

2001

Multivalent Metadata : Exploiting the Layers of Meaning in Digital Resources

Alison M. Anderson
Edith Cowan University

Follow this and additional works at: https://ro.ecu.edu.au/theses_hons



Part of the [Cataloging and Metadata Commons](#)

Recommended Citation

Anderson, A. M. (2001). *Multivalent Metadata : Exploiting the Layers of Meaning in Digital Resources*.
https://ro.ecu.edu.au/theses_hons/881

This Thesis is posted at Research Online.
https://ro.ecu.edu.au/theses_hons/881

Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author's moral rights contained in Part IX of the Copyright Act 1968 (Cth).
- Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

**MULTIVALENT METADATA: EXPLOITING THE LAYERS OF
MEANING IN DIGITAL RESOURCES**

By Alison M. Anderson

This Thesis is submitted in part fulfillment of the requirements for the award of

Bsc (Interactive Multimedia) Honours.

Department of Communication Health & Science

Edith Cowan University.

Supervisor : Dr Arshad Omari

Date of Submission March, 2001

ABSTRACT

The rapid growth of the World Wide Web was due in part to the simplicity of the Hypertext Markup Language (HTML). It is anticipated that the next generation of web technology, coined the *Semantic Web*, by Tim Berners-Lee (1989, p. 1), will be driven by the Extensible Markup Language (XML). The XML suite of technologies provides a framework for the application of metadata, and hence semantic information, to web resources. Advantages of a semantic web include improved sharing and reuse of resources, enhanced search mechanisms and knowledge management.

The knowledge or meaning contained in digital information may vary according to the perspective of the viewer and can be seen therefore as multivalent in nature. Semantic information that is highly relevant to one user may be of no interest to another.

The aim of this project was to demonstrate the layers of meaning inherent in a data sample and how they could be encapsulated in metadata then accessed and manipulated using current technologies, thus leveraging the knowledge contained.

Analysis of the data sample, a typical component of an online training product, determined meaningful ways in which the knowledge contained could be reused and adapted. From this analysis a set of test criteria was generated. Metadata was then created for the sample data and the tests implemented using a range of XML technologies.

To prevent ambiguity and facilitate information interchange across heterogeneous sources metadata should adhere to relevant schemas, taxonomies and vocabularies. To this end standards relevant to the data sample were researched and adopted. Facilitating the sharing and reuse of information also requires a consensual understanding of the concepts and relationships involved. Ontology is a field of Artificial Intelligence research that specialises in the representation of the concepts and relationships within a domain of knowledge. The final test criteria explored how relationships defined in an ontology could be encoded using XML technologies and used by an inference engine to enhance search options specific to the data sample.

A range of XML technologies including, the Resource Description Framework (RDF) and the Extensible Style sheet Language for Transformations (XSLT) was used to apply and manipulate the meaning contained in the data sample.

I certify that this thesis does not, to the best of my knowledge and belief:

- i. incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education;
- ii. contain any material previously published or written by another person except where due reference is made in the text; or
- iii. contain any defamatory material.



Alison M. Anderson

ACKNOWLEDGMENTS

I would like to thank Dr Arshad Omari in his capacity as Supervisor for his support, direction, patience and making sense of wild ideas.

I would also like to thank Simon Payne for loaning me his XML books and Melanie Burke for her comments on holistic assessment.

Lastly I would like to thank my husband and sons for being self-sufficient and doing their own cooking and laundry while I was preoccupied with this project.

TABLE OF CONTENTS

Chapter 1.....	1
<i>Introduction.....</i>	<i>1</i>
Background.....	1
Significance.....	3
Project scope.....	6
<i>Definition of Terms.....</i>	<i>8</i>
Chapter 2.....	11
<i>Literature Review.....</i>	<i>11</i>
Introduction.....	11
Knowledge management.....	11
Metadata.....	15
Ontologies.....	18
XML Technologies.....	23
Summary.....	33
Chapter 3.....	34
<i>Methodology.....</i>	<i>34</i>
Introduction.....	34
XML Notepad.....	34
Internet Explorer 5.5.....	35
XT.....	36
XP, SAX and DOM.....	37
RDF, SiRPAC and SiRLI,.....	38
Summary.....	39
Chapter 4.....	40
<i>Implementation Stage 1.....</i>	<i>40</i>
Introduction.....	40
Data sample.....	40
Metadata guidelines.....	43
Test Criteria.....	45
<i>Implementation Stage 2.....</i>	<i>46</i>
Standards and schemas.....	48
Conceptual information.....	58
Summary.....	66
Chapter 5.....	68
<i>Tests and results.....</i>	<i>68</i>
Introduction.....	68
Test 1.....	68
Test 2.....	72
Test 3.....	75
Summary.....	77
Chapter 6.....	78

<i>Conclusion</i>	78
References	80
Appendix 1	86
<i>Data Sample</i>	86
Appendix 2	92
<i>Metadata at the highest level of granularity</i>	92
Appendix 3	94
<i>XML Schema - Scenario.xsd</i>	94
Appendix 4	96
<i>XML version of the data sample – team.xml</i>	96
Appendix 5	100
<i>RDF Scheme</i>	100
Appendix 6	102
<i>Java Program (using SAX)</i>	102
Appendix 7	105
<i>DOM Example</i>	105
Appendix 8	109
<i>XSLT style sheet</i>	109

TABLE OF FIGURES

Figure 1. RDF node and arc graph.....	30
Figure 2: XML Notepad.....	35
Figure 3: Processing XML.....	37
Figure 4: RDF model.	38
Figure 5: Using multiple XML schemas in a single instance document.....	57
Figure 6: Simple Logic-based Interpreter (SiRLI).....	59
Figure 7: Learning resources in an RDF metadata repository.	66
Figure 8: Output from test 1.....	71
Figure 9: Output form Test 2.	73

CHAPTER 1

Introduction

Background

Recent developments in information technology have resulted in an unprecedented growth in the amount of digital information available to us. Accordingly we require systems and mechanisms to manage both the copious amounts of information and its expansion. An even greater challenge is to derive benefit from the information by converting the vast repositories of raw data into meaningful knowledge, a process aptly coined by Stabb, Erdmann, Maedche, & Decker (2000, p. 1) as *knowledge provisioning*.

Davenport & Prusak (cited in Baxter & Chua, 1999, p. 4) define knowledge as “high value information ready to apply to decisions and actions”. This value is added by applying context and relationships to information components, thus providing semantics or meaning to the raw data. Metadata can be used to apply this semantic information to digital information. The semantic web envisaged by Berners-Lee (1989, p. 1), uses the XML family of technologies to store and manipulate metadata.

XML is a subset of the Standard Generalised Markup Language (SGML). SGML was designed as a universal format for the interchange of data. It is a comprehensive and stable technology but was too complex for the requirements of web publishers. The World Wide Web Consortium (W3C) created XML using only

the SGML features that were required for the web environment (Ladd & O'Donnell, 2000, p. 48). Despite its name XML is not a markup language in itself. It is standard for creating markup languages and is therefore referred to as a meta-language. Markup languages that conform to the standard are referred to as XML applications. The XML standards overcome several of the problems inherent in the Hypertext Markup Language (HTML) currently used by web publishers.

HTML was also derived from SGML. Its success is largely due to its simplicity and the fact that it was compatible with almost every computer system. The main criticisms of HTML are its lack of syntactical rigor and the fact that it is not extensible. Of particular interest to this project is how the removal of these limitations would impact on the application and manipulation of metadata.

HTML provides a standard set of tags that inform a browser how to display information. They provide no indication of what the information is about. This does not present a problem to humans who can infer the meaning as they view the text on screen. The computer however, has no information to work with and thus the opportunity to computationally manipulate the information is lost. Any semantic information about the data such as keywords or source details must be stored within the HTML `<META>` tag in the head of a document. The browser does not display the information in this tag but it can be accessed by software applications such as search engines. In contrast the XML standard allows users to create their own sets of tags that describe what the information is about rather than how it should be displayed. XML separates presentation from content. This framework provides interesting opportunities for the storage and manipulation of semantic information which in turn will drive the knowledge provisioning described by Stabb et al.

This project implements several XML applications, including the Resource Description Framework (RDF) and the Extensible Style sheet Language for Transformations (XSLT) to demonstrate their capabilities as a contribution to knowledge provisioning through the manipulation of metadata. This process would evaluate how these technologies and semantic information could be used in regards to the particular situation the data sample represented and identify the specific benefits that can be derived for the associated users.

Significance

The growth of electronic commerce and global information exchange drives demand for tools that can enable the flow of data between diverse systems. In the foreward to Goldfarb and Prescod's XML Handbook , Paoli (2000, p. 1iii) shares his vision that data should "easily travel from server to server, from server to client, and from application to application, fostering universal communication with anyone, anywhere." XML goes a long way in making this a reality. It provides us with the technical framework to smooth the flow of data between a host of platforms and systems.

If we consider this technical framework as the physical level, then in order to utilise it effectively we also require universal communications on a logical level. In other words there is also a need for a common language and understanding of knowledge components, if not at a universal level certainly at a domain level. This can be observed in e-commerce and business-to-business (B2B) transactions. A common business language allows information flow between all contributors in a business domain. The internal transactions of an organisation are extended to include outside goods and service providers such as manufacturers and transporters.

Thus the supply chain is transformed into an integrated value system (Yang & Papazoglou, 2000, p. 39).

Advances in technology are also impacting on the traditional approach to education and training and once again there is a demand for common communication systems. At a global level Beard (2000, p. 2) ambitiously talks about the development of courses based on “shared resources created by several instructors at several universities located around the world”. Schatz (2000, p. 2) suggests that training should no longer be seen as linear processions with a beginning, middle and end, but as clusters of independent, stand alone bits of knowledge that can be adapted and reused. Goldberg Tschritzis (1999, p. 93) recommends that universities need to find ways of creating financial rewards through reusing the content they generate, and selling it to other educators on a global scale. On a much smaller level Intelligent Education Systems (IES) use computer agents to exchange information between learners and a learning product (Chen & Mizoguchi, 1999, p. 1). These processes all rely on integrated systems with a common language for communication.

Training and educational material also need to be constantly updated to reflect changes in the workplace, learner preferences and demands, and evolving technologies. Developing and updating online materials is a costly exercise for trainers and educators. Using existing knowledge to update or create new material is therefore a desirable process. Some customisable course components are already available on the Web such as WebCT's *e-packs* which offer items such as lecture notes, glossaries and video clips (WebCT, 2000). In his article *Publisher Beginnings* Goldberg suggests that educators use this kind of content as a starting point and then customise it to their specific style and needs. WebCT is however a proprietary

software system and as such can be problematic and inflexible for users to customise. Since XML is a globally accepted standard resources encoded in XML languages would overcome this problem. Also XML technologies can allow knowledge bits to be identified at much lower levels of granularity thus increasing flexibility and extensibility, allowing as Schatz suggests individual ‘bits’ of knowledge to be combined and utilised by the end user at their discretion.

This concept is similar to the object oriented approach widely used in software engineering. Objects are ‘bits’ of software code that model real world entities and are defined by their *methods* and *variables*. Objects can be reused with various instances of their methods and variables. To continue the analogy training knowledge objects could be defined by methods such as the instructional method or programming language used and variables such as the expected learning outcomes or creators. The ability to locate, reuse and adapt such knowledge objects would enable the leveraging of training knowledge and demonstrates the concept of knowledge provisioning.

This process is dependent on the application of metadata to identify the object components and their specific attributes (methods and variables). There are three key issues that need to be acknowledged.

1. While XML allows users to define their own tags and thus their own metadata, adherence to standards and schemas will enhance the interchange, sharing and reuse of knowledge objects.
2. The meaning inherent in raw information can be multivalent in nature and therefore when using metadata to describe knowledge objects a wide range of user viewpoints should be considered.

3. Machine understandable metadata has powerful implications for search mechanisms and intelligent software applications. While all XML applications are machine-readable only RDF is machine understandable.

Each of these issues was evaluated in the context of the data sample. For the purpose of this project the data sample is considered as representing a knowledge object pertinent to the field of training and education, that contains attributes that can be defined by metadata. The benefits to be gained will be viewed in respect of the situation represented by the data sample. However it should be noted that the overall implications are also applicable to other domains.

Project scope

This focus of this project centres on the knowledge contained in a single component of an online training package. By restricting the project to a small sample a more thorough evaluation could be done. Although multimedia is an important aspect of training and education, this area was considered too comprehensive to be included in this particular project thus no media items are contained in the data sample. It is also acknowledged that data relevant to training exists in a wide range of formats and data repositories including digital libraries, online courses and various web-based resources. The data sample represents but one aspect of this divergent industry.

Issues relating to copyright in regards to the sharing and reuse of resources and individual resource components were not considered relevant to the objective of this project and are therefore not covered in any way.

No attempt was made to validate the effectiveness or deficiencies of any of the software applications used. The focus was instead to demonstrate the benefits of harnessing and manipulating various levels of semantic information.

