include parsing, lemmatizing words and constructing visual models of the language semantics. Database systems can assist in information retrieval of texts, indexing, concordances, word lists, KWIC/KWOC (key-word-in-context, key-word-out of-context).

Lemmatized text brings together irregular variants of a word or verb: for instance words am, are, and is, would appear as *be*, and the words car, cars, car's and cars'

would appear as *car*. A lemmatised sentence can help the translating and interpretive process, and is also useful in the integration of concordances and dictionaries of pronunciation. For instance, the phrase *the boy's cars are different colours* could appear in a lemmatized text as *the boy car be different colour*. In languages other than English, this process would involve similar, though not exactly the same, principles of reduction.

Text Parsing involves the analysis of a free-form sentence and the analysis of its structure for verbs, nouns, adjectives, etc.
Content analysis involves the detailed parsing of text for conceptual structures.

Aspects of content analysis can include:

qualitative: looking for regularities and differences in text, exploring the whole text (QDAS - qualitative data analysis software) per NUD*IST
event data: analysis of events in textual data
quantitative: analyse the text selectively to test hypotheses and draw statistical inferences. Output is a data matrix that represents the numerical results of the coding.

A review of software products in the field will illustrate the uses of information systems in Linguistic analysis.

KEDS & TABARI – facilitates the coding and distribution of international event data in real time based on content analysis of defined sources. Over time this feeds into crisis analysis systems. The Kansas Event Data System (KEDS) is a program for the machine coding of international event data using pattern recognition and simple grammatical parsing. It is designed to work with short news articles such as those found in wire service reports or chronologies. KEDS codes particular events from Reuters wire service lead sentences but in principle it can be used for other event coding schemes. SPAD-T analyses texts of automatically by associating numerically coded information. Comparisons of texts are done with probabilistic type and methods. SPAD-T counts words and word sequences (phrases) using sort order tables and exclusion criteria like length or frequency. Using probabilistic methods characteristic words, word sequences, or sentences are found. Comparisons of the vocabularies of texts are performed with different types of factorial analyses and correspondence analyses. Contingency tables of common words or the segments repeated within the texts are also possible. Cluster analyses (hierarchical using reciprocal neighbours) using Ward's method allow e.g. an automatic classification of responses to open ended questions.

Semio Taxanomy 2.0. Linguistic analysis technology and statistical clustering with user-defined vocabulary requirements to create an intuitively browsable structure of categories that provides intelligent access to the global information space within a mass of formerly unstructured text. Important phrases and keywords are extracted from a variety of text sources such as intranet/Internet sites, Lotus Notes, Documentum, ODBC-compliant databases, XML, etc. This process combines language detection, proximity analysis and stemming and normalization rules to produce the cleanest, most informative extraction technology available. Semio's phrase extraction pulls

relevant, informative phrases from within the text. The phrases are attached to a set of categories which can come from a thesaurus, pre-built category set from Semio, or a custom structure of the user's choosing. The category structures can then be validated and modified in an easy, iterative process to ensure quality and consistency.

Intelligent Miner for Text 2.3. The text analysis tools can be used to analyse all types of online documentation, from customer requests and technical reports to newspaper and magazine articles. It handles many document types, and is used to organise and facilitate navigation of the documents. It has automatic annotation features for the summarisation of document contents.

HAMLET. This is a simple tool for Word frequency analysis. It supports individual word frequencies (fi), joint frequencies (fij) for pairs of words (i,j), both expressed in terms of the chosen unit of context, and the corresponding standardised joint frequencies (sij = (fij) / (fi + fj - fij)) are displayed in a similarities matrix, which can be submitted to a simple cluster analysis and multi-dimensional scaling. Multi-dimensional scaling to matrices of joint frequencies derived from a number of texts, using Procrustean Individual Differences Scaling (PINDIS).

TextAnalyst. TextAnalyst is an intelligent text processing tool capable of automated semantic analysis, summarisation, and navigation of unstructured natural language texts. Clustering of documents in your textbase, semantic information retrieval, and focus your text exploration around a certain subject.

Other language tools facilitate Speech analysis & phonetics, Phonology & Morphology, analysis of Syntax & Grammar (parsing software), the building of Lexicons (using parsing and lemmatization), language survey & analysis. An example of integrated language analysis tools is *Shoebox.* This software is a linguistically-oriented freeform database program with facility for producing glossed interlinear text. Up to seven database files can be open at once with instant access to all. Another is *askSam* for more general text analysis. It is a commercial free-form textbase program for DOS and Windows intended for data management, word processing, and text retrieval. It supports unstructured text, structured fields, text and data retrieval, sorting, formatting, report generation, and multiple file access. Of course, integrated Office Productivity suites are also including more complex text analysis tools.

COMPUTATIONAL LINGUISTICS AND TEXT DATA MINING

Finally, complex tools for text comparison assist the detailed analysis of languages and the construction of Generation of concordances & wordlists. These produce annotated interlinear text. *IT (Interlinear Text processor)* for instance, maintains the vertical alignment of the interlinear annotations and stores all word and morpheme annotations in a lexical database, thus enabling semi-automatic glossing.

Empirical computational linguistics computes statistics over large text collections in order to elicit useful patterns which can then address other concerns of natural

language processing, such as part-of-speech tagging, word sense disambiguation, and bilingual dictionary creation.

Emerging from computation linguistics is a new research domain of Text Data Mining. With the emergence of data mining techniques applied to the domain of text corpora, we now see the emergence of and automated text categorization of data. A complex example of such text mining was the ScamSeek project at the School of Information Technology at University of Sydney (http://www.cs.su.oz.au/~lkmrl/scamseek.htm).

SYSTEM THINKING

The diversity of information systems for managing text, content analysis and linguistic analysis is itself transforming the scholarship directions in the humanities. Packages such as NUD*IST used for qualitative data analysis illustrate a new trend in the integration of commentary with original texts, and the construction of complex information systems for the analysis of all aspects of texts. A work will comprise not only the scholarly publication but also the ongoing scholarly annotations, comments and discourse. Researchers can progressively annotate and "enrich" the document itself, and the context of the document can enhanced through clustering and association techniques. This is the "bibliography" writ large!

ARCHAEOLOGY

The last two decades have seen a revolution in the technology used in archaeology. The technological tools available to archaeologists in the field have considerably advanced their capacity to analyse a site physically and temporally, and to project the visualisation of their work in new ways. This section will give a brief overview of the systems available to archaeologies in their fieldwork, before exploring in more detail the Arts Informatics Analytical and Representational systems that are currently in use.

FIELDWORK AND ANALYTICAL SYSTEMS

The technology available in the field for archaeological teams has dramatically advanced the archaeologist's capability in the analysis across the layers of time that are represented by a site or a region.

The prelude to new approaches in detection of sites of archaeological significance came in the extensive use of ariel photography during World War II for military purposes. Photographers, conducting military reconnaissance missions were also discovering sites of archaeological interest.