Definition of Terms

Agents	Agent technology is diverse, however for the purpose of this project an Agent is assumed to be an autonomous piece of software that is capable of <i>spontaneous execution</i> , and <i>initiative</i> that works collaboratively with users and other Agents to achieve an objective.
Attribute	An attribute encapsulates data about an entity, for example <i>Surname</i> is an attribute of <i>Person</i> .
B2B	Business to Business refers to data exchange between businesses operating in a common industry domain.
DC	The Dublin Core (DC) is a standard that provides a core set of semantics for web resources. Details can be found at http://purl.org/DC/about/index.htm .
Element	For the most part the term <i>element</i> is used in this project in reference to a metadata identifier. However it is also used in relation to Training Package learning outcomes such as <i>element of competency</i> .
Frame Logic	A formalism for representing/modelling fundamental concepts. It is based on object oriented design principles.
IES	Intelligent Educational Systems, also referred to as Intelligent Tutoring Systems. IES software use 'intelligence' by altering a systems response to a learner as they work. The software usually includes Agent technology and makes decisions based on assessment of the learner's interaction and progress and pedagogical knowledge about how to teach.
LOM	IEEE Learning Object Metadata Standard that specifies the syntax and semantics of learning object metadata, used to describe a learning object.
Metadata	Data about data necessary for the identification, use and management of resources.

Method	A section of code that when executed performs a specific function.
Multivalent	The opposite of bivalence or two –values (True or False, 1 or 0). Multivalence means three or more options.
Object Oriented	Design and development based on <i>objects</i> . An object is an entity that models some real life thing or problem. Objects with similar properties and behaviours are grouped as a <i>class</i> .
Ontology	A formal representation of a body of knowledge based on concepts and relationships (as opposed to a vocabulary or taxonomy that defines only the terms used).
Parser	A parser reads the succession of characters contained in an XML/RDF file and converts them into a logical structure that can be used by other software applications.
Scheme/ Schema	The term <i>scheme</i> (classification) is used by the LOM specification to describe their standards. The XML community use the term <i>schema</i> to describe data models.
Semantics	Meaning or connotations in words.
Syntax	Formal rules that govern how instructions are written in computer software.
User Agent	User agent programs request information from web servers. Some such as a browsers will display the information for end users others such as search engine spiders collect and index web pages for databases and do not interact directly with the end user.
Variable	A reference name or identifier used to store values in a computer program.

W3C

The World Wide Web Consortium, develops interoperable technologies (specifications, guidelines, software,) for the Web through forums, and work groups with collective input from a range of industry and commerce sectors world wide.

CHAPTER 2

Literature Review

Introduction

This project was informed by literature from a range of disciplines but focussed specifically on four main areas: knowledge management, metadata, ontologies and XML technology. Defining the meaning contained in the data sample required research into the wider field of information science and an understanding of knowledge as a concept. The role of metadata in digital resources varies considerably from its origins in traditional paper based resources, thus research in this area focussed on case studies demonstrating its use in the digital environment and monitoring relevant schema standards that are currently evolving. Principles expressed in the field of ontology were analysed for their relevance to the modeling and harvesting of semantic information at both a logical (vocabulary and relationships) and physical (technical) level. Finally a review of the wide range of emerging XML technologies was necessary to determine those suitable for implementation of the project. The following Literature Review covers each of these areas in more detail.

Knowledge management

Stoll (cited in Suzuki, 1998, p. 40) reminds us that isolated facts do not make an education and that meaning does not come from data alone, it requires human input to create context, interrelationships, and experience. This required interaction

between humans and information becomes formidable as the growth of information outstrips our capacity to harness it. Diffusion and absorption of new information is increasingly challenging. One study found that “two years after wide publication, fewer than 50% of general practitioners knew that laser surgery could save the sight of some of their diabetic patients”(Detmer & Shortliffe, 1997, p101). Researchers trying to maintain scientific models find their task increasingly demanding. Altman et al, point out that the growth in biological data has been nearly exponential in the last fifteen years and consequently “even the most experienced biologists struggle when they try to integrate their own observations with the published literature to build coherent models” (1999, p. 68). Educational establishments find themselves pressured into change as the development and transfer of knowledge is redefined by advancements in technology, (Tsichritzis, 1999, p93). Similar challenges are being experienced in many fields, (Baxter & Chua, 1999; Boehme, 1999; Detmer & Shortliffe, 1997; Wise, 1999; Kushmerick, 1999 & O’Leary, 1998). This situation has generated the need for efficient methodologies to enable human interaction with information and meaningful ways in which to work with the resulting knowledge. Thus ‘knowledge management’ has become an increasingly important aspect of current research.

Baxter and Chua (1999, p. 4) correctly ascertain that there is no clear and certain definition of knowledge. Despite this fact, there is an abundance of research work on topics such as knowledge engineering, knowledge management and knowledge mining. Also terms such as “knowledge intensive”, “knowledge-based” and “knowledge representation” have become common in the business world as organisations seek to gain a competitive edge through maximisation of knowledge resources. There is agreement amongst authors from various fields that the addition

of semantic information in the form of context and relationships adds meaning and value to raw data (Baxter & Chua, 1999; Colombo et al, 1999, Berners-Lee, 1998). Tim Berners-Lee coined the phrase *Semantic Web* with his vision of a World Wide Web marked up with machine-readable and in some instances machine understandable semantic metadata. Semantic information may hold the potential to turn the vast repositories of digital information into banks of knowledge, however Berners-Lee himself warns that “The concept of machine-understandable documents does not imply some magical artificial intelligence which allows machines to comprehend human mumblings”.

The Distributed Systems Technology Centre (DSTC, 2000) is a joint venture supported by the Australian Government's Cooperative Research Centres. Their *Information ecology* research project (Ward, 2000) uses a rainforest analogy to express the growing complexity between human and computer information exchange.

“Our investigations assume that information environments of the future will be complex and dynamic systems both by their inherent social nature and due to the influence of new computing paradigms. We argue that increasing complexity in the relationships between people and information invites us to consider the metaphor of organisms in a diverse natural environment such as a rainforest.”

They adopt the term *context* to express the dynamic interlocking relationships in this information system and are investigating context-sensitive models of business activity, computer mediated communication, and inference techniques that can be applied to social information.

While context and semantics can clearly assist the exchange and reuse of knowledge, creating flexible resilient and adaptive communication structures is still thwart with difficulties. Lopez, Gomez-Perez & Sierra, (1999, p. 37) outline some of

these as “heterogeneity of representation formalisms, languages, and tools; lexical and semantic problems; assumptions implicit in each system; and commonsense-knowledge losses”.

Solutions to these problems can be sought from a range of research domains. Yang (cited in Pedersen, 1999, p. 30) outlines the importance of exploiting the synergy between information-retrieval techniques and Artificial Intelligence (AI) research. This research includes intelligent agents, machine learning and natural-language processing. However, the success of retrieval techniques will depend heavily on the representation of the knowledge by metadata systems. Bylander & Chandrasekaran as quoted in Guarino (1997, p1) state that “Representing knowledge for the purpose of solving some problem is strongly affected by the nature of the problem and the inference strategy to be applied to the problem.” This is further complicated by the fact that the meaning drawn from information will differ according to the use of the information by various individuals and organisations. In other words the nature of the problem is often variable or transient.

As mentioned in Chapter 1, multimedia was considered too comprehensive to cover in this particular project. However it should be noted that dealing with media components provides a further challenge in that resources include composite and time-based media. Resources of music, film and video, theatre and performance art, broadcasting arts and dance have their special needs and requirements. Boehme (1999, p. 180) describes performing arts resources as “(a) being made out of different types of data, (b) containing differing complexities of data and (c) possessing different relationships”. Also most research in multimedia is focused on the management of media items themselves rather than the management of processes or associated knowledge (Gecsei, 1997; Miller, Tsatalos & Williams, 1997 and

Colombo, Del Bimbo & Pala, 1999). It should be noted that knowledge management systems that cater for various media must also be flexible enough to cater for the constant updates and developments in media formats.

Metadata

Metadata is essentially data about data. The most obvious advantage of applying metadata to digital resources is improved search and retrieval mechanisms, however as Gilliland Swetland (2000, p. 3) explains it also facilitates:

- a range of access points to cater for different user needs
- the identification of relationships between resources
- certification and authenticity of the resource content
- technical information about the structure of the resource
- the identification of persons involved in the creation and making of digital resources for intellectual property reasons.

One of the first metadata systems for digital media was the Meta Content Framework (MCF) introduced by Apple Computer Inc in September 1996. With support from Netscape MCF was extended to XML and was used by working groups at the World Wide Web Consortium (W3C) in the development of new frameworks for manipulating digital resources including RDF (Boye, 1998).

One of the major problems associated with metadata is the lack of consistency since the author of each resource can decide on their own format and naming conventions. Standards are being developed that offer a common description for semantic information at both a physical and logical level. One of the better known of these is the Dublin Core Metadata Initiative (DC, 1998). This initiative is

an international collaborative project coordinated by an executive committee. The standard provides a set of fifteen elements that may be used to describe a web resource. DC has been rapidly accepted over a range of industry sectors and used as a foundation for other standards. The Picture Australia web site (<http://www.pictureaustralia.org/metadata.html>) is a good example of a Dublin Core implementation. They use XML to store and manipulate the metadata. Guidelines and examples are given for image providers to the web site to follow.

Standards in the educational field include the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE, 2000), the IMS Global Learning Consortium (IMS, 2001) in the U.S. and the Education Network Australia (EdNA, 2001) Metadata Standard which is based on the Dublin Core.

On a global scale the Learning Object Metadata Working Group (LOM, 2000) are working on a standard for “Information Technology - Education and Training Systems - Learning Objects and Metadata” known as the LOM Scheme. The working group are part of the Institute of Electrical and Electronics Engineers (IEEE, 2000). The standard is based on a proposal submitted by IMS and ARIADNE and includes elements of the EdNA standard. It is still a working draft and subject to change. The working group is one of many set up by the IEEE Learning Technology Standards Committee (LTSC, 2000), that is currently working on developing standards for the interoperability of learning resources. Another group, the Competency Definitions Working Group (Ostyn, 2000) are working on a standard for the description of competencies. The group, chaired by Claude Ostyn states that the purpose of the project is to:

“define a universally acceptable Competency Definition model to allow the creation, exchange and reuse of Competency Definition in applications such as Learning Management Systems, Competency or Skill Gap Analysis, Learner and other Competency profiles etc.”

The evolving nature of metadata schemas poses numerous problems for those trying to implement them. Alliances between areas of expertise and associated sectors are an effective way to promote consistency. University and industry groups are creating metadata schemas to represent their domain knowledge. Five universities including the University of Queensland in partnership with the Institute of Engineers, Australia (IEAust), the Centre for Mining Technology & Equipment (CMTE) and the Distributed Systems Technology Centre (DSTC) have developed the Australian Virtual Engineering Library (AVEL) (Talmacs, 2000, p. 5). The main objective of AVEL is to improve sharing of information between industry and university researchers. Their metadata is also based on the Dublin Core elements.

Another group of Australian universities is working on a metadata system to improve the access to research information in Theses and Dissertations. Based on the US Networked Digital Library of Theses and Dissertations the Australian Digital Theses project uses nine Dublin Core elements and a small number of qualifiers (qualifiers are applied to an element to make the meaning more specific). These are automatically applied to digital versions of the research resources (Talmacs, 2000, p. 6).

A range of options has emerged relating to the creation, storage, retrieval and communication of metadata. It can either be attached to a resource or stored separately in a database or metadata repository. Embedded metadata may be in the form of HTML META tags in a document heading or encoded in the content with XML tags. In his paper *Metadata in the University Environment* Talmacs (2000, p.

1) points out that a “considerable commitment is required to assess standards; train staff and utilise metadata appropriately”. The development of systems to automatically generate metadata can alleviate the often onerous task of creating tags and descriptions. Software tools such as the EdNA Metadata Toolset (Currie, 2000, p.1) are becoming available to assist with this task. Information extraction systems can also be used to automatically search legacy documents that have no metadata applied and extract the relevant metadata.

The Learning Object Metadata (LOM) and Dublin Core (DC) standards are used in the implementation of this project.

Ontologies

Standard metadata vocabularies provide a good basis for knowledge representation but they can also be frustratingly inadequate. As Swartout (1999, p 19) points out “one key impediment to sharing knowledge is that different systems use different concepts and terms for describing domains.” Research in the field of ontology offers new and interesting perspectives to this and other knowledge management problems. The word “ontology” stems from philosophy, in which it refers to the subject of existence. It is also associated with epistemology, which is about knowledge and knowing (Gruber, 2000). Its association with knowledge management originated in the field of Artificial Intelligence (AI). Even here there is confusion and controversy over its meaning, although, Tom Gruber’s definition seems to be the most widely accepted. He describes ontology as “a body of formally represented knowledge based on a conceptualisation”. Conceptualisation referring to an abstract simplified view of a domain of knowledge.