> “The first known aerial photographs of an archaeological site were taken from a war balloon by Lieutenant P. H. Sharpe in the early 1900s. The target was Stonehenge. In World War I, photographers conducting military reconnaissance flights kept running across sites of archaeological interest. It wasn't long before military officers began actively seeking out such sites on their own. One pioneer was Lieutenant-Colonel G. A. Beazeley, who discovered the extensive outlines of ancient canals in Mesopotamia's Tigris-Euphrates plain. But as useful as

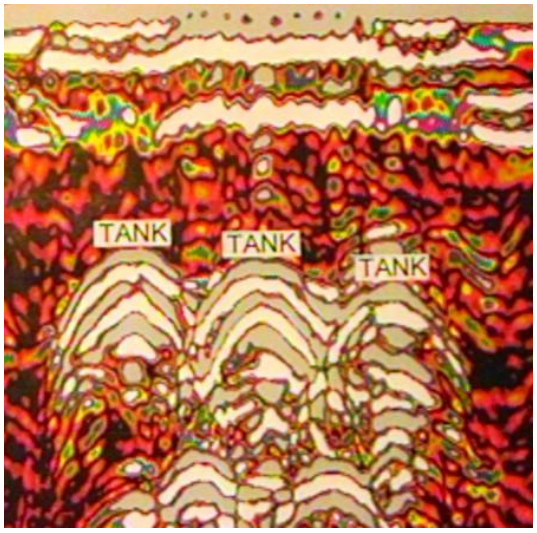
> aerial photographs are, they have their limitations: namely, airplanes can fly only so high and human eyes can see only so much."
> (http://www.pbs.org/wgbh/nova/ubar/tools/index.html)

Technology has assisted since the 1980's in enhancing the human eye with the additional technologies of Aerial Infrared Photography, Thermographic Infrared Multispectral Scanner (TIMS), Imaging Radar and other approaches. Infrared photography detects light spectrums beyond those visible to the naked eye, and in particular can detect sources of heat, including vegetation.

In 1972 the first LandSat satellites bought multi-spectrum technology to play from orbit, vastly extending geographic mapping capability and the ability of archaeologists to see beyond the visible spectrum. The launch of the SIR-A satellite in 1981 went even further in the use of imaging radar to penetrate areas that were otherwise very difficult to map, such as forests, or the layers below sand dunes.

> "Archaeologists didn't appreciate the full potential of space imaging until 1981, when NASA launched an imaging system called SIR-A on the Space Shuttle. Unlike Landsat 1, which used reflected sunlight to make an image, SIR-A sent out its own radar signal and then "listened" to the echo. Archaeologist Farouk El-Baz, now director of the Center for Remote Sensing at Boston University, had asked NASA to fly SIR-A over the eastern Sahara desert, hoping it could make sense of the anomalous rock formations he had been studying there. No one was quite prepared for the images that came back.
>
> The Sahara is the driest place on earth right now, but SIR-A was able to penetrate the sand and reveal an ancient landscape below that, amazingly, had been carved by running water. "The Sahara once looked like the landscape of Europe," El-Baz reported, "with rivers, lakes, mountains and valleys." The banks of the old rivers beds, dubbed "radar rivers" by researchers, turned out to be excellent sites for archaeological excavation, yielding a bevy of Palaeolithic tools and artefacts." (WGBH Educational Foundation. http://www.pbs.org/wgbh/nova/ubar/tools/index.html, 2001)

Closer to the ground, non-invasive tests are available which can reveal important archaeological information. Soil Resistivity maps can show places where the soil has been disturbed, such as filled holes, trenches, and ditches will collect more moisture than surrounding soil and will show up on the test as having low resistivity.
Ground Penetrating Radar and the *magnetometer* go further to revealing the sub-strata of a site. A Manetometer detects the variances in the earths magnetic field created by objects buried underground or in different ground layer densities.



(http://www.geomodel.com/tanks.htm)

Sonar technology also provides a means for non-invasive investigation through use of *Geophysical Diffraction Tomography.* A GDT system uses sharp noise blasts to cause sound wave echoes which are then detected by microphones sunk down bore holes. Focus on Site Preservation as led to additional technologies for maintenance of artefacts, such as use of the Electron Microscope and Fluron Light Test.

Using such field technology to investigate a site has been combined with *Global Positioning Systems (GPS)* and *Geographic Information Systems (GIS)* to provide detailed mapping of a site. The combination of information systems to support detailed mapping of a site and GPS to assist with highly accurate reconstruction of individual surveys into a whole-of-site map transforms capability to analyse the results from new technologies. Key uses of GIS systems in the field are:

- Detailed mapping of the site, physically and temporally through accurate Geo-referencing (made possible by GPS)
- Layering of site information (location of roads, streams, etc overlayed with information on site finds)
- As an aid to predicting the location of sites by using historical information regarding to scan for likely matching characteristics

Statistical analysis tools are also a fundamental part of the analytical toolkit in the field. The use of statistical analysis packages provides an effective means of collating large quantities of data together to identify patters.

REPRESENTATIONAL TECHNOLOGIES

GEOGRAPHIC INFORMATION SYSTEMS

The evolution of VRML (Virtual Reality Modelling Languages) and representation technologies has come at a time when GIS systems have begun to provide detailed and accurate plots of archaeological information in a special and temporal framework. The result has been a revolution in the representation and delivery of archaeological information.

Scanning and 3D scanning of objects in conjunction with Computer Aided Design technologies (CAD) has enabled effective Artefact interpretation and replication. Similarly, the use of GIS in conjunction with text and object databases has expanded the opportunities for visualisation of archaeological models and databases.

GIS AND TIME SERIES DATA

In the very nature of analysing and presenting information that covers diverse periods of time, the analysis of spatially placed information is crucial. In archaeological digs, the Geographical Information Systems (GIS) serves a role in both the data gathering and data analysis activities of the archaeologist. Collection of information at the data gathering phase (for instance, at a dig) can enhance the preservation of important information regarding exact placement of artefacts. This exact placement is enhanced through the use of Global Positioning Systems (GPS). The collation of this information within a GIS facilitates the subsequent analysis of the overall results of the dig: the GIS acts as an effective information resources not only on the progress and detail of the archaeological activity but also for the detailed analysis of the results.

The *time series* representation of information becomes vital, and this is an area of significant contribution of Archaeology to GIS systems. The ability to layer historical information, and to represent in GIS form the progression in time as well as place of artefacts has many applications. In particular, the combination of GIS, GPS and layering of time series information allows the effective integration of different sources (such as historical maps, surveys) and current GIS information collection.

Software such as *MapEdit* is now part of the toolkit of the Archaeologist. Representation of information in a GIS system with a time or special element present additional challenges.

CONTENT PRESERVATION

Representational systems have bought alive archaeology in the public mind. They provide a means by which a large amount of underlying analysis and speculative work can be presented as a more concrete representation of the original look and feel of a site. Challenges abound in the collation and management of underlying data coming from both field activities and diverse sources of information:

> "The amount of investment in developing such a system that would allow us to use archaeological data in its native formats (relational databases and vector drawings) is high, especially since there is little standardisation in the use of software packages by archaeological projects." Vince & Nevelle (1997)

The scientific focus of archaeology has come into considerable conflict with the capabilities for public information delivery that are now at hand. In 1993 Luc Genevriez released simulations of the Abbey of Cluny, an important medieval monastery near Paris. There was considerable debate following this release over the level of interpretation implicit in such visualisations:

> "There were tremendous battles among the experts over the site. The resulting film stirred up a dreadful row…" Boukhari (2000)

Such debates are equally prevalent in the museum community, though they have not dampened the drive to presenting visual modelling in museum exhibitions. The issue of authenticity is profound. The visual representation of historical information can be very convincing in its detail – so much so that the many assumptions that underlie its construction may not be evident. Impressive visualisations may be backed by equally impressive research – or they may be impressionistic conceptualisations that are not anchored in a concrete chain of reasoning and research.

The background research to a visualisation is vital as are the assumptions that went into its construction. The growth in *Virtual Museums* hosted by reputable organisations with properly curated content may provide a counter-weight to the easy with which fanciful projections and ideas can be given convincing imagery.
A challenge to those building Archaeological Information Systems, particularly when delivering this content to the public, is degree to which the final representation integrates not only the creative and reconstructive elements that it presents but also the conveys the assumptions, weaknesses and contradictions that are entailed in the representation. With public enthusiasm inspired by vivid integration of reconstruction work in movies such as *Gladiator* the boundaries between impressionism and realism are important to maintain.