Ontologies come in various types according to their intended functions. However, there is no consensus in the current literature as to how these types are specified. For instance, Chandrasekaran, Josephson & Benjamins (1999, p.24) define two areas for ontology specification; *domain factual knowledge and problem-solving knowledge*, whereas Waterson & Preece (1999, p. 45) refer to *task or physical* ontologies. Guarino (1997) in his article *Understanding, Building and using Ontologies* discusses amongst others *application* ontologies, *generic and representation* ontologies. The name given to an ontology will generally imply its function. For instance a domain ontology would typically represent (objects, relations, events, states, and so forth) in relation to a particular domain such as breast cancer. A general ontology would represent generic concepts applicable to several domains and a task ontology would define an operational aspect such as scheduling or diagnosis.

A call for papers on Ontology by the IEEE Intelligent Systems Journal in January 1999 resulted in such an overwhelming response they had to expand their coverage of the topic over two editions. Swartout, the guest editor (1999, p. 18) predicts that: "Ontologies will fundamentally change the way in which systems are constructed...in the future, intelligent systems developers will have libraries of ontologies at their disposal". This is in sharp contrast to Uschold's paper "*Where are the Killer Apps?*" presented at the ECAI-98 Workshop on Applications of Ontologies and Problem Solving Methods, which questioned the dearth of ontology applications available at the time. The increase in interest in ontologies over the last two years can be seen in a number of areas and in particular where there is a need for the sharing and reuse of knowledge across heterogeneous systems and sources. They have been widely adopted by science and medical researchers (Lopez et al, 1999 &

Hendler & Stoffel, 1999). Altman et al (1999, p. 68) stress that the near exponential growth of knowledge in areas such as DNA sequence databases, has created a critical need for “technologies that represent and interpret multiple, diverse data sources and support collaborative scientific interpretation of these sources”.

In the education environment ontologies are being created by Artificial Intelligence (AI) researchers developing Intelligent Educational Systems (IES). These systems, also referred to as Intelligent Tutoring systems make extensive use of agent software. Agents can take several forms, for example they can simulate the role of a personal assistant and help a learner to complete tasks, or work together in a network (multi-agent system) to solve problems and interact dynamically with learners. In his article *Towards a Standardization of Multi-Agent System Frameworks* (1999) Flores-Mendez comments:

“It is important that agents not only have ontologies to conceptualize a domain, but also that they have ontologies with similar constructions. Such ontologies, when they exist, are called common ontologies. Once interacting agents have committed to a common ontology, it is expected that they will use this ontology to interpret communication interactions, thereby leading to mutual understanding and (ultimately) to predictable behaviors”.

Chen & Mizoguchi (1999, p. 1) use a common ontology to facilitate information exchange between computer Agents in their IES. They explain that in the development of IES systems the focus is to develop task dependent but domain-independent ontologies. This is achieved by examining “the information exchanged between the learner model agent and other agents, and abstracting the domain-independent concepts”. In this instance the domain refers to the subject being taught for example geometry or art. The domain-independent concepts identified and represented in their ontology will assist in creating standards for the communication between agents in educational systems.

Ontologies provide not only a vocabulary of terms but also the relationships between terms. This is an important consideration for the design of search mechanisms. The OntoSeek project (Guarino, Masolo, & Vetere, 1999) is a good example of how both lexical and ontological information can be used together to improve search results. The project uses WordNet, a linguistic ontology, and demonstrates how a search in the yellow pages catalogue from a flat list of word *senses* outperforms a search from a flat list of words. In WordNet a *sense* is gained by associating words with particular categories and synsets (terms grouped into semantic equivalent sets – each assigned to a lexical category such as a noun or verb). The OntoSeek project concluded that:

“at least for online yellow pages, the combined use of linguistic ontologies such as WordNet and structured representation formalisms can help an information-retrieval system to

- a) decouple the user vocabulary from the data vocabulary, by covering the most common English words;
- b) increase recall, by exploiting the hierarchy to make generic queries and recognising synonyms;
- c) increase precision, through the disambiguation mechanism and the ability to navigate the hierarchy to select specific queries; and
- d) further increase precision, by considering the structure of queries and descriptions.”

Several web based collaborative projects currently support the development and reuse of ontologies. The Ontobroker system developed by Decker, Erdmann, Fensel & Studer (1999) uses formal ontologies to extract and generate domain specific metadata. The Simple HTML Ontology Extensions (SHOE) system, developed by the Department of Computer Science at the University of Maryland, is

an ontology-based knowledge representation language that is embedded in web pages (Heflin & Hendler, 1999). Both projects provide tools, information and tutorials on how to use their systems. *The OntoEdit - Ontology Engineering Environment* (OntoEdit, 2000) is a commercial product under development by the Knowledge Management Group at the University of Karlsruhe Institute AIFB. It is based on XML and offers an easy to use GUI tool that will assist organisations to create their own ontologies.

It is important to note that one of the advantages of ontologies is the ease by which they can be expanded, adapted and updated. Once a domain ontology has been defined for a particular area of knowledge it can be expanded upon to represent concepts such as beliefs, goals, and predictions. The structure for the activities and terms associated with these concepts could be taken or adapted from an existing general ontology. For example the IEEE Standard Upper Ontology (Schoening, 2000) provides a structure for several lower level ontologies. The standard seeks to improve interoperability through ‘educational applications in which students learn concepts and relationships directly from, or expressed in terms of, a common ontology’.

The dearth of ontology applications available between 1995 and 1998, observed by Uschold may have been due in part to the lack of a suitable technology to represent ontological information. Ram, Catrambone, Guzdial, Kehoe, McCrickland & Stasko (1999, p. 41) created a Procedural Markup Language (PML) to implement their ontology based project *PML: Adding Flexibility to Multimedia Presentations*. PML is an XML application and their project is an example of how XML provides a framework for encoding ontological information. Research by Erdmann & Studer, 1999; & Fensel et al further supports this conjecture.

XML Technologies

The Extensible Markup Language (XML) was developed as a standard to help overcome the limitations of the Hypertext Markup Language (HTML) that is currently used for web publishing (Goldfarb & Prescod, 2000, p. 1iv). It is a new technology but is derived from the older and stable technology the Standard Generalised Markup Language (SGML). Although it is often referred to as a *language* it is actually a standard for creating languages (meta-language). Users who share the same domain knowledge can create vocabularies, using the XML language, to represent their particular domain. For example:

- Scalable Vector Graphics (SVG)
- MathML
- Chemical Markup Language (CML).

XML is actually more than one technology, it is a family of related technologies that continues to be expanded and improved. The family currently includes:

- *XML Namespaces* - used with Uniform Resource Identifiers (URI) for identifying vocabularies/schemas. URIs are based on Internet domain names but unlike Uniform Resource Locators (URLs), do not need to actually point to anything on the Internet. They can simply be a prefix that once applied to an XML element name will provide an unambiguous source for that name.
- *XLink and XPointer* - Xlink allows XML documents to associate metadata with a link and create links between other XML and non-XML resources. XPointer supports the location of fragments of XML within a document, including element types and attribute values.

- *Xpath* – a language for providing access to individual parts of XML documents. It is used by Xpointer and XSLT.
- *XSL* – Extensible Style Sheet Language, for creating style sheet for formatting and transforming XML documents into other document formats. The XSL Transformations (XSLT) language is used in conjunction with XSL.
- *XSLT* – used to manipulate, transform and output data from XML source documents.

The W3C recommendation Extensible Markup Language (XML) 1.0 Second Edition (XML 1.0, 2000) is the foundation for all of the XML family of technologies. It describes the syntax for XML documents and parser software that reads XML.

XML documents that adhere to certain grammatical rules in the XML 1.0 Specification are said to be well-formed (Hunter, 2000, p.27). This will allow software applications such as XML parsers to access the code. A Document Type Definition (DTD) sets further rules on how a document should be structured including the type of elements and the kind of data or values that can be used. An XML document that meets the standards of a specified DTD is said to be *valid*. A DTD can be an external document with the file extension .dtd, or it can be applied to a document internally. If it is external a number of documents can be associated with it through the use of a variable called DOCTYPE. It is not necessary for an XML document to be valid, however creating documents that conform to a known standard assists interoperability and reliable data transfer. DTDs are defined in the XML 1.0 specification, however they have been found to have certain limitations and XML Schemas have been introduced in an effort to overcome these limitations.

Schemas

The term *schema* is commonly used in relation to database systems to define the elements and relationships of the system. This section discusses the *XML Schema Candidate Recommendation* (Fallside, 2000) defined by the W3C. Fallside (2000, p1) defines *schema* as follows:

“A schema can be viewed as a collection (vocabulary) of type definitions and element declarations whose names belong to a particular namespace called a target namespace. The target namespace enables us to distinguish between definitions and declarations from different vocabularies.”

Note that in this instance the *target namespace* can be more than just a unique identifier. It can point to the actual schema that defines the boundaries of the elements and attributes. The schema could then be used for validation of the elements by a software application.

The XML schema will address many of the existing restrictions imposed by DTDs. The main advantages of a schema are as follows:

- *Simple to parse* - a schema is written in XML and can therefore be read by any XML application or tool including XML aware browsers. DTDs in contrast have their own syntax.
- *Schemas are extensible* - DTDs, like the Hypertext Markup Language (HTML) were not designed to be extensible. They were both derived from the Standard Generalised Markup Language (SGML). One of the main objectives of SGML was to standardise document representation and restrict changes. However, the dynamic multifaceted documents that make up the World Wide Web require a more flexible approach. Extensibility allows for the adaptation of existing

schemas to meet specific requirements and as such is an important aspect of document modeling.

- *Multiple namespaces* - a document can make use of multiple schemas through the use of XML namespaces. In contrast only one DTD can be associated with a document.
- *Stronger typing* - schemas provide improved options for defining Datatypes.

Currently, there are few applications that support schemas, however the W3C announcement of the advancement of the XML schema to Candidate recommendation Status on the 24th October, 2000, is likely to encourage developers and result in a range of parsers and tools becoming available. In light of the benefits they offer and the likelihood that they will be the chosen format for future web development, a schema was used in place of a DTD for the implementation of this project.

XSL and XSLT

The Extensible Style sheet Language (XSL) is a language for expressing style sheets. As Froumentin (2001, p. 1) from the W3C explains “it consists of 2 parts:

1. XSL Transformations (XSLT): a language for transforming XML documents.
2. An XML vocabulary for specifying formatting semantics (XSL Formatting Objects)”

XSLT is a declarative rule based language that deals with the transformation of information from one XML vocabulary to another. The two languages can be used together, with XSLT transforming the data and XSL formatting the display. The most common current application of XSLT is to facilitate interoperability in

Business to Business (B2B) interactions. Each business can have its own format for representing information in XML, when they need to transmit it outside of their organisation they transform it into a common form that others can access and transform according to the specific needs of each user. XSLT can also output information in HTML and ordinary text.

XHTML

The Extensible Hypertext Markup Language (XHTML, 2000) is a W3C standard released in January 2000. It is a reformulation of HTML 4 as an XML 1.0 application. It can therefore be interpreted by XML tools. The adoption of this standard by web developers will play a major part in the transition towards the semantic web envisaged by Berners-Lee.

The benefits gained by XHTML come at a cost, as the simplicity of HTML that allowed non-programmers to publish on the web will be lost to a more elaborate and stringent language. For example syntactical errors were ignored in HTML, but with XHTML just one syntax error will result in a document being rejected. Ladd & O'Donnell (2000, p. 296) also point out that XML requires several technologies working together to get the power, so in comparison to HTML there is a steeper learning curve. Annoying though this may seem, the strict format of XML allows for better computer understanding of information and improved computer to computer communication (Ladd & O'Donnell, 2000, p. 297). The separation between presentation and content provides more flexibility and easy editing. Web publishers using Cascading Style Sheet (CSS) technology with HTML would be aware of this benefit. Also the strict syntax leads to machine understandable information that is pivotal to knowledge provisioning. As Stabb et al explained HTML facilitated the

“transportation of information from sources to humans’ while XML technologies allow the “communication of knowledge form web resources to machines”. This in turn facilitates the use of powerful computational intelligence tools.

The XHTML standard gives other indications as to why web publishers should embrace the technology:

“Alternate ways of accessing the Internet are constantly being introduced. Some estimates indicate that by the year 2002, 75% of Internet document viewing will be carried out on these alternate platforms. The XHTML family is designed with general user agent interoperability in mind.”

The term user agent refers to software programs that request web information from servers to display (browser) or manipulate in some way.

In some instances XML can be used to extend existing systems. It can be used in conjunction with legacy applications and databases. For example the Oracle XSQL Servlet for Java generates XML documents from SQL database queries. (GoldFarb & Prescod, 2000, p.462). Using XML to compliment traditional database systems offers several technical advantages. Currently most information transactions take place in a server client architecture whereby the server controls proprietary data sources and serves information to clients, namely users on their personal computers. This system can be slow as each query is handled by the server and transported back to the client. XML provides a third layer to this architecture that allows information to be manipulated on the client side, resulting in improved customer experiences and more efficient use of Internet technology. Once an initial query has been made to the server either through XML documents or an ODBC database connection all the relevant data is transferred to the client machine in XML format. Client applications or scripting languages such as JavaScript or Visual Basic can then manipulate the data.