Archaeological work in many instances involves the partial or complete destruction of the site at which work has been undertaken: either as part of the process of extracting the artefacts, or because the archaeological teams are working in advance of other building work. Many of the detailed resources resulting from such archaeological work is retained in printed archives in Museums, Universities and Archaeological foundations. Information Systems play an increasing role not only in the initial research work but also the permanent recording process. A clear advantage of this process is the better dissemination of these information resources: this is apparent in the evolution of archaeological information clearing houses such as ECAI (Electronic Clearinghouse for Archaeology Initiative – www.ecai.org). Issues of obsolescence are equally apparent, as has been discussed earlier in this book. The duty of the archaeologist may lie not only in the collection and analysis, but also the choice of suitable technologies for the long term preservation of the information collected. Content Management Systems can play a role not only as a vehicle for the dissemination of the information to current audiences but also to the future.

CONTENT DISSEMINATION

Virtual Museums have an important role in enhancing the immediacy and visual clarity of archaeological information. Both as an integral part of existing exhibitions and as an online public resource, the design approaches for information delivery are important to its effectiveness. In many cases, considerable re-work is undertaken in the presentation of information in different exhibitions, online services. The integration of this content and its re-use as information objects will in the long term allow the collection of information resources that will be progressively enhanced through the accretion of commentary. Some sites are exploring just such an approach to integrate the cross-cultural narrative in information presentation.

The use of XML as a vehicle for communication between Archaeological Information Resources is an emerging trend:

> "An appropriate standardized data model is necessary to facilitate electronic publication and analysis of archaeological data on the World Wide Web. A hierarchical "item-based" model is proposed which can be readily implemented as an Extensible Markup Language (XML) tagging scheme that can represent any kind of archaeological data and deliver it in a cross-platform, standardized fashion to any Web browser. This tagging scheme and the data model it implements permit seamless integration and joint querying of archaeological datasets derived from many different sources." Schloen (2001) p. 124

Religion and Critical Analysis

Theologians have drawn extensively on the use of Information Systems to enhance the process of critical analysis of religious texts and to enhance text comparison. Systems for text comparison (eg Dead Sea Scrolls analysis against Jewish and Christian texts) can yield valuable insights in the interpretation of meaning. Of course, Critical analysis is itself a cultural phenomenon – and is not always welcome in all cultural contexts – witness the reception tog Rushdie's "Satanic Verses". Database capture of religious texts also offers new opportunities for the analysis of the full corpus of texts for a given period. In the last century one of the most comprehensive compilations of the works of the early Christian Fathers, the *Patralogia Latina* was published. Its availability in searchable full text form dramatically changes the ability to analyse this text. Such database systems can also allow the greater dissemination of such texts that would otherwise have been possible. *Project Gutenburg* is focussed on formulation of just such a database of full text content.

Libraries such as the Vatican have also undertaken major digitisation programmes in order to provide scholarly access to the original manuscript versions of texts in their Libraries.

Debates regarding the "authority" of texts have lasted several thousand years. Academic research regarding authority of texts is often disjoined from religious and community impressions. Linguistic and content analysis tools of the sort just discussed have considerable utility for the analysis of religious texts. Integration of

this material in a database further transforms the ability to address debates that have lasted more than 2000 years on the authorship of various religious texts. *Authenticity* of meaning is vital. Translation of religious texts in an area with the fidelity of the translation is of absolute importance to religious communities. The use of systems as an aid to the efficient translation, recording and production of key religious texts into different languages is therefore of considerable importance. Finally, the use of language interpretation and analysis tools can give a deeper understanding of original texts.

History

Just as in religion, the historical analysis of texts through database systems is opening new opportunities for critical analysis. Once again, there arises the possibility of completely encoding the entire body of written work arising from a given period. Biographical analysis of historically situated individuals can be greatly enhanced with such resources.

The use of hypermedia to annotate the multicultural discourse surrounding historical events also transforms the communication of historical information. The *Endeavour Project* at the Australian National Library is a project which is publishing the journals of Captain James Cook in the first Pacific voyage (1768-71). These journals record many indigenous Australian and Pacific islander encounters which have profoundly influenced European intellectual and cultural history. The purpose of this digital collection is to integrate the cultural discourse surrounding this encounter in one location (Turnbull, 2000).

Philosophy

In addition to the use of linguistic and database systems to enhance the philosophers commentary, a specific class of software seeks to teach, present and understand formal logic. For example, *The Logic Works* integrates concepts for formal logic learning. The *hypermedia* paradigm is valuable as a way of extending the complex narrative around philosophical argument. *Discussion lists* and other loosely structured communication mechanisms provide an effective framework for free development if concepts. The role of internet related news groups, discussion groups and forums extends the ability to conduct international discourse.

Art and Art History

The common theme of the use of online database systems to enhance the narrative activities of researches is a common theme to Art and Art History.

Content Databases play a crucial role in the organisation and preservation of photographic image collections that have immense cultural value. However, unlike text, image databases need considerable enhancement with meta-data in order to have utility as a searchable resource. Such databases can have unexpected benefits: for example as a research tool, for example, in the recovery of works by Holocaust survivors. They can server to aid the preservation and collection of photographic images and detailed records of art collections

The Kings College Project *Daedalus* is a database of text and images of Greek sculptors through to Hellenistic times. It provides facilities for image/image and image/text comparison, once again in a searchable full-text framework.
The Art Historian also deploys technological innovations as an investigative tool, for detailed comparison of images and research of image content, and for image criticism and interpretation.

Finally, Multimedia systems as a new creative paradigm. The integration of multiple forms of media, and the integration of computer systems with genetic and other research innovations offers the artist new and often bizarre creative opportunities. Certainly we wee art creation entirely in an online, digital environment.

Music

Music present particular difficulties in the integration of music in information systems and databases. The traditional musical score is itself a "high content" information source. The information may include presentation properties (movement of performers).

Music is multi-faceted. The theoretical construct has many possible physical manifestations. A musical score makes not reference to the physical placement of the musicians, for instance

It can be argued that interpretation of musical scores is of the essence. A virtuoso presentation has an element of the genius both of the composer and the performer. The latter can be captured in a specific instance (through digital recording) but cannot be encoded.

Researchers are also interested in encoding *and searching on* of relations between data items in order to derive themes (such as a musical “motif”). Unfortunately, there are no universally accepted common computer-readable method for music representation despite 20 years of research. Database technologies nevertheless can act as research adjuncts such as language analysis, translation systems, content analysis systems.

Several XML approaches have emerged recently, including:

- SMDL – Standard Music Description Language from HyTime
- MML – Music Markup Language
- 4ML - Music and Lyrics Markup Language
- MUSICXML (from humdrum & MuseData)

As in Religious textual analysis; fidelity to the source and authenticity are quite crucial in the digital encoding and rendering of music.

Obsolescence

A common issue to all approaches to implementation of information systems in the Arts and Humanities is obsolescence. Today’s key information is tomorrows research archive. Internet websites are symbolic of the transience of much information delivery. A permanent archive of Australian sites considered of cultural significance

covers little more than 1100 sites (the Pandora project). Yet there is little doubt that many innovators expect the creations to have more than transient longevity. Yet it is very apparent that many systems supporting these endeavours are themselves subject to significant issues of obsolescence. This is the final, and key, consideration in Systems Thinking: the long term preservation of creative endeavours.

QUESTIONS

What is the role of the GIS in archaeology?

Describe three issues in digital management of music.

Discuss changes and opportunities research in the Arts and Humanities with the availability of large content databases.

What is meant by a Virtual Museum?

What is the role of a Content Management System in supporting resources in the Humanities?