RDF

While XML is ideal for metadata exchange it fails to provide a broad framework for describing and organising documents and resources. A lack of such a framework has delayed the development of web technologies such as content rating, digital signatures and intellectual property claims (St. Laurent & Biggar, 1999, p208). XML is human and machine readable, but fails to meet the needs of higher order applications such as computer Agents that require a machine understandable format. The Resource Description Framework (RDF) is an application of XML designed for the description and interchange of metadata that is machine understandable and thus offers options to overcome these restrictions.

Although the web provides access to millions of resources the organisation of these resources is nearly non-existent. Most search engines use full-text searching and follow links between documents to extract information. This is very demanding on bandwidth and processing power and results can be wildly inaccurate. The HTML META tag, was introduced to allow authors to provide keywords and descriptions for their resources. However, many authors fail to apply metadata and even when they do there is confusion through incompatible vocabularies (St. Laurent & Biggar, 1999, p211).

The RDF model provides a basic but powerful framework for resource metadata. Documents are considered resources and the information about them is considered as properties, which themselves can be resources. In fact a resource is not strictly a document it can be anything from a person to a chunk of information. RDF uses Uniform Resource Identifiers (URI) to uniquely identify resources and the XML namespace function is used to point to the URI of a resource. Each resource has a minimum of three properties: `rdf:subject`, `rdf:object` and `rdf:predicate`. Directed

labeled graphs that use nodes and arcs are used to represent the relationships between resources. Nodes that hold a resource have an oval shape, nodes that contain a String Literal have a rectangle shape. Lines joining the nodes (the predicate) have an arrow pointing from the subject to the object.

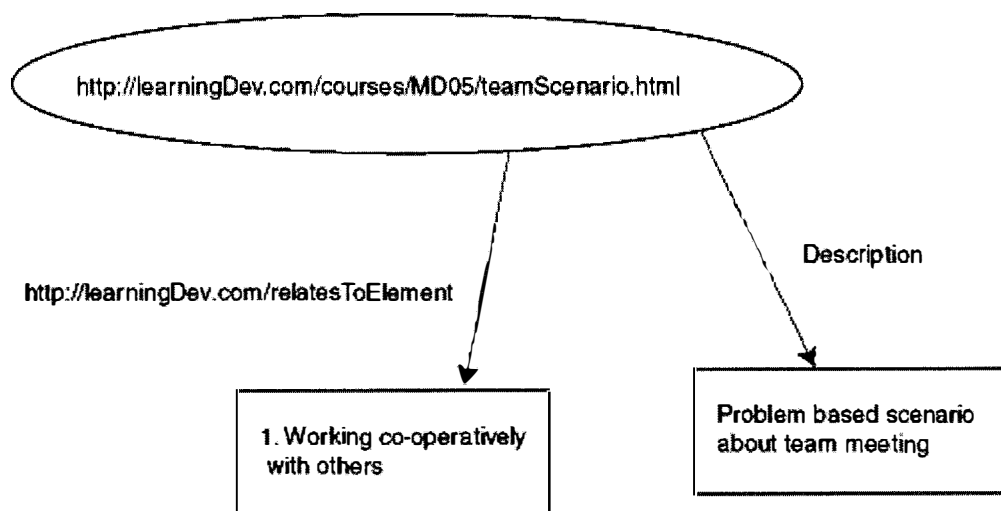


Figure 1. RDF node and arc graph

RDF Schemas

An RDF Schema is based on the Resource description Framework Schema Specification (Brickley & Guha, 2000) and performs the same function as an XML Schema or DTD. An RDF schema is associated with instance documents using namespaces. For example the Simple RDF Parser and Compiler SiRPAC, discussed further in Chapter 3, can be configured to automatically fetch corresponding RDF schemas from their declared namespaces.

RDF schemas specify class hierarchies and constraints. In this way the RDF schema system is similar to that of Object-Oriented (OO) programming languages. Yet it differs from classical OO systems in that it is *property* centric as opposed to

class centric. As the RDF Candidate Recommendation specification explains:

“instead of defining a class in terms of the properties its instances may have, an RDF schema will define properties in terms of the classes of resource to which they apply.”

For example in the RDF schema created for this project the property *containsPC* has a range of *PerformanceCriteria* and a domain of *ElementCompetency* as follows:

```
<rdf:Property ID="containsPC">  
  <rdfs:domain rdf:resource="#ElementCompetency"/>  
  <rdfs:range rdf:resource="#PerformanceCriteria"/>  
</rdf:Property>
```

In a classic OO system this would be defined as a class *ElementCompetency* with an attribute *containsPC* and a type of *PerformanceCriteria*.

The two primary benefits of RDF are:

1. it provides a simple but extensible model for resource metadata; and
2. it will facilitate significant improvements in retrieval techniques.

Iannella (1998, p. 7) claims that RDF will “allow the resource description communities to primarily focus on the issues of semantics rather than the syntax and structure of metadata”. The Education Network Australia (EdNA) Online service is an example of how RDF can be implemented to manage large resource repositories. The EdNA Online service is based on a concept of distributed ownership and administration. Part of their initiative is to formalise a process for participants who contribute resources to the EdNA Online Directory. To automate the process of collecting new resources and maintaining the associated metadata, they have developed a Metadata Toolset and a Harvesting Robot (software application) to mine

their resource repositories. The Toolset assists participants, known as 'Registered Contributors', to apply and maintain their metadata by allowing them to:

- detect and analyse existing metadata for a site
- add metadata to local and remote resources
- save the metadata into the resource or to a separate repository, in either HTML or RDF format
- simultaneously create/edit common metadata elements in multiple resources
- implement different metadata schemas and thesauri as selected by the user and
- generate a metadata collection for the EdNA repository in RDF format.

The Harvesting Robot can then extract information from the RDF metadata files on the Registered Contributor's public website.

RDF's machine understandable format will lead to improved search and data mining applications. There are a number of research projects currently focussing on the query languages and applications to exploit RDF metadata. In their paper *Enabling Inferencing*, Guha et al propose that a query language should be expressed in terms of the RDF logical data model rather than one particular syntax.

Karvounarakis, Christophides, & Plexousakis, take a similar approach with their query language called RQL. It is based on a graph model that captures the RDF modeling primitives. Decker et al, have developed the Simple Logic-based RDF Interpreter (SiLRI) application based on Frame Logic and Marchiori and Saarela propose a query language called METALOG based on datalog (a subset of logic

their resource repositories. The Toolset assists participants, known as ‘Registered Contributors’, to apply and maintain their metadata by allowing them to:

- detect and analyse existing metadata for a site
- add metadata to local and remote resources
- save the metadata into the resource or to a separate repository, in either HTML or RDF format
- simultaneously create/edit common metadata elements in multiple resources
- implement different metadata schemas and thesauri as selected by the user and
- generate a metadata collection for the EdNA repository in RDF format.

The Harvesting Robot can then extract information from the RDF metadata files on the Registered Contributor’s public website.

RDF’s machine understandable format will lead to improved search and data mining applications. There are a number of research projects currently focussing on the query languages and applications to exploit RDF metadata. In their paper *Enabling Inferencing*, Guha et al propose that a query language should be expressed in terms of the RDF logical data model rather than one particular syntax.

Karvounarakis, Christophides, & Plexousakis, take a similar approach with their query language called RQL. It is based on a graph model that captures the RDF modeling primitives. Decker et al, have developed the Simple Logic-based RDF Interpreter (SiLRI) application based on Frame Logic and Marchiori and Saarela propose a query language called METALOG based on datalog (a subset of logic

programming). Each approach has benefits and drawbacks, for example SiLRI is currently unable to deal with container classes; rdf:Bag and rdf:Seq. It is envisaged that a lot more research will be required in this area before the full potential of RDF can be realised.

RDF has also been criticised for its complexity and inability to deal with some types of metadata requirements. It is excellent for describing properties of web pages but describing detailed relationships between real world objects can result in an unmanageable web of nodes and arcs (Klyne, 2000, p. 2). There is, therefore, still a lot of issues to be resolved with this technology, yet it is likely that it will play an important part in future metadata processing particularly in the areas of:

- cataloguing
- resource discovery
- information exchange between intelligent software agents
- intellectual property rights
- privacy preferences
- digital signatures.

Summary

This Literature Review has covered emerging concepts in knowledge management, the status of digital metadata and the general aspects of the technologies required for this project. In the following chapters, where necessary, further details are discussed.

CHAPTER 3

Methodology

Introduction

The implementation process required three distinct stages: analysis of the sample data, application of relevant metadata and running tests using the selected software. The first step was to choose an appropriate data sample that would provide an opportunity to explore the benefits of sharing and reuse. This data sample was then analysed to determine who would be interested in using it and how. This information informed the metadata requirements for stage 2 and determined the test criteria for the third stage of the implementation. In stage 2 the metadata required was selected and applied to the data sample in XML format. A metadata schema, sample ontology and RDF schema were also created as part of the process of defining the metadata. This stage resulted in 2 test files, an XML version of the data sample and an RDF file describing it as a resource. 3 additional files; a style sheet for displaying the XML, and a query and rule file for the inference engine were created for the execution of the tests in stage 3.

The following is a list and brief overview of the software applications used to implement stage 3.

XML Notepad

Like HTML XML documents can be created in a simple text editor or they can be created using specialised editors. Microsoft's XML Notepad (XML Notepad,

2000) as seen in Figure 2, provides a GUI interface that makes creating and editing XML documents easy.

More advanced products such as XMLwriter (XMLwriter, 2000) which provides functions such as checking for 'well formedness' and DTD validation are beginning to appear as shareware or commercial software (Ladd et al, 2000, p322).

XML Notepad was sufficient for this project, as DTD validation was not required. It was also readily available and easy to use.

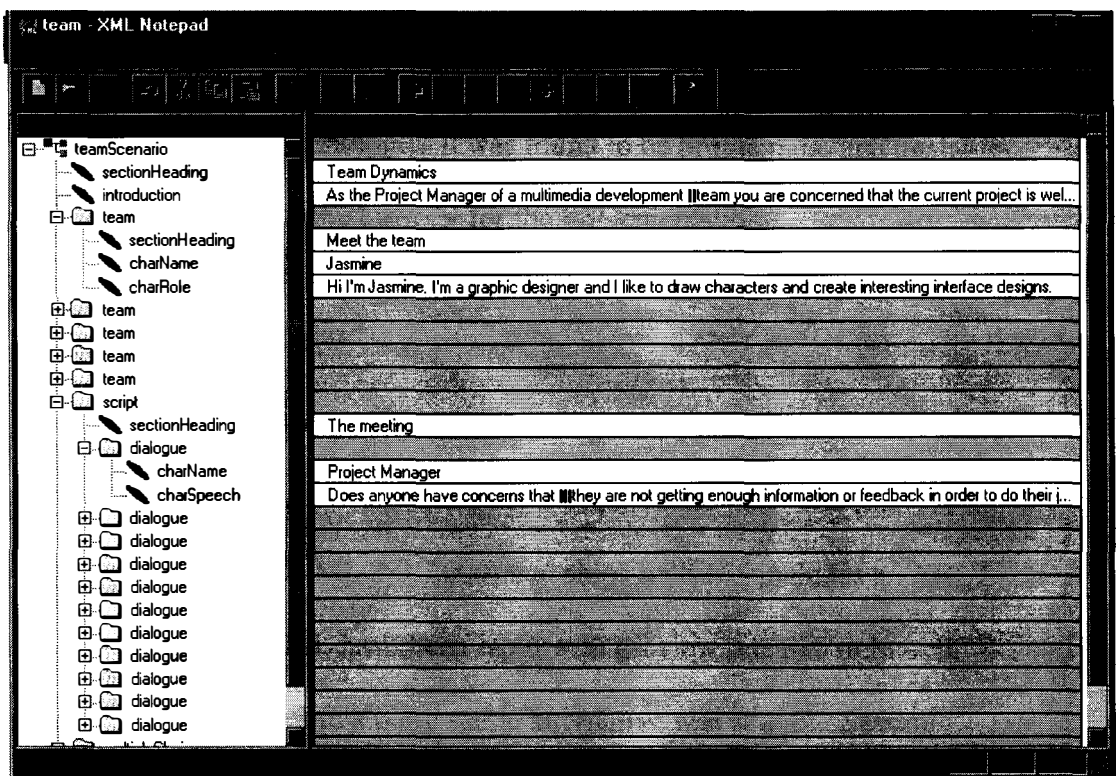


Figure 2: XML Notepad.

Internet Explorer 5.5

XML can be viewed in Internet Explorer IE5. The latest version 5.5 has significant updates in support for XML related technologies and it is expected that this trend will continue as the technology matures. When an XML document is

opened in IE5 a default style sheet is applied. The tags are colour coded for easy reading and small plus and minus signs allow the document structure to be expanded or collapsed. A style sheet can be associated with an XML document by adding the following processing instruction element.

```
<?xml-stylesheet type="text/xsl" href="team.xsl"?>
```

The style sheet in this example is *team.xsl* and it is stored in the same directory as the XML document. However, browser support for XSL is still poor. Netscape 6 does not appear to recognise XSL and relies on Cascading Style Sheet (CSS) technology currently used to format HTML. Although Netscape has been reported as having extensive support for XML technologies it was found to be problematic and unstable and thus IE5.5 was selected as the browser for the implementation.