FURTHER READING

BOUKHARI , Sophie. (2000). COMPUTERS REBUILD THE PAST. UNESCO Courier March 2000 p40

EISNER AB, Jett QR, Korn HJ. Web-based periodicals as an emerging cultural form: incumbent and newcomer producers in the early stages of industry revolution. International Journal of Electronic Commerce, vol.5, no.2, Winter 2000-2001, pp.75-93. Publisher: M.E. Sharpe, USA.

KOONTZ, John E. (1990). "Shoebox: a review", Newsletter of The Society for the Study of the Indigenous Languages of the Americas 9(2):10-12. [Compares Shoebox v. 1, askSam, and Lexware.]

LANCASHIRE, Ian (ed.). (1992). The Humanities Computing Yearbook 1989-90, 158-159. Oxford: Clarendon Press.

http://www.sil.org/computing/shoebox.html

SCHLOEN J. D. (2001) Archaeological data models and Web publication using XML. Computers & the Humanities, vol.35, no.2, May 2001, pp.123-52. Publisher: Kluwer Academic Publishers, Netherlands.

SOCIETY FOR THE STUDY OF THE INDIGENOUS LANGUAGES OF THE AMERICA (2000) "Interlinear Text Processing, Part 1: IT", Newsletter of The Society for the Study of the Indigenous Languages of the America, 7(4):17-18.

SUMNER T. (2000) Promoting scholarship through design. Information Communication & Society, vol.3, no.4, 2000, pp.597-615. Publisher: Routledge, UK.
TURNBULL, I. (2000) "The Endeavour Project". LASIE March 2000

VINCE, Alan and Nevelle, Sandra Garside (1997)." Publishing multimedia in archaeology". http://intarch.ac.uk/news/eva97.html

Chapter 9: Intellectual Property

Authors put considerable time & effort into their creative and intellectual endeavour. *Intellectual Property Rights* centre on the protection of and reward for effort. Authors may seek such protection to ensure the fidelity of their work – to ensure that it is not altered and changed in ways they do not approve. An author may wish to gain financial reward from those who make use of the product of their creative endeavour, or may wish to ensure that others do not profit exclusively from their effort. Finally, an author may wish to restrict the community of users and the extent of publication or use. *Intellectual Property* is the tangible output of this creative activity. Examples of the outcomes of such intellectual creativity include: books, inventions, software programmes, music, etc.

Intellectual Property laws are not an absolute defence of the rights of the creator: after all, the oppressive enforcement of such rights could grossly limit the development of new ideas and lead to monopolisation of concepts in the arts, business and research. Intellectual Property legislation has evolved over the last two hundred years endeavouring to achieve a balance between the social need to facilitate communication of ideas through a concept of "fair use", and the desire of individual

and corporate authors to protect and control their creations. Excessive protection of rights can limit the free flow of ideas and information. Distain for protection of rights can see stagnation. This balance has been achieved internationally through copyright protection, licensing, trademarks and patents. A time-limited set of rights is granted to the author in exchange for various provisions for access and control of use.

The "Copy Right" was originally a protection for publishers against other publishers. The limited access to content duplication systems (such as printing presses) has historically limited the flagrant copying of creative artefacts without permission. Where it occurred, the culprits were generally large organisations to be identified and sanctioned. The “Internet” has been called the “greatest copying device” ever invented. The ease with which content and systems can be copied, and the mobility of content encourage the activity. This is compounded by the quality of the copy – often identical or sufficiently similar to be indistinguishable. Finally, collaborative sharing systems can be built that transcend normal distribution networks. There are, in the digital arena several very contrary directions in the arena of Intellectual Property:

- The “open source” movement, which epitomises those advocating at the unfettered free flow of ideas and knowledge.
- The “copyright” and “trade mark” directions, which are (very) slowly catching up with the digital age.
- The technological rate of change, which make copying and replication of content quick, easy and low-cost.
- Progressively stricter enforcement of Copyright by large content creators (especially in Music, but also video and software) that actively targets individual infringement for the first time.

This has not stifled technological efforts at protection of content and the rights of the originator. Images and Video are the technologies most amenable to such protection: *Digital Watermarking* methods are approaching a stage of maturity that will allow the embedding of source “meta” information that will enhance the tracking and ownership control of image, animation and video content. Text and Audio content are much harder to control in this manner. Database service providers are not attempting to protect the information content itself, but rather to build systems of such range and depth that they cannot easily be replicated – the *Information System* and access to it provides the means of price access to the content.

Copyright and Software Licensing

The protection of copyright is extended to the specific expression of an idea, program, or design. An advantage of copyright is that it is automatically applicable – generally no formal registration needed, although some formal registration bodies exist. Generally international copyright protection extends only to certain categories of content, including:

- Literary works
- Software
- Artistic works

- Musical works
- Cinematography
- Sound Recording
- Radio & TV broadcasts

Similarly, copyright only applies to content delivered in some “material” form: that is – *published* content. The key benefit of copyright is that it can be treated as a tradeable "product" which can be bought and sold for a limited duration. The duration varies with some countries, but has generally been:

- 50 years from the date of the death of the author, or
- 50 years from the first release of software, or
- 50 years from the first broadcast for radio/TV.

This term was specifically extended in the US to provide protection for the Disney rights over "Mickey Mouse". The US/Australian trade agreement will extend the copyright coverage to 70 years in Australia, joining the "Mickey Mouse" club.

The Digital Millennium Act, and bi-lateral Free-Trade agreements such as between the US and Australia are tending to force standardisation on longer terms and stronger digital copyright provisions emerging in the US.

In traditional print media the right of control has extended only to the first sale of the "copy". The Holy Grail of digital publishers is to gain formal control over not only the primary sale market but also the secondary (or second-hand) market. Ironically, with digital media the client may have fewer rights over the redistribution of an e-book than they would have with a traditional book.

The publisher of digital resources may know a great deal more about their customer population than the traditional publisher, especially when these resources have Information Systems and Networking components. Issues of information privacy can therefore apply in ways previously inconceivable: the publisher potentially can track a great deal more about you and your interests, and exactly what you read, when and where the resource was sourced.

Copyright is extended internationally through the World Intellectual Property Organisation (WIPO). This treaty is in the process of gradual change to address the difficult issues of digital copyright treatment. For instance, the original revised treaty A*rticle 7* covers "…direct & indirect references either permanent or temporary" (Hayes,2000). In an Internet context this would even potentially cover in-RAM references to images or text and the transient movement of this content over modems and network devices.

While this clause was deleted from the signed treaty it illustrates the difficulties of copyright management in an Internet framework. The MP3.COM case of the Recording Industry Association of America case against MP3.COM in 2000 upheld the Internet as a separate "market" for content delivery and the right of the copyright owner to license each specific market separately. This has been used subsequently in the management of broadcast rights of sporting events such as the Sydney Olympics. NBC employed a team of web searchers to seek out sites that infringed their video-streaming rights:

> "The IOC expects some 40 percent of the Games' revenue to come from the sale of exclusive TV rights to about 20 broadcasters around the globe.
> The IOC says that revenue might be at risk if fans could bypass TV and tune in on the Internet, although only 4 million U.S. households, for example, have high-speed access to the Net. Without high-speed access, online viewing of streaming video is unbearable."

Their efforts were, within the specific timeframe of the games, quite successful:

> "The anti-piracy firms hired to police the Internet for unauthorized broadcasts of the Sydney Olympics said Thursday that they have caught some 30 violators since the competition began last week. "

While this expresses the use of copyright as a restrictive force for the protection of the author, it also acts as a positive force in funding new creative endeavours, and attempts to ensure that the financial benefit of a creative endeavour goes to the originator.