XT

Three tests were required in the third stage of the implementation. Test 1 and 2 both use XSLT technology. An XSLT engine is required for this. The Microsoft XML Parser (MSXML), (MSXML 3.0, 2000) that ships with IE5 includes an XSLT engine. Updates to the MSXML have been rapid with preview releases in March, May and July of 2000. Therefore a more stable alternative was sought. Several XSLT engines are also available over the Internet. The XT engine (Clark, 1998, p.1) was written by James Clark and is available as a Win32 executable, in Java or as a Servlet. The Win32 executable version of XT was selected for the implementation.

XT accepts an XML source file and an XSL style sheet containing XSLT code as input. It uses these two documents to output an HTML document. It can also output other document formats.

XP, SAX and DOM

An XML file is essentially a succession of characters that can be read by a text editor or parser. Before a computer can extract any logical meaning from these characters it needs a way of converting them into a logical structure (Ladd & O'Donnell, 2000, p. 334). A *parser* separates the elements (tags) from the information contained within them and produces output that another piece of software can use. There are several parsers currently available such as the Microsoft MSXML parser mentioned previously, the xml4j by IBM and XP developed by James Clark, which was used for this project.

The next piece of software involved is usually an Application Programming Interface (API) such as the Simple API for XML (SAX), (SAX 2.0, 2000) or the Document Object Model (DOM), (Le Hégarret, 2001, p. 1) which in turn is used by an application to process the information. This process can be seen in the following diagram.

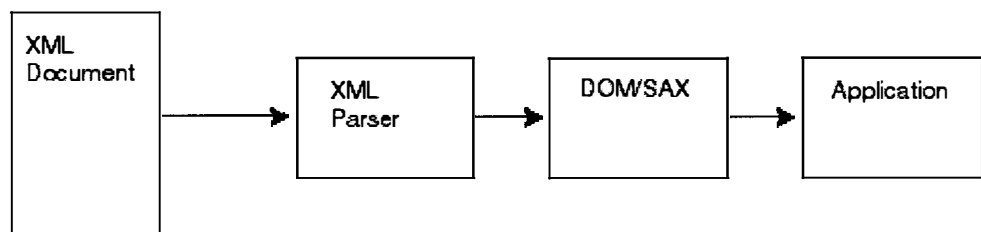


Figure 3: Processing XML.

The DOM and the SAX both achieve the same thing but they work in different ways. Skonnard (2000, p. 1) explains:

“The DOM models the Infoset through a hierarchy of generic nodes that support well-defined interfaces...SAX, on the other hand, models the Infoset through a linear sequence of well-known method calls.”

Each method has its benefits and drawbacks. SAX has not been approved by a standards body but has been adopted by the majority of XML tool vendors including Microsoft who have included support for it in their MSXML library. The latest release SAX 2 adds support for XML Namespaces, which are required for standards such as the Extensible Style sheet Language (XSL), XML Schemas and RDF. It is considered to be the most lightweight of the two and is more suited to large documents.

The DOM on the other hand has been sanctioned by the W3C and its tree-based approach is better suited to some applications. SAX as a stream-based method would work better for file transfer while DOM as a tree is better structurally. Skonnard suggests that the best results may be obtained by using a combination of the two “SAX can be used to build DOM trees (or more interestingly, portions of DOM trees). Conversely, developers can traverse DOM trees and emit SAX streams”. These technologies were explored as part of the implementation of stage 3.

RDF, SiRPAC and SiRLI,

RDF parsers build on XML parser capabilities by translating the XML representation of RDF data into an abstract form of the RDF data model. The base of the RDF data model is the triple: a subject resource and an object resource linked together by a predicate resource.

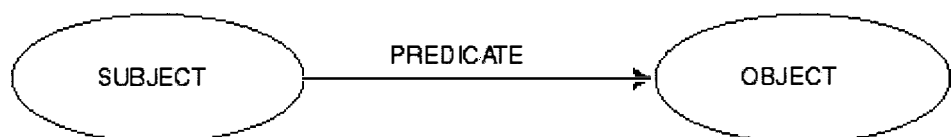


Figure 4: RDF model.

There are several RDF parsers available on the Internet as open source or pre release versions. One of these, the Simple RDF Parser and Compiler (SiRPAC) created by Janne Saarela (1999) has been selected for the implementation. It is a set of Java classes that use the SAX interface and a SAX compatible XML parser to parse RDF files and produce triples.

Once the RDF information is in 'triple' form it can be used by software applications. Stefan Decker, Dan Brickley, Janne Saarela and Jurgen Angele have developed the Simple Logic-based RDF Interpreter (SiLRI). This software is an inference engine and can be used to show how rules based on an ontology can be used to infer additional meaning from source data. It was available under the general Open Source license until November, 19th, 2000. It is now being commercialised but a restricted version is still available under a limited open-source license for non-commercial use (Decker, 2000, p. 1). SiLRI can be used to manipulate RDF information through the use of Frame-Logic rules (discussed further in Chapters 4), and queries. It was used in conjunction with SiRPAC to demonstrate ontology principles relevant to the data sample.

Summary

During the implementation XML Notepad was used to create and edit an XML document. In tests 1 and 2 the information in this document was transformed using a style sheet and the XT engine. XP, SAX and the DOM were also used to extract individual information components from the document. Test 3 used SiRPAC and SiLRI to manipulate metadata encoded in RDF.

It should be noted that the technologies used in the implementation are constantly evolving and more appropriate solutions may eventuate over time.

CHAPTER 4

Implementation Stage 1

Introduction

In this chapter the data sample is introduced and analysed to determine the knowledge it holds from a range of user viewpoints. This analysis was used to inform both guidelines for the metadata requirements for stage 2 and test criteria for stage 3, thus providing set goals for the desired output of the implementation.

Data sample

The data sample selected for this project is a problem based learning activity that represents a typical component of an online training product. It was in HTML format and contained some JavaScript to facilitate a multiple-choice interaction. It contained no metadata and since multimedia is not covered in the scope of this project, no media items. The complete file can be viewed in Appendix 1.

This data sample was chosen because it contains information that had the potential to be reused or adapted to suit other learning products or training environments. The challenge was to create and apply metadata that can accommodate the needs of the various parties associated with the development and implementation of this type of data, thus leveraging the knowledge it represents.

In Chapter 1 the word multivalent is used to describe the range of meaning that can be inferred from a sample of information. Identifying the meaning unlocks the knowledge. It converts the raw data into the *high value information ready to*

apply to decisions and actions as described by Prusak and enables the *knowledge provisioning* discussed by Stabb et al. Finding the meaning in the data sample involved viewing it in as many different perspectives as possible. This included considering it as a whole and in sections, as a discrete component and as a subset of some larger whole, and most importantly from the viewpoint of who would be interested in the information and for what reason.

By analysing the data sample in this way the following conclusions were made. As a whole the information would be of interest to trainers who wished to impart the concepts represented to their students. It would be of interest to instructional designers as an example of the learning strategy employed and the method of presentation. A developer or web publisher would be interested in the layout of the information, the JavaScript and HTML employed. A learner undertaking teaching in team work and communication skills would find the problem and the related article that it references relevant to their studies.

These same people may only want to use some of the information and not all of it. The trainer may be interested in the entire multiple-choice question, while the instructional designer may want to use only the question posed and not the options or feedback provided.

Prospective users would also be interested in information that was not currently available in the data sample such as the creator; the date of creation and the course it belonged to if any and technical issues such as the minimal computer specifications required to implement it. More specific pedagogical properties may also be relevant to a seeker such as the level of difficulty, or prerequisites.

Having considered who would use the information it was necessary to establish what they could possibly do with it. The JavaScript for the multiple choice

could be re-used by replacing the text in the statement, options and feedback. This is a fairly simple section of code but the same principle could be applied to more programming intensive interactions.

An Instructional Designer could reuse the *scenario* for another course by simply changing the setting. For example instead of a team of multimedia developers as used in the sample they could have a team of hospitality workers running a restaurant. Many of the elements of competency that training packages are based on are centered on generic principles that can be applied to a range of industry sectors. For example the team work, communications and report writing in the data sample are also applicable to Marketing, Management, and Hospitality industries. Learning activities designed to address these elements would differ only in the setting that they need to represent. The ability to quickly locate and customise these activities would reduce redundancy and development time for those involved in creating online training products.

Registered Training Organisations (RTOs) that purchase an online training product may want to customise it to meet their particular industry sector or user needs. This can be achieved by changing examples, graphics and sections of the text. For instance in our data sample the multimedia team are developing instructional material. This may be completely irrelevant to a production house that specialises in e-commerce. They would however have the same interest in team dynamics, communication and report writing. They could adapt this problem-based scenario by changing character roles and dialogue to better reflect their industry setting.

Having identified ways in which the information could be adapted and reused some thought was given as to how the various users could locate the information components they need, this being the fundamental role of the metadata – to map the

information to its meaning. The meaning to be defined by the metadata was grouped into 3 areas that would be used as guidelines for stage 2.

Metadata guidelines

1. *administrative* – information about who created the data and when. What its intended purpose was, whether it relates to other components or is part of something else, when it was last updated and so forth.
2. *technical* – information about the physical structure such as the programming language used and browser compatibility.
3. *conceptual* – information about the concepts represented such as the learning outcomes, associated elements of competency, the instructional strategy used and the metaphor or story line.

Metadata would be required at various levels of granularity to identify this information since some relates to the resource as a whole and some relates to individual components of the data sample.

The 3rd area, conceptual information, posed a further interesting challenge. The learning outcomes of a training package are structured around competencies and performance criteria. These are structured into logical groups. For instance in the sample data the performance criteria could come from the following Elements of Competency:

- Working cooperatively with others
- Communicating in the workplace
- Documents and reports.

Training package developers and teachers tend to follow this logical grouping, creating activities and assessment tasks to match each element of competency. However, in most real life situations workers would draw on a number of competency skills while carrying out a single work activity. There is therefore a need to reflect this holistic approach in the design of learning activities and assessment tasks. The following feedback from the evaluation of a training package shows the importance of this to industry trainers:

“...not one mention anywhere of integrated competency assessment or holistic assessment of competencies – the lynch pin, in my opinion of assessment for this training package and industry requirements. I have a real problem with this!!! It is an endorsed requirement of the training package – there is no compromise!!”

(Melanie Burke, Research Officer, Retail & Business Services Training NT Industry Training Advisory Board , October, 2000)

There are also situations when it is easier to work with the elements of competency and the associated performance criteria in their logical structure. For example, an RTO may wish to recognise prior learning. If a learner can demonstrate that they have good communications skills they may not be required to undertake this particular element of competency. When a course has been developed logically it is very easy to determine which part of the course they should omit. However, when the course is structured around holistic activities the individual elements of competency are not so clearly defined.

It is therefore necessary to be able to view/source the learning outcome from two perspectives:

- by each outcome in its logical setting as laid out in the training package; and
- by outcomes grouped together to facilitate a holistic approach.

In other words to source the learning outcome knowledge from the sample data users need to be able to search for it in two different ways:

1. using the language of the elements of competency such as ‘communication’ and ‘work cooperatively’; and
2. conceptual language such as ‘team work’ and ‘meeting’ that encapsulate the holistic version of the learning outcome: identifying communication issues during a team meeting.

Test Criteria

The analysis of the data sample indicated that the desired functionality that was required from the implementation in stage 3 was:

1. *The ability to identify and reuse information components* – this would be tested by attempting to single out individual sections of the data sample and reuse them in some way.
2. *The ability to easily customise the information* – this would be tested by attempting to remold the sample so that it would be applicable to a different industry setting. Of particular importance here, is the ease by which this can be done.
3. *The ability to extract conceptual information* – the data sample is based on elements of competency but for search purposes it would be advantageous if it could also be sourced using a more abstract term or *theme*. This test, therefore requires a search for the abstract term ‘team work’ to also yield the related

element of competency ‘work cooperatively with others’ and thus the related resource.

Having created guidelines for the metadata and test criteria for stage 3, the next step was to create and apply the metadata.

Implementation Stage 2

In this stage existing metadata identifiers are created, using current standards and schemas, and applied to the sample data.

Schatz (2000, p. 13) states that “The only way you can easily call up useful knowledge bits is by having a useful tagging schema”. Before a useful tagging schema can be defined it is necessary to consider who will use the knowledge contained in data and what they will use it for. The guidelines developed in stage 1 of the implementation were followed to ensure that the metadata created would result in the *useful tagging schema* that Schatz recommends.

Gilliland-Swetland (2000, p. 4) arranges the possible functions metadata can perform into distinct categories – administration, descriptive, preservation, use, and technical. In stage 1, the guidelines established for the metadata defined similar categories: administrative, technical and conceptual. All these categories relate to the meaning or logical aspect of the sample data. There is also however a need to consider the size or structural aspect of the data. A sample of digital information can be deconstructed into infinite levels of granularity. The size, boundaries and nomenclature applied to the granularity is variable. As Peereboom (1998, p. 1) points out:

“The vocabulary used for entities of Web information is ambiguous and largely derived from the printed environment, (e.g. "title page", "document", "publication", "version", "edition" etc.) which is not always adequate.”

Similarly an entity of information may also be referred to as a chunk, a bit, a digital object or a component. The Learning Object Metadata (LOM) Scheme Version 3.5 (LOM, 2000) defines an element *Aggregation level* to identify the granularity of a digital component using the following numerical scale:

- 0 A single media item or piece of raw data.
- 1 A collection of items, such as a web page with text and graphics.
- 2 A collection of web pages, that group logically together (activity).
- 3 The largest level of granularity such as an entire course.

As with the logical form, the level of granularity that metadata should be applied to will be influenced by the needs and viewpoints of the end user and is therefore once again multivalent in nature. For instance an Instructional Designer may be looking for a learning activity that relates to communication. This requires descriptive metadata that applies to the entire learning activity. At a smaller level of aggregation, an Instructional Designer may be looking for a single graphic to reuse. A programmer creating the multiple-choice interaction would need to identify the relevant components: the statement or question, the individual options and their associated feedback. Thus metadata for the data sample was required at multiple levels of granularity. The highest identifies the resource as a whole and the lowest identifies the individual information components such as the question options. The highest level will be considered first.

Standards and schemas

When designing metadata it is wise to build on existing standards, vocabularies or schemas to enhance interoperability with applications and other interested parties. This also prevents duplication of work and assists in dissemination in a wider sphere. There are several organisations currently defining metadata schemas and standards that relate to education and training. Of particular interest to this project is the Learning Object Metadata (LOM) Schema Version 3.5. The LOM is a Standards Draft produced by a working group at The Institute of Electrical and Electronics Engineers (IEEE, 2000). It is still a working draft and subject to change. The working group is one of many set up by the IEEE Learning Technology Standards Committee (LTSC, 2000), that is currently working on developing standards for the interoperability of learning resources. LOM specifies “the syntax and semantics of learning object metadata, defined as the attributes required to fully and adequately describe a learning object”. Thus it provides both a conceptual structure for metadata and technical specifications such as data types and field lengths. The scheme provides examples of the applications it is relevant to including: “computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, web-based learning systems and collaborative learning environments”. The working group have created a set of core elements that can be used to describe a learning object and grouped them into nine meaningful categories. They are outlined in the scheme as:

1. *General* - groups all context-independent features plus the semantic descriptors for the resource.
2. *Lifecycle* - groups the features linked to the lifecycle of the resource.

3. *Meta-metadata* - groups the features of the description itself (rather than those of the resource being described).
4. *Technical* - groups the technical features of the resource.
5. *Educational* - groups the educational and pedagogic features of the resource.
6. *Rights groups* - the features that deal with the conditions of use for the resource.
7. *Relation* - groups features of the resource that link it to other resources.
8. *Annotation* - allows for comments on the educational use of the resource.
9. *Classification* - deals with characteristics of the resource described by entries in classifications.

These nine categories form what is called the *Base Scheme*. Each category contains a number of elements and sub-elements that can be further refined by qualifiers. A qualifier can be a defined vocabulary or a list of suggested best practice values.

The LOM scheme was used for part of the data sample metadata. Firstly administrative metadata was created using elements from the LOM general purpose category. The elements were applied using XML tags as follows:

```
<general>
  <identifier>MD05</identifier>
  <title>Team Scenario</title>
  <language>en</language>
  <description>Problem based scenario involving a team meeting in a multimedia
  environment
  </description>
  <aggregationLevel>1</aggregationLevel>
</general>
```

The <identifier> element is a unique identifier for the resource. It could be manually applied or generated automatically by a metadata management system. The <title> element contains the title given to the data sample HTML page and the <language> element specifies the language used. The <description> element allows 2048 chars

to be used to describe a resource. It corresponds to the Dublin Core element DC.Description which is one of the 15 DC elements for describing web resources discussed in the Literature Review. The <aggregationLevel> element as mentioned previously refers to the functional size of the resource. Since the sample data is a single web page the level 1 is appropriate.

The LifeCycle category provides identifiers that relate to the life cycle of the information such as version number and creator. It was used as follows:

```
<lifeCycle>
  <contribute>
    <role>Instructional Designer</role>
    <entity>Joe Smith</entity>
    <date>2000-05-11</date>
  </contribute>
</lifeCycle>
```

The <contribute> element and its sub-elements are used to describe persons or organisations involved in the creation of the resource including edits and publication. The <role element> has a best practice list that includes “Author, Publisher, Unknown, Initiator, Terminator, Validator, Editor, Graphical Designer, Technical Implementer, Content Provider, Technical Validator, Educational Validator, Script Writer and Instructional Designer”. The <entity> element contains the contributor’s name.

Any software or hardware restrictions can be identified using the technical category. The following elements were applied to the data sample. The <size> element contains the size of the data sample in bytes and must be represented in numbers only. Using the assumption that the HTML resource will be converted to XML (as metadata is applied at the lower level of granularity), technical requirements have been added to indicate IE5 and Netscape 6 as the minimum browser standard.

```

<technical>
  <format>XML</format>
  <size>5865</size>
  <requirements>
    <type>Browser</type>
    <name>Internet Explorer</name>
    <minimumVersion>5</minimumVersion>
  </requirements>
  <requirements>
    <type>Browser</type>
    <name>Netscape</name>
    <minimumVersion>6</minimumVersion>
  </requirements>
</technical>

```

The next set of metadata contains descriptive information specific to the educational category.

```

<educational>
  <interactivityType>expositive</interactivityType>
  <learningResourceType>problemStatement</learningResourceType>
  <interactivityLevel>Very Low</interactivityLevel>
  <semanticDensity>Medium</semanticDensity>
  <context>Industry training</Level>
  <difficulty>Medium</difficulty>
</educational>

```

The `<interactivityType>` element has a restricted vocabulary of active, expositive, mixed or undefined. The scheme states that documents are considered *active* if they contain simulations or questionnaires. It is presumed that any form of interactivity would also come under this description. The data sample includes a multiple choice question that allows interaction, however it represents only a small portion of the whole so *expositive* was used. Expositive includes hypertext documents, video clips and graphical content. The `<learningResourceType>` element has an open vocabulary with best practice values, `problemStatement` being the most appropriate for this resource. The `<interactivityLevel>` ranges from Very Low to Very High and the `<semanticDensity>` refers to the amount of information conveyed as compared to the size and duration of the resource. The `<context>` element describes the principal environment within which the resource will be used and includes values such as Primary Education and Vocational Training. The amount of

difficulty is judged in consideration of the end user and a restricted vocabulary of 1 to 4 is applied, with 1 being very easy and 4 being very difficult.

The Relation category was used to indicate a relationship between the learning activity in the sample data and the training package it was part of. It was also used to provide an association with the article 'Team Dynamics' by Steve Barnsworth' referenced in the introduction section of the data sample.

```
<relation>
  <kind>IsPartOf</kind>
  <resource>Certificate 2 in Media Studies</resource>
</relation>
<relation>
  <kind>Requires</kind>
  <resource>Team Dynamics Article</resource>
</relation>
```

The <kind> element is qualified by the Dublin Core list of relationships including IsBasedOn, HasPart, IsVersionOf, HasVersion and IsReferencedBy.

Extensibility

It is to be expected that specifications such as LOM will not cater for all the needs of individual organisations. For example there is no standard element in the LOM to describe the learning strategy applied to the resource. The IMS Global Learning Consortium (IMS, 2000) is a member organisation and key contributor to the IEEE standards committee. The IMS Metadata Extensibility Proposal (Ostyn, 1998) acknowledges that metadata schemes must be extensible and proposes that extensions within the core elements are the best way to achieve this. Thus a sub element <strategy> can be added to the previous educational category.

```
<educational>
  <interactivityType>expositive</interactivityType>
  <learningResourceType>problemStatement</learningResourceType>
  <interactivityLevel>Very Low</interactivityLevel>
  <semanticDensity>Medium</semanticDensity>
  <context>Industry training</Level>
  <difficulty>Medium</difficulty>
  <strategy>Problem based learning</strategy>
</educational>
```


More than one scheme or vocabulary can be used to describe an individual resource. For instance the Dublin Core (DC) elements could be used together with LOM elements. Different schemas might use the same tag to identify different elements and hence potentially change the semantics. XML namespaces and Uniform Resource Identifiers (URI) are used to prevent this. URIs are based on Internet domain names Uniform Resource Locators (URLs), but unlike URLs do not need to actually point to anything on the Internet. They are simply a prefix that once applied to an element name will provide an unambiguous source for that name. Since URIs can be quite long, XML documents could become verbose and difficult for humans to read. XML namespaces address this by allowing the URI to be specified in full only once and associated with elements and attributes by means of a prefix. Namespaces are declared using the reserved XML attribute *xmlns*. The following is an example of how this can be done. It shows two namespace declarations that associate the namespace prefix *dc* with the namespace http://purl.org/metadata/dublin_core (DC), and the prefix *lo* with the namespace name <http://ltsc.ieee.org/wg12/index.html> (LOM).

The prefixes are then used to uniquely identify the *title* and *comment* elements.

```
<dataSample
  xmlns:dc="http://purl.org/metadata/dublin_core#"
  xmlns:lo="http://ltsc.ieee.org/wg12/index.html#">
  .....
  <lo:title>Team Scenario</title>
  <dc:comment>Can be adapted to other industry settings</dc:comment>
  .....
</dataSample>
```

It should be noted that as XML develops the namespace facility will grow in importance and may point to schemes that software applications can use to verify and manipulate the data.

The amount of metadata created for the data sample thus far provides administrative information at the highest level of granularity that will assist in the identification and extraction of the knowledge contained. This section of metadata can be viewed in Appendix 2. The next section of metadata applies to the lower levels of granularity.

XML schema

The sets of elements defined by the LOM or by the DC are not suitable for describing the components of the data sample at the lower levels of granularity, such as a question option or a section of dialogue. The metadata guidelines developed in stage 1 showed that identifying components at this level is necessary therefore a new set of metadata elements were created specifically for this purpose. In the interest of consistency and future interoperability, it was advisable to create an XML schema to represent the knowledge contained in the data sample components. A schema is a data model and documents that are based on the model are considered *instances* of the model. As discussed in the Literature Review an XML schema address many of the existing restrictions imposed by DTDs and it is anticipated that they will play an important role in future web developments.

In his discussion paper regarding web technologies for Vocational and Educational Training (VET) materials, Gray speculates that a schema defined for student records and used by all training providers would overcome data-sharing problems currently experienced with heterogeneous databases with different fields and data formatting. The document provides the following recommendation:

“It is recommended that the development of XML schemata required for the Australian VET sector be commenced. These developments should accompany all relevant initiatives in the VET Sector. This concept was further extended by other Working Groups to an electronic student record and the XML schema to support it.”

The schema for the data sample was developed using the *XML Schema Part 0: Primer* (Fallside, 2000) which is a guide to using the XML Schema language defined in the W3C Candidate Recommendation 24 October 2000.

Elements defined in the schema fall into two main categories *complexType* and *simpleType*. Complex types may have attributes and elements in their content whereas simple types have neither. The following extract is an example of a *complexType* definition.

```
<!--teamScenarioType describes the structure of the scenario. It contains five subelements. The question element has 'minOccurs' set to 0 and is therefore optional within an instance. The other four subelements use the default of 1 and are therefore compulsory. The compulsory elements must be used in the order specified and be of the types specified-->
```

```
<xsd:complexType name="teamScenarioType">
  <xsd:sequence>
    <xsd:element name="introduction" type="xsd:string"/>
    <xsd:element name="team" type="characterType"/>
    <xsd:element name="script" type="dialogue"/>
    <xsd:element name="question" type="xsd:multipleChoice" minOccurs="0"/>
    <xsd:element name="task" type="taskType"/>
  </xsd:sequence>
  <xsd:attribute name="creationDate" type="xsd:date"/>
</xsd:complexType>
```

Note that the schema also defines data types and minimum and maximum occurrences for an element. New types can be derived by placing restrictions on existing simple types such as *integer* as demonstrated in the following extract.

```
<!--optNo type is used to restrict the number of valid options to 5. This is done by restricting an existing xml schema type namely positiveInteger-->
```

```
<xsd:simpleType name="optNo">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:maxInclusive value="5"/>
  </xsd:restriction>
</xsd:simpleType>
```

An XML schema uses the file extension *.xsd* and declares the namespace name *http://www.w3.org/1999/XMLSchema* using the prefix *xsd*. The full XML schema developed for the sample data, *scenario.xsd*, can be viewed in Appendix 3.

The elements defined in `scenario.xsd` were then used to describe individual components in the data sample using XML tags. The following extract shows an option from the multiple choice question.

```
<questionOption>
  <optionNumber>1</optionNumber>
  <option>The team don't get along together</option>
  <feedback>The team probably do get along well but they may need help in organising structured
communication that will assist them with their work.
  </feedback>
</questionOption>
```

The complete XML document, *team.xml* can be viewed in Appendix 4.