In an environment where digital copying technologies are readily accessible, most users are still happy to make a "reasonable" contribution to cover copyright. If the implementation is too onerous or restrictive, users will ultimately seek other ways to achieve their end. The case of GIF as an image format is a good example. GIF became a defacto image standard for web icons in the early 1990's. In 1994 Compuserve announced its intention to license use of the images, based on its copyright.
Soon after, Compuserve was forced by community reaction to back down. A twist on this right came in 1995 when Unisys announced their patent rights on the embedded compression algorithm used in GIF (unbeknownst to Compuserve !) In 1995 Unisys announced a royalty charge on any software that generates a gif image – they targeted they software developers rather than the wider community of image users.

Hyperlaw Inc vs West publishing (US, 2001) illustrates the extent to which copyright protection may extend in unexpected ways. Hyperlaw built a CDROM-based service facilitating access to the West legal database. West argued that the page numbers links their database content were themselves copyright, even if the content itself wasn't. In 2000, Scientologists sued Slashdot.com for publishing the quotations from Scientology texts. The site was forced to "self-censure" – although it continued to link to other sites which did not self-censure.

Changes such as the *1998 Digital Millennium Protection Bill* in the US appear to be progressively resiling from the "fair copying" protections that libraries have traditionally sought for free flow of information. A reaction to this trend has seen three "alternative" models for distribution of creative works which do not directly seek traditional copyright protection – although the protection of all three still fundamentally draws on copyright rights:

- Collaborative software development through "open source"
- "Free" release but with embedded advertising
- "Shareware" - try & buy

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. Linux – and in particular companies like *Red Hat* have shown that such a model can work effectively as a business model. The nine key elements of Open Source are:

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code.
5. No Discrimination Against Persons or Groups.
6. No Discrimination Against Fields of Endeavour.
7. Distribution of License.
8. License Must Not Be Specific to a Product.
9. License Must Not Contaminate Other Software.

There are risks entailed in the use of "Open Source". Unless crafted carefully through the license to distribute, the author may find others attempting to patent their open source code. In the Apple Open Source License the following caveat can be found:

> "This License and the rights granted hereunder will terminate automatically without notice form Apple if You, at any time during the term of this License, commence an action for patent infringement against Apple."

Another popular approach is the "Free" distribution of software with embedded advertising. The user of the software gains the benefit of using the software and the author gains revenue from advertisers rather than directly from the users. Software products such as *Medical Director* in Australia have achieved dominance in the market through such distribution models. *Eudora* an email client, is distributed in two forms: a free version with embedded advertising and a professional version with enhanced functionality and a fee-based distribution.

Case Study: Search Engines
http://www.computerworld.com.au/cwt.nsf/All/6710746D4ABDD02E4A2569EB007B8B56!OpenDocument&NavArea=Email&SelectedCategoryName=Computerworld+Today

Data mining or trespass?

BY SANDRA VAN DIJK

7 February, 2001 6:00 SYDNEY, AUSTRALIA

[OTHER STORIES ABOUT DEACONS LAWYERS DIGITAL INDUSTRIES GROUP]

Australian companies that use Web crawlers or similar data mining technologies face legal liability under trespass laws following two recent court cases in the US.

Leif Gamertsfelder, a member of Deacons Lawyers Digital Industries Group, said both cases have legal implications for every Australian company using this type of technology as part of its operations.

"If Australian trespass laws are used to protect certain Web sites from spiders, courts will need to ensure the correct balance is struck. If these laws are abused there may be stark implications for the use of technologies that are the very basis of the Internet, such as linking and metasearch engines," he said.

Register.com Inc (a domain registry) and eBay have successfully argued that companies using data mining technologies that extract data from their respective sites breach trespass laws.

Both cases are headed for the US Federal appeals court but if both organisations can defend the injunctions granted by the lower courts Gamertsfelder believes it will have "chilling implications" on the use of spider technologies. He said it is not implausible that an Australian test case could be brought before the courts to determine whether trespass laws could be used to protect valuable information and the information processing resources of Australian companies.

As a result Australian companies that spend significant resources aggregating valuable data may use these laws to protect data from 'free riders'. Gamertsfelder said it is not just the valuable data that sits on company servers that is at stake.

"The use of Web spiders by rival companies can tie up substantial processing resources and if company servers are tied up delivering data to rival companies then legitimate customers miss out; this translates to lost revenue for those who are targeted by Web spiders," he said.

"Australian courts are generally more conservative than their US counterparts and this should ensure that remedies under local trespass laws are only granted in appropriate cases."

Gamertsfelder said an example of an appropriate case could be where a company used data mining technology on a competitor's site: to such an extent that it was clearly a breach of the Web site's terms of use, had an adverse affect on the performance of the competitor's site, and where it can be argued that the competitor implied consent to such activity

The Open Source Definition *(Version 1.7)*
From **http://www.opensource.org/osd.html**

Bruce Perens wrote the first draft of this document as `The Debian Free Software Guidelines', and refined it using the comments of the Debian developers in a month-long e-mail conference in June, 1997. He removed the Debian-specific references from the document to create the `Open Source Definition'.

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution
The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale. (rationale)
\
2. Source Code
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost -- preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed. (rationale)
3. Derived Works
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software. (rationale)
4. Integrity of The Author's Source Code.
The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software. (rationale)
5. No Discrimination Against Persons or Groups.
The license must not discriminate against any person or group of persons. (rationale)
6. No Discrimination Against Fields of Endeavor.
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research. (rationale)
7. Distribution of License.
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties. (rationale)
8. License Must Not Be Specific to a Product.
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution. (rationale)
9. License Must Not Contaminate Other Software.
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software. (rationale)

Patents

Patents cover the protection a specific, implementable, method (for example in software this might include a sorting algorithm, or a caching technique). Software techniques can be patented in some countries. Patents see the publication of the details of an invention for public access in exchange for an exclusive right to determine who and how it can be used for a fixed period of time (10-20 years). Patents only apply to a new technique or invention, and this protection can co-exist with copyright.

Countries that respect International Patents conventions generally manage a Name and Subject index database at the national level. Online international patent databases (fee based searching) provide a means for those seeking patent protection of researching what is current covered by patents – however some databases may only show granted patents. Many use commissioning agencies to perform detailed searches. The process of seeking patent protection can be slow and expensive, and the patent seeker will often find that large Corporates take out tactical and strategic patents. The cost of seeking a patent however is rewarded with particularly strong rights commanding exclusive control on the technique.

Trade-marks

Trade-marks protect your investment in a “brand” or “name” . Many organisations make a large investment in brand or product awareness, key among which investments is ensuring that customers can reliably recognise your product by name, image or icon. Trade-marking is regional rather than international in scope: you must register your trademarks in every country in which you want to achieve protection. Characteristically, this may cost $AUS1000-$10000 for coverage in a single country or $30,000 to $100,000 to gain international coverage in multiple countries.

Trade Secrets

A key disadvantage of patents is their time limitation. For a corporation seeking long term protection of their Intellectual Property, the 10-20 years exclusive control bestowed by a patent may not be satisfactory. Where the invention can be securely protected and not easily replicated (and thus patented) by another, Trade Secrets are another way forward. Perhaps the most popular example of Trade Secrets as an alternative is that of Coca-Cola. The formula for the famous drink is not protected through a patent. Trade secrets can include: formulae, designs, processes, specifications, reports and so on. There are no formal specific protections associated with Trade Secrets. To rely on Trade Secrets is therefore to rely on your capability to protect those trade secrets within your organisation.