XML schema allows instance documents to reference several schema using the namespace facility. It is envisaged that the schema for the sample 'scenario.xsd' would be one of many that could be created to cover a range of learning resource fragments and interactions. They could be stored in one location and accessed through a main schema document using a *target namespace* and the XML schema *include* element. This is demonstrated in Figure 4. XML Schema also permits multiple schema components to be imported, from multiple namespaces using the *import* element.

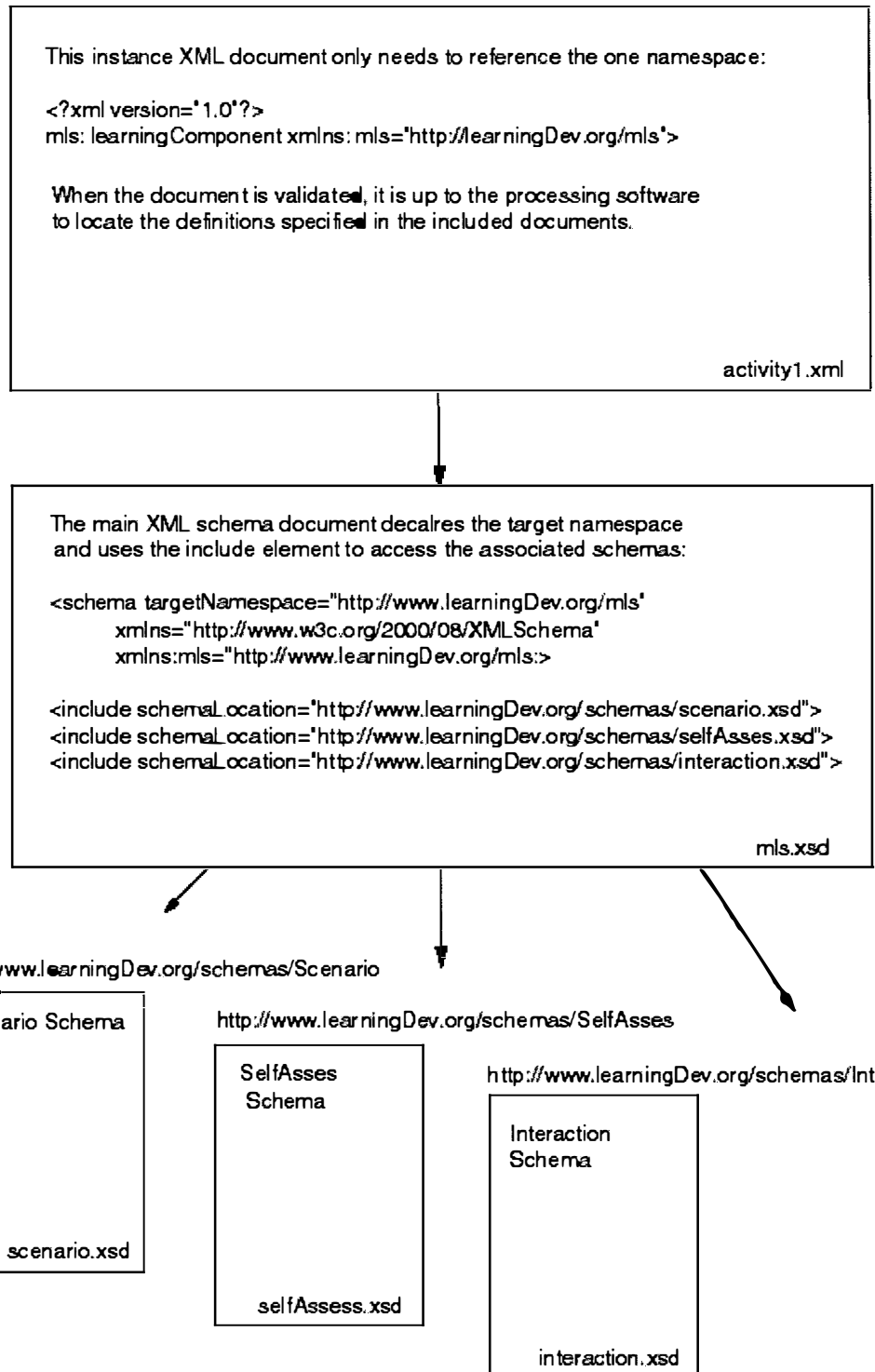


Figure 5: Using multiple XML schemas in a single instance document.

Conceptual information

The metadata applied to the sample thus far provides descriptive and administrative information and identifies the individual components of the resource. This will be sufficient to facilitate test 1 and 2. However, test 3 requires a user to be able to source the learning outcome value from the data sample by searching for it in two different ways:

1. using the language of the elements of competency ‘work cooperatively’, and
2. the abstract term ‘team work’.

To create a relationship between these two options metadata was developed based on principles from the field of ontology.

Semantics

XML is often referred to as *semantic markup*, since it provides a methodology to describe digital data. By adding metadata encoded in XML to the sample data the information it contains is richer and hence easier to locate and reuse. The XML provides a structure of nested elements and this strict structure can be accessed and manipulated by software. However, the metadata descriptors and the strict structure fail to meet all the complex needs of human users and intelligent software applications.

The concise oxford dictionary defines *semantic* as “relating to meaning in language”. As we have seen the meaning drawn from words will be influenced by a number of factors including; the context in which they are presented, the cultural background of the individual user and their prior knowledge.

Erdmann & Studer (1999, p .2) suggest that to facilitate knowledge management it is necessary to “speak about concepts and semantic relationships

rather than element nesting or sequential order”. They argue that to add true semantics to XML documents they should be related to an ontology.

By providing a conceptual view of a knowledge domain Ontologies help to reduce ambiguity and assist with inference mechanisms. An ontology for training packages could for example define that the concept of *team work* is related to the element of competency *working cooperatively*.

This would assist a user in locating the data sample since a software program could use the relationship defined in the ontology to return the data sample in response to a query about team work. The ontology would provide “background knowledge to the query” (Erdmann & Studer, 1999 p. 3). The Simple Logic-based RDF Interpreter (SiLRI), (Decker, 2000) is a software application that demonstrates this approach. Figure 6 below depicts how the search engine is informed by both the resource metadata (in RDF format) and a set of rules based on an ontology.

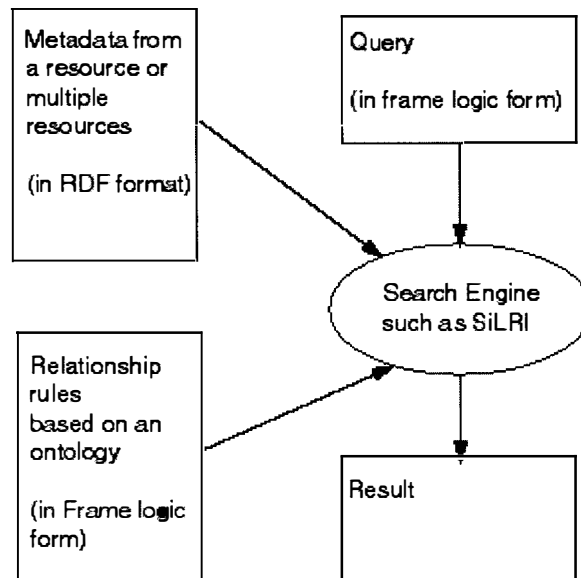


Figure 6: Simple Logic-based Interpreter (SiLRI).

Note that SiLRI uses Frame Logic for the query and rules and the Resource Description Framework (RDF) for the source documents. Frame Logic provides a formalism for representing fundamental concepts that come from object-oriented programming such as class hierarchies and inheritance (Kifer, Lausen, & Wu, 1990, p741). It is also suitable for modeling the semantic information in an ontology, which is also based on a hierarchical class structure. RDF as discussed in the Literature Review was developed to “provide a foundation for metadata interoperability across different resource description communities”(Iannella, 1999, p. 1). It provides a more abstract data model than that of XML DTDs and is also closely related to object-oriented models with class hierarchies. The RDF Schema Specification became a Candidate Recommendation in March, 2000 with the goal of the project and of RDF specifically, being “to produce a language for the exchange of machine-understandable descriptions of resources on the Web” (Brickley, & Guha, 2000).

This machine understandable structure allows more advanced manipulation of metadata by software agents and applications. Since it uses XML as an interchange syntax it is easily read and understood by humans too.

SiLRI was used for Test 3, therefore it was necessary to:

1. create an ontology to represent the learning outcome knowledge for the data sample;
2. create an RDF schema based on the ontology; and
3. create the metadata document as an instance of the RDF schema .

Creating an Ontology

The ontology used is based on an example by Erdmann & Studer (1999, p. 8) in their paper *Ontologies as Conceptual Models for XML Documents*. It defines a hierarchy of the knowledge concepts and the associations between them. It is represented in Frame Logic, thus the term $x::y$ means that x is a sub concept of y . Note that not all concepts and relationships have been defined, only what is necessary to demonstrate the methodology. The ontology consists of three distinct sections. The first section defines the hierarchy of the learning outcome knowledge and is represented as follows.

```
Object[]
LearningOutcome :: Object
  ElementCompetency :: Learning Outcome
    PerformanceCriteria :: ElementCompetency
  Theme :: Learning Outcome
```

The second section defines the relationships between the concepts. Relationships are realised by attributes. So an attribute such as 'relatesToElement' can be used along with attributes with values such as STRING, with their value being another concept. In the following extract the attributes are listed within square brackets.

```
LearningOutcome
[loDescribe=>>STRING; relatesToElement=>>ElementCompetency]

ElementCompetency
[eDescribe=>>STRING; containsPC=>>PerformanceCriteria; relatesTo=>>Theme]

PerformanceCriteria
[pcDescribe=>>STRING isPartOf=>>ElementCompetency]

Theme[tDescribe=>>STRING; isAbout=>>ElementCompetency]
```

The LearningOutcome element has a loDescribe attribute which is of type string and would contain a text description and a relatesToElement attribute which is of type ElementCompetency. The value of this attribute would be an Element of

Competency. ElementCompetency has three attributes, eDescribe which is of type string, containsPC of type PerformanceCriteria and relatesTo of type Theme. PerformanceCriteria has a string attribute pcDescribe and isPartOf, which is of type ElementCompetency. Lastly Theme contains two attributes tDescribe which is of type string and isAbout of type ElementCompetency. The Theme concept is used to identify holistic principles associated with the learning outcome.

The third and final section of the ontology provides rules or axioms. These can be used by an inference engine such as SiLRI to infer new knowledge based on the given facts. This enables incomplete knowledge to be rounded out (Erdmann & Studer, 1999, p. 9).

The following rule states that if the performance criteria pc1 is part of the element of competency e1 then the element of competency e1 contains the performance criteria pc1.

```
FORALL pc1, e1
e1:ElementCompetecny[containsPC->>pc1]<->pc1:isPartOf[eDescribe->>e1]
```

This relationship would allow a user to locate an element of competency even if they only knew the performance criteria and visa versa. This concept was adopted to allow a user to source the learning outcome of the data by either:

- the elements of competency ‘work cooperatively’; or
- the abstract term ‘team work’.

The following rule was created to establish a relationship between the two nomenclatures.

```
FORALL e1, t1
e1:ElementCompetecny[relatesTo->>t1]<->t1:isAbout[eDescribe->>e1]
```

The three sections, hierarchy, relationships and rules completed the ontology. The next step was to apply these identifiers to the data sample using metadata encoded in the Resource Description Framework RDF.

RDF Schema

In Chapter 2, it was mentioned that the RDF schema specifies class hierarchies and constraints. Thus the hierarchical concepts in the ontology can be expressed as classes in the RDF schema as follows:

```
<rdf:Description rdf:ID="LearningOutcome">
  <rdfs:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf
    rdf resource=" http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Description>
```

The first two lines declare *LearningOutcome* as a resource of type *class*. It will therefore inherit all the characteristics that are expected of a class. Everything described by RDF is considered a *resource* and all resources are instances of the class *rdfs:Resource*. Hence *LearningOutcome* is declared a sub class of the class *Resource*. The prefix *rdfs* denotes the resource description framework schema. An abbreviated form of the syntax can be used as in the following class declaration.

```
<rdf:Class ID="PerformanceCriteria">
  <rdfs:subClassOf rdf:resource="#ElementCompetency"/>
</rdf:Class>
```

In this case the class *PerformanceCriteria* is a sub class of the class *ElementCompetency*.

The relationships that were realised by attributes in the ontology are defined as properties in RDF as follows:

```
<rdf:Description rdf:ID="eDescribe">
  <rdfs:type resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description >
```

RDF properties are constrained by *rdfs:domain* and *rdfs:range*. The RDF Schema Specification 1.0 (Brickley & Guha, 2000) states that:

“The value of a property should be a resource of a designated class. This is known as a *range* constraint....a property may be used on resources of a certain class. This is known as a *domain* constraint.”