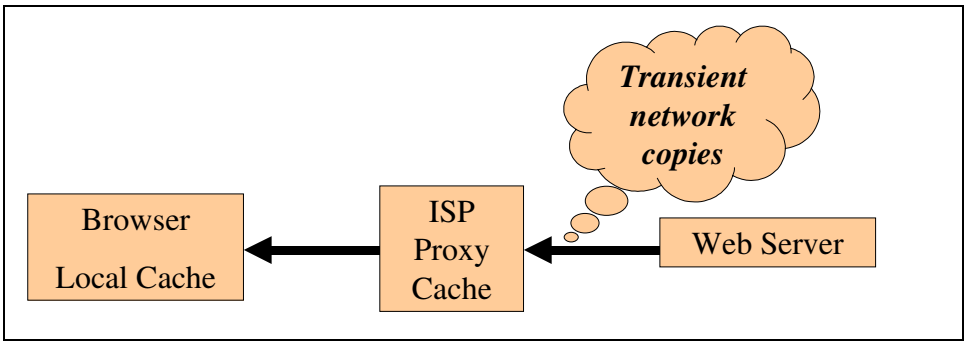
HORSES FOR COURSES

	Used for	Protection	Examples
Copyright	Creative works: software (binary and source *but not usually screens/user interface*), written expression, paintings, works of music, art, drama	Usually 50 years after the death of the author. Longer in some "Mickey Mouse" countries.	A book A program (compiled) Source code for a program A video A music CD
Trade Secret	Internal documents	Your own security systems	Customer lists
Trade Mark	Brand Names	As long as the trademark remains registered	*Eg "Coca-Cola", "Swoosh" symbol of Nike*
Patent	New inventions, software algorithms, business processes	10-20 years depending on the country	A new computer search algorithm

DIGITAL CONTENT ISSUES

Copying of intellectual property resources has never been easier. This presents an immense difficulty for those attempting to enforce their Intellectual Property rights. Copying equipment (digital and photocopy) is ubiquitous, and copying results in identical or near-identical copies often at a much lower cost. Tracking and counting the copies has never been harder in the context of the Internet. In this context, the effort applied to copyright protection must be relative to its long term value. For instance, news broadcasts have relatively high short-term value, and news organisations in the first instance will place considerable resources in asserting their rights at the immediate time of delivery. Some Intellectual Property has long term value to the copyright holder – this may include news items that eventually gain historical database value. The Intellectual Property protection systems should therefore reflect the relative value of the item to be protected. Once again, the efforts to be protected may be guided by financial desire to contain or control copying, or they may derive from a desire to maintain authenticity.

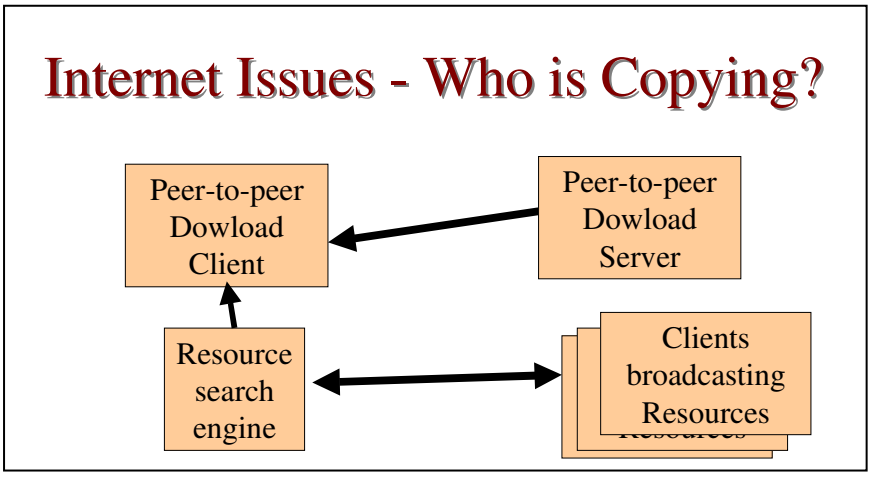
A systemic problem with Internet publication is the use of caching – that is the transient storage of content (web pages, images, etc) on intermediary network devices. Caching can persist for a considerable amount of time and it remains an open question whether caching represents a "copy" for purposes of Copyright law.



Problematic in standard web service delivery is the identification of the copying agents.
This is not to say that Intellectual Property rights cannot be protected ; however, it is still a very expensive exercise to police the protection of sites. Peer-to-Peer models of interaction provide models whereby content (such as MP3 audio cuts or recordings) can easily be shared. The lack of anonymity which prejudices the longevity of the Napster site can quite easily be overcome with slightly different software design.

Traditional Internet servers provide some level of capability to monitor the flow of information.

Internet caching servers complicate this issue, both legally and in terms of tracking – when content is served by a caching server rather than your own server, there are issues at law still as to whether the "cached" copy, or indeed the transient copies at all stages of the network, are copyrightable items.



Peer-to-peer networking tools overlayed on top of the traditional networking infrastructure, however, add an additional complexity in monitoring the copying process.

Whilst the various Napster legal precedents have limited the extent to which Napster can continue to deliver the sort of MP3 audio sharing with which it made its name, alternative approaches to this technology are even now being deployed.

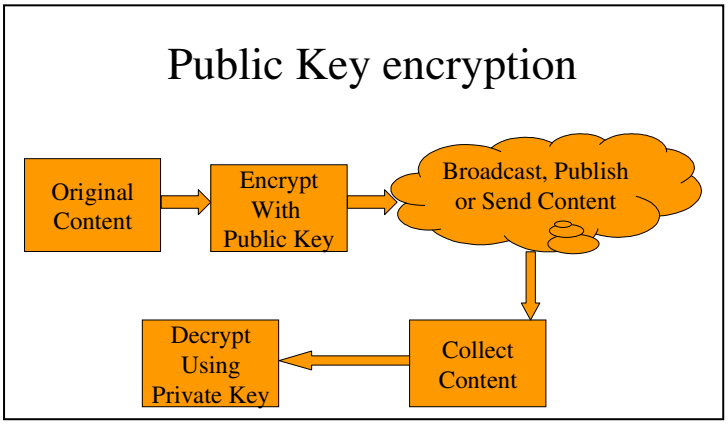
The Sydney Olympics were, therefore, an interesting first effort to monitor and enforce copyright and licensing restrictions in web use on a global basis. NBC, having licensed the TV rights in the US launched, and by-and-large succeed in severely restricting the use of digital video streaming of events by other broadcasters (in order to protect the currency of their TV broadcasts). The success in suppressing non-licensed digital video streaming demonstrated that existing legal protections can cover Internet activities. The prohibitive cost of undertaking such monitoring (teams of Internet researchers constantly monitoring web site activity) will probably itself be aided by technology.

Digital Watermarking and Public Key Encryption

Efforts to build protection for publicly accessible resources have seen two innovations in content delivery, sometimes integrated for better protection:

- Digital Watermarks
- Public Key encryption

Public Key Encryption may be used to both protect information as it transits a network and to provide identification of the sender and the receiver. An E-commerce site might deploy such methods to ensure that a user sending orders does not reveal important financial information. A subscription-only site might use such methods to authenticate access to the site.



Public Key encryption uses a publicly available key to encrypt the source contents to be sent. This transient information can only be decrypted by those holding the corresponding *Secret Key.* Its advantage over simply deploying a secret key at both ends is the ability to disseminate widely the *Public key.* It is a useful approach for centrally based hosts who need to receive information. *Private key encryption* approaches are effective for security but face the difficulty of widely sharing and updating the key needed at both ends.

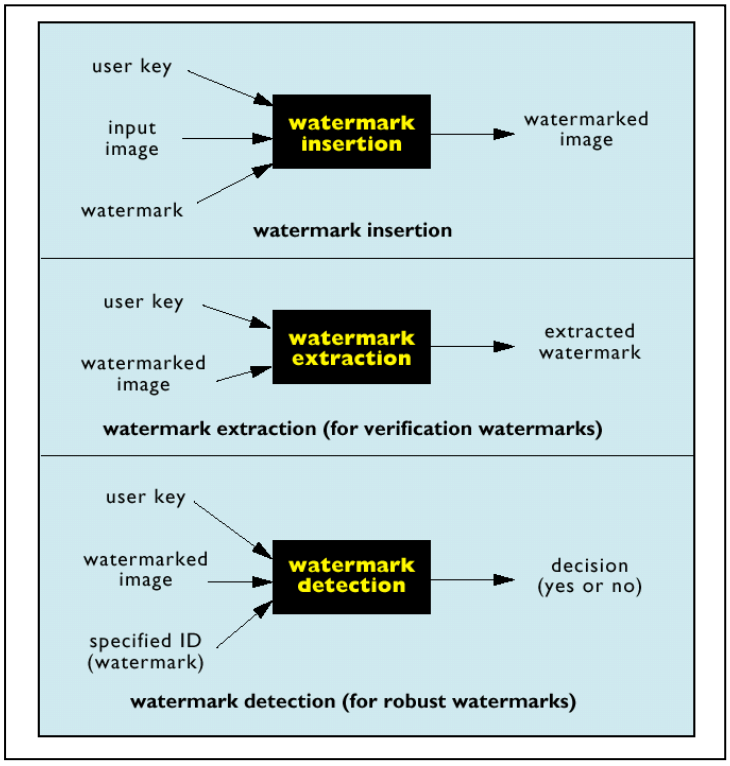
The benefits of proper privacy and authentication are significant as information resources are shared over common network architectures such as the Internet.

Authentication requires that the user community has ready access to the registration authority. Of course there is a trade-off: the easier the registration the weaker the strength of authentication.

A second technique to managing digital Intellectual Property rights is the use of 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.

Digital Watermark advantages can be used for fingerprinting – that is identification of the usage of a document or content item by uniquely identifying each manifestation (for example: who leaked this document?). It is also effective in asserting authenticity – the absence of a validated watermark may be taken as an indication of corrupted or suspicious content. This is strengthened with used in conjunction with public key encryption methods: making it harder to falsely insert a watermark. Finally, Digital Watermarking can have the effect of making copying less valuable if such copying must persist showing the originators meta-identification information.



Cable TV uses a combination of public key encryption and watermarking for content delivery. DVD uses public key encryption and illustrates the danger of loose control of private keys – DVD copying techniques abound. In digital broadcasting the issue of Intellectual Property on the image being broadcast is a fiercely contested issue.

The ability of broadcasters to use Watermarking techniques to overlay advertising on sport broadcasts has seen test cases on the ownership of the broadcast image.

OTHER APPRAOCHES

Login authentication without encryption is a popular means of limiting copying or access right to digital resources. In a large population, password sharing and control is a problem, but cost of implementation is considerably lower than the use of *Public Key Encryption.* This approach is very amenable to Cracking and dictionary-based attacks - where a standard dictionary is used repeatedly against the login form to attempt to guess an access password.

Trickle Feed is another approach. Rather than limit access, information is provided only a little information at a time. A fine granularity of information delivery frustrates large-scale copying, or makes it un-economical.

Finally, the intellectual copyright holder may prefer to act retrospectively: to detect copyright breaches rather than prevent them. This can be facilitated through agencies that specialise in such approaches, or through development of network scanning techniques.

SYSTEMS THINKING – IS IT FAIR USE?

There are some simple checks that are warranted to determine fair use of resources created by others. Copyright and other IP legislation provides for individual penalties which are quite significant[Koch, 1996 #6]. The following simple questions are warranted whenever you use resources obtained elsewhere:

(1) Is the work IP copyright protected? In particular, copyright protection does not extend to:

- Works that lack originality
- Logical, comprehensive compilations (like the phone book) – if you are quoting from them
- Unoriginal reprints of public domain works
- Works in the public domain
- Freeware (not shareware, but really, expressly, available free of restrictions-ware -- this may be protected by law, but the author has chosen to make it available without any restrictions)
- Facts
- Ideas, processes, methods, and systems described in copyrighted works

(2) If the work is copyright protected, can you fairly use a portion of the material?

- Copying a single chapter or a single journal article for research purposes (however, repeated use of such materials may still require permission of the author)

- Performances for purposes of teaching.

SUMMARY

Commonly content authors now work in teams, and the creative process for many products more closely resembles a movie studio than the artists loft. Intellectual Property issues are too often an afterthought for authors in the Arts and Humanities. Even where the goal of the author is to achieve free distribution of their creation considerations of Intellectual Property protection can still apply. Systems Thinking should incline the author to considerations of the methods, consequences and rights surrounding the creative process.

QUESTIONS

1. Discuss the differences between Open Source Licensing and traditional software Licensing in the protection of Intellectual Property Rights.

2. Describe in detail two methods of digital copyright protection.

3. What is meant by an Open Source License?

4. What is the purpose of a Digital Watermark?

FURTHER READING

AUGOT, D., Boucqueau, Jean-Marc, Delaigle, C. F. and Goray, E (1999). Secure Delivery of Images over Open Networks. In Proceedings of the IEEE, Vol 87, No 7, July 1999, pp. 1251-1265.

BALNAVES, E. "Systematic Approaches to Long Term Digital Collection Management ". Literary and Linguistic Computing. Sept 16, 2005

DIJK , Sandra Van (2001). "Data mining or trespass?" Computerworld, 7 February, 2001 6:00 SYDNEY, AUSTRALIA

ESPN. "Sites attempt to show Games online." http://sydney2000.espn.go.com/oly/summer00/news/2000/0921/765070.html
"The Open Source Definition, v.1.7". http://www.opensource.org/osd.html

HAYES,D (2000) "Internet Copyright: advanced copyright issues on the Internet." Computer Law & Security Report, Vol 16, No 6, 2000, pp.363-377

HARTUNG, F and Kutter, M. (1999) Multimedia watermarking Techniques. In Proceedings of the IEEE, Vol 87, No 7, July 1999, pp. 1079-1107.

"MCB". "The GIF Controversy: A Software Developer's Perspective" http://cloanto.com/users/mcb/19950127giflzw.html

FURTHER READING - WEBSITES

KENNEDY, Dennis. A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture. http://www.denniskennedy.com/opensourcedmk.pdf

IP AUSTRALIA. http://www.ipaustralia.gov.au. - the Australian copyright/trademark website, with excellent online resources on Intellectual Property

SURFIP. http://www.surfip.gov.sg - the Singapore IP portal – another excellent Intellectual Property resource gateway.

COPYRIGHT COUNCIL. http://www.copyright.org.au/ - the Australian Copyright council website.

http://www.lib.utsystem.edu/copyright/ - has an excellent tutorial covering many aspects of copyright, *Services*, Volume 23, Issue 4, Winter 1999, Pages 459-467

INDEX

Archaeology, 19, 162, 168, 169
ASP, 51, 56-79, 80, 82, 83, 84, 86, 88, 95, 99, 191
Cascading Style Sheets, 34, 46, 52, 54, 55, 120, 191
Cataloguing, 108, 121
Classification, 44, 108, 121, 122
CMS (Content Management Systems), 77, 104, 105, 106, 108, 110, 116, 169, 191
Computational Linguistics, 164
Content Delivery
  HTTP, 55, 61, 79, 80, 96, 97, 120, 143, 192, 195
  Knowledge Management, 193
  SOAP, 97, 103, 195
  Web Services, 102, 103, 117, 195
Copyright, 37, 40, 134, 175, 176, 182, 186, 188
Data Mining, 143, 165
Delphi Method, 138, 140, 161
Digital Convergence, 27, 28
Digital Libraries, 136
Document standards, 30-35, 36, 46, 48, 49, 52, 53, 54, 55, 59, 60, 63, 71, 74, 77, 78, 88, 89, 91, 93, 97, 98, 100, 109, 118, 120, 121, 125, 144, 162, 191, 192, 193
Federated Searching, 135
HTML, 30-36, 46, 48, 49, 52, 53, 54, 55, 59, 60, 63, 71, 74, 77, 78, 88, 89, 91, 93, 97, 98, 100, 109, 118, 120, 121, 125, 144, 162, 191, 192, 193
Hypertext, 28, 55, 116, 120, 192, 194
Hypertext Transfer Protocol, 55, 61, 79, 80, 96, 97, 120, 143, 192, 195
Intellectual Property, 2, 5, 174, 175, 176, 181, 182, 183, 185, 187, 188, 193
  Copyright, 37, 40, 134, 175, 176, 182, 186, 188
  Open source, 135, 178, 180, 187, 188, 194
  Patents, 181
Intelligent Agents, 131
IT Governance, 22, 23, 24
ITIL, 22, 23, 24
Markup, 4, 31, 34, 44, 45, 47, 54, 55, 105, 117, 120, 170, 172, 195
  MARC, 55, 120, 135, 193
  Ontology, 102, 194
  RDF, 55, 101, 102, 103, 118, 120, 194
  RSS, 3, 101, 116, 118
  SCORM, 102, 194
Multimedia, 3, 27, 28, 29, 30, 33, 35, 37, 38, 39, 40, 41, 42, 43, 55, 100, 107, 113, 116, 117, 118, 119, 120, 149, 152, 172, 175, 177, 188, 193, 195
Music, 162, 172, 175
Obsolescence, 17, 19, 41, 105, 172, 194
Ontology, 102, 194
Open Source, 135, 178, 180, 187, 188, 194
Patents, 181
Philosophy, 21, 162, 171
PHP, 56- 79, 80, 82, 83, 86, 90, 91, 92, 93, 94, 95, 99, 194
Python, 4, 50, 51, 56-79, 80, 81, 83, 84, 87, 88, 89, 94, 95, 99
Qualitative Research Methods, 138, 161
  Action Research, 10, 138, 139, 141
  Data Mining, 143, 165
  Delphi Method, 138, 140, 161
  Focus Group, 138, 139, 160
  Grounded Theory, 140, 141, 142
RSS – Rich Site Summary, 3, 101, 116, 118
SCORM (Sharable Content Object Reference Model), 102, 194
Scripting, 2, 4, 54, 56, 57, 65, 67, 70, 74, 80, 81, 82, 84, 85, 87, 94, 95, 96
Scripting Languages
  ASP, 51, 56- 79, 80, 82, 83, 84, 86, 88, 95, 99, 191
  Javascript, 35, 56, 72, 193
  PHP, 56-79, 80, 82, 83, 86, 90, 91, 92, 93, 94, 95, 99, 194
  Python, 4, 50, 51, 56-79, 80, 81, 83, 84, 87, 88, 89, 94, 95, 99
Search Engines, 130, 137, 178
  Federated Searching, 135
Service Level Agreements, 23
SGML, 21, 31, 33, 34, 42, 45, 46, 47, 48, 49, 53, 101, 111, 112, 192, 195, 196
  HTML, 30-36, 46-55, 59, 60, 63, 71, 74, 77, 78, 88, 89, 91, 93, 97, 98, 100, 109, 118, 120, 121, 125, 144, 162, 191, 192, 193
  XHTML, 192
  XML, 30-64, 47, 48, 49, 50, 51, 52, 53, 54, 55, 78, 97, 101, 102, 103, 105, 110, 111, 112, 120, 162, 163, 170, 172, 174, 192, 193, 194, 195, 196
SOAP (Simple Object Access Protocol), 97, 103, 195
System Thinking
  Obsolescence, 17, 19, 41, 105, 172, 194
Systems Theory, 6, 7, 12, 26
  Archetypes, 10, 11
  Systems Thinking, 2, 4, 7, 8, 24, 143, 173, 187
UniCode, 33, 46
Web Services, 102, 103, 117, 195
XML, 30, 31, 33, 34, 47-55, 78, 97, 101, 102, 103, 105, 110-112, 120, 162, 163, 170, 172, 174, 192, 193, 194, 195, 196

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 rate at which information can be passed between computers. A wider bandwidth means more content can traverse the network in a shorter amount of time.
Browser: software for navigating the Web, retrieving documents and other files, commonly in HTML mark-up format.
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.
Book: An analog device for random access to printed multimedia content. An information storage device which is portable, requires no power supply, and has minimal issues of obsolescence.
BBS: Bulletin Board System. BBS, chat and other personal information exchange systems remain a highly popular form of Internet communication. The BBS provides a topic-based environment for a community of users to exchange text-based communication.
Codec: The compression and decompression algorithm for audio and video content.
CMR: Content Model for Reuse. The model proposed in this thesis for systematic reuse of multimedia content.
CMS: Content Management System.
CSS: Cascading Style Sheet. The CSS defines rule-based presentational instructions for HTML content mark-up. The Style Sheet has the merit of gaining a greater freedom from the specific encoding of procedural mark-up within the text itself (with the <font> tag and others).

- D -

Digital (content): Information that is encoded in binary (discontinuous) form particularly and mediated by computers.
Directed Acyclic Graph: A directed graph where no path starts and ends at the same vertex.

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.

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.
DCMI: *Dublin Core Metadata Initiative* **-** a standard for consistent meta-identification of website publications

---

- 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 (eg a Web server). In the Internet a Client Web Browser connects to a Host Web Server to exchange HTML and other information.
HTML: HyperText Mark-up Language. A set of mark-up 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 IEEE Learning Object Model - 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.
ISO 639-2 - An International standard for country codes - used by XML and HTML for country definition.

- 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.
KWIC: *Key Word In Context*. A search result display which shows the keyword searched in the sentence context in which it occurs.

- L -

Local Area Network (LAN): a group of computer connected together for high-bandwidth file and application sharing.

- M -

MARC - Machine Readable Cataloging. A metadata ontology for exchange of bibliographic information.
Mark-up: 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 Mark-up*
NewsML: *News Mark-up Language* - a content exchange framework specifically designed for XML interchange and syndication of news items.
Multimedia: Any combination of text, audio animation and video content in a digital form.

- 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: as 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.
PHP: PHP Hypertext Processor (yes, the definition is self-referential, or recursive).
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.
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 heterogonous 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. 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.
SGF: *Structured Graph Format* - Defines an XML metadata format for exploration of overlapping hierarchies of content - especially websites.
SGML: Standard Generalised Mark-up Language. A universal syntax for defining mark-up language. A "meta-language".

SMIL: *Synchronized Multimedia Integration Language* - a multimedia document ontology for presentation of heterogenous 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 mark-up standard for SGML mark-up of texts in the Humanities.

- U -

URI*: Uniform Resource Identifier*. A generalised format for resource identification. A URL is a specific implementation of a URI.
Unicode: An international standard for binary character set encoding of text in different languages.
Unix: an operating system developed in the1960's and a popular platform for Internet applications. Linux is closely modelled on Unix.
URL: *Uniform Resource Locator*, the address of a document or other Internet resource - a particular instance of a URN 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: The World Wide Web consortium. Responsible for publishing the WWW standards.
Web Services: That 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 mark-up hidden and presented as it would be finally published.

- X -

XML – *Extendsible Mark-up 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 mark-up 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