The property *eDescribe* is declared with the range *literal*. This means that it is a member of the class of Literals (atomic values), that represents string types. The following property declaration for *containsPC* has a range of class *PerformanceCriteria* and a domain of *ElementCompetency*. In other words, the property can be applied to an instance of the *ElementCompetency* class and must be of type *PerformanceCriteria*. Note that domains and ranges are always classes. This declaration also uses the abbreviated form of syntax.

```
</rdf:Property ID="containsPC">  
  <rdfs:domain rdf:resource="#ElementCompetency"/>  
  <rdfs:range rdf:resource="PerformanceCriteria"/>  
</rdf:Property>
```

The full RDF schema can be viewed in Appendix 5.

RDF Metadata

This schema was then used to create metadata for the learning outcomes of the data sample. As discussed in Chapter 2, metadata can be stored within source documents or separately in a metadata repository. For instance the EdNA metadata tool (Currie, 2000, p.1), creates RDF metadata about contributors' resources. It is then stored in a repository that can be searched using their Harvesting Robot.

For this project the RDF metadata was stored in two files. The main file *team.rdf* contained metadata about the data sample. For simplicity only one Element of Competency and one Performance Criteria were used.

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/TR/RDF-syntax#"
  xmlns:lo="http://learningDev.com/">

  <rdf:Description
    rdf:about="http://learningDev.com/courses/MD05/team.xml">

    <lo:relatesToElement rdf:parseType="Resource">
      <lo:eDescribe>1. Working cooperatively with others</lo:eDescribe>
      <lo:containsPC rdf:parseType="Resource">
        <lo:pcDescribe>1.1 team decisions.</lo:pcDescribe>
        <lo:isPartOf>Working cooperatively with others</lo:isPartOf>
      </lo:containsPC>
    </lo:relatesToElement>

  </rdf:Description>
</rdf:RDF>

```

Note the first few lines of code in an RDF document tell the user agent (browser) which schemas will be used. In this case, the rdf schema as defined by the W3C and the learning outcome schema created specifically for this project. A namespace declaration and the prefix *lo* was used to create an unambiguous reference to the learning outcome schema.

The second file *theme.rdf* stored information about the theme. Again for simplicity only one theme was used.

```

... ..
<rdf:Description
  rdf:about="http://learningDev.com/courses/theme.rdf">
  <lo:tDescribe>Holistic theme of learning outcome</lo:tDescribe>
  <lo:relatesTo>group work</lo:relatesTo>
</rdf:Description>
... ..

```

On a larger scale it is anticipated that many resources with a range of learning outcomes could be related to a single theme, as shown in the diagram below.

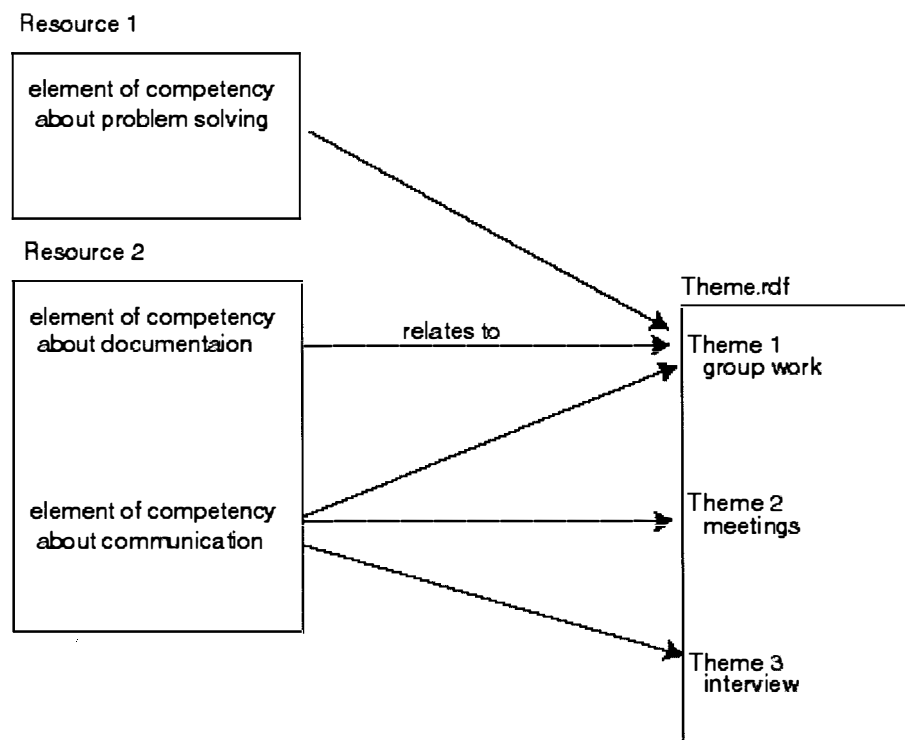


Figure 7: Learning resources in an RDF metadata repository.

Therefore rather than store the *theme* information within each resource, it is stored separately. The relationship between the elements of competencies and themes are set out in a rule file, which will be used by the inference engine.

Summary

At the conclusion of this stage of the implementation, the following documents had been created:

1. *team.xml* – an XML file containing the data sample marked up with metadata based on both the team.xsd schema and the LOM scheme.
2. *team.rdf* – an RDF file containing metadata about the learning outcomes of the data sample (team.xml).

3. *theme.rdf* – an RDF file containing a description of an holistic theme that encapsulates the learning outcome of the data sample.

These files were used in the next stage of the implementation that is covered in Chapter 5.

CHAPTER 5

Tests and results

Introduction

The final stage of the implementation involved carrying out the three tests specified in Chapter 4:

1. *The ability to identify and reuse information components* – this was tested by singling out individual sections of the data sample and reusing them in another context.
2. *The ability to easily customise the information* – this was tested by attempting to remould sections of the sample so that it was applicable to a different industry setting.
3. *The ability to extract conceptual information* – This was tested by using an inference engine to return the element of competency ‘Working cooperatively with others’ from a search of the term ‘team work’.

Test 1

Having created an XML version of the data sample, a software solution was required to manipulate it. As discussed in Chapter 2 the Document Object Model (DOM) and the Simple API for XML are the two most common approaches to processing XML. Both of these require custom applications and both options were explored.

A Java program was written using SAX 1.0 methods to extract the 'role' components of team.xml, which introduce each team member. The *startElement* method was used to catch the start of each element. If the element = 'role' the contents were stored in a variable and printed. The source code for the Java program can be viewed in Appendix 6 together with the output created when the program was executed using the Java Development Kit (JDK).

This approach was successful in that the 'role' components of the data sample could be located and extracted, however it would require a user to have programming skills and access to the JDK. Also it only provided a part solution since having extracted the components further work would be required to adapt them for reuse.

The DOM approach required an HTML document with JavaScript. A DOM model of the xml document was created using the following code:

```
objDOM.load("team.xml");
```

Individual components of team.xml could then be extracted using DOM methods. For example the *selectSingleNode* method was used to extract the *introduction* element and store it in a variable called *introElement*:

```
var introElement = objDOM.selectSingleNode ("/teamScenario/introduction");
```

Where more than one instance of an element existed the *getElementByTagName* method was used:

```
var objNodeList = objDOM.getElementByTagName ("role")...
```

The HTML document can be viewed in Internet Explorer (IE) 5, which uses the MSXML parser. It contains a Form with 5 radio buttons. By selecting a radio button the user can extract the individual components of team.xml. This approach was also successful in that individual components of team.xml could be located and

extracted. The DOM also has methods that would allow a user to amend, delete and add to the original document structure. This approach was more user friendly but still required some programming skills. This technology is constantly evolving and being updated and problems with cross browser compatibility were encountered. The source code for this implementation can be viewed in Appendix 7 along with a screen shot of the HTML document.

Since neither the DOM or the SAX fully met the requirements of the test it was decided that XSLT may be a better approach. XSLT is a declarative language and is used to search and manipulate XML documents in the same way that the Structured Query Language (SQL) is used to manipulate data in relational database systems. Ladd & O'Donnell et al (2000, p350) suggest that "Just as SQL is, in a sense, the powerhouse of a relational database management system, so XSLT is the powerhouse of the XML family of technologies." This approach required an XSLT style sheet that could be used to transform the data in team.xml into HTML format by an XSLT engine. The following is an extract from the XSLT style sheet that was created. It shows how the statement from the multiple choice question is extracted and placed in an HTML table row:

```
<td>
<P><xsl:value-of select="/teamScenario/question/multipleChoice/Statement"/></P>
</td>
```

The full XSLT style sheet can be viewed in Appendix 8. It can be seen that XSLT works with a tree model of the data similar to the DOM, however since XSLT is a declarative programming language it is easier to learn and use. This approach was the most suitable for the test since the same style sheet could be used to produce any number of variations of team.xml. Furthermore team.xml could be easily adapted using Microsoft's XML Notepad.

The objective of test 1 was to reuse a component of the sample data. To demonstrate this the multiple-choice question component was selected. Using Microsoft's XML Notepad, all other unwanted components were removed from the team.xml document. New information, relevant to another training product, was entered in the value fields of the question elements. The new XML document was saved as reuse.xml.

James Clark's XSLT engine, XT was used to do the transformation. XT was run from the DOS command prompt with the following input files:

- reuse.xml (*source document*) team.xsl (*style sheet*) reuse.htm (*result document*)

The following screen captures shows the reuse.htm result document generated by XT.

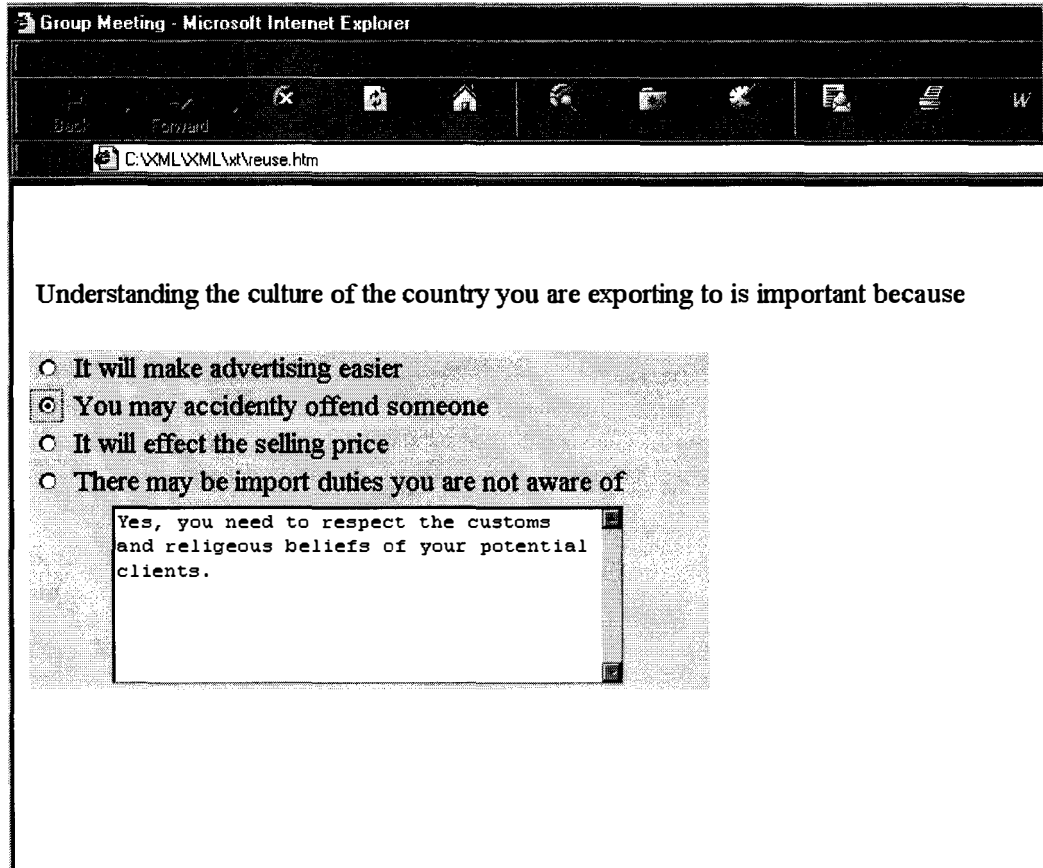


Figure 8: Output from test 1

Test 2

The objective of test 2 was to allow easy customisation of the data sample. In Chapter 4 it was suggested that Registered Training Organisations (RTOs) that purchase an online training product may want to customise it to meet their particular industry sector or user needs. The example given was that in the data sample the multimedia team are developing instructional material that would be completely irrelevant to a production house that specialises in e-commerce. They would however have the same interest in team dynamics, communication and report writing. In test 2 the values of the introduction, the team characters and their dialogue were changed to reflect the e-commerce industry sector. Once again XML notepad was used to make the changes and the new document was saved as customisedTeam.xml.

XT was run with customisedTeam.xml and team.xsl as input and customisedTeam.htm specified as the result file. The following screen captures shows an extract from the customisedTeam.htm document generated by XT.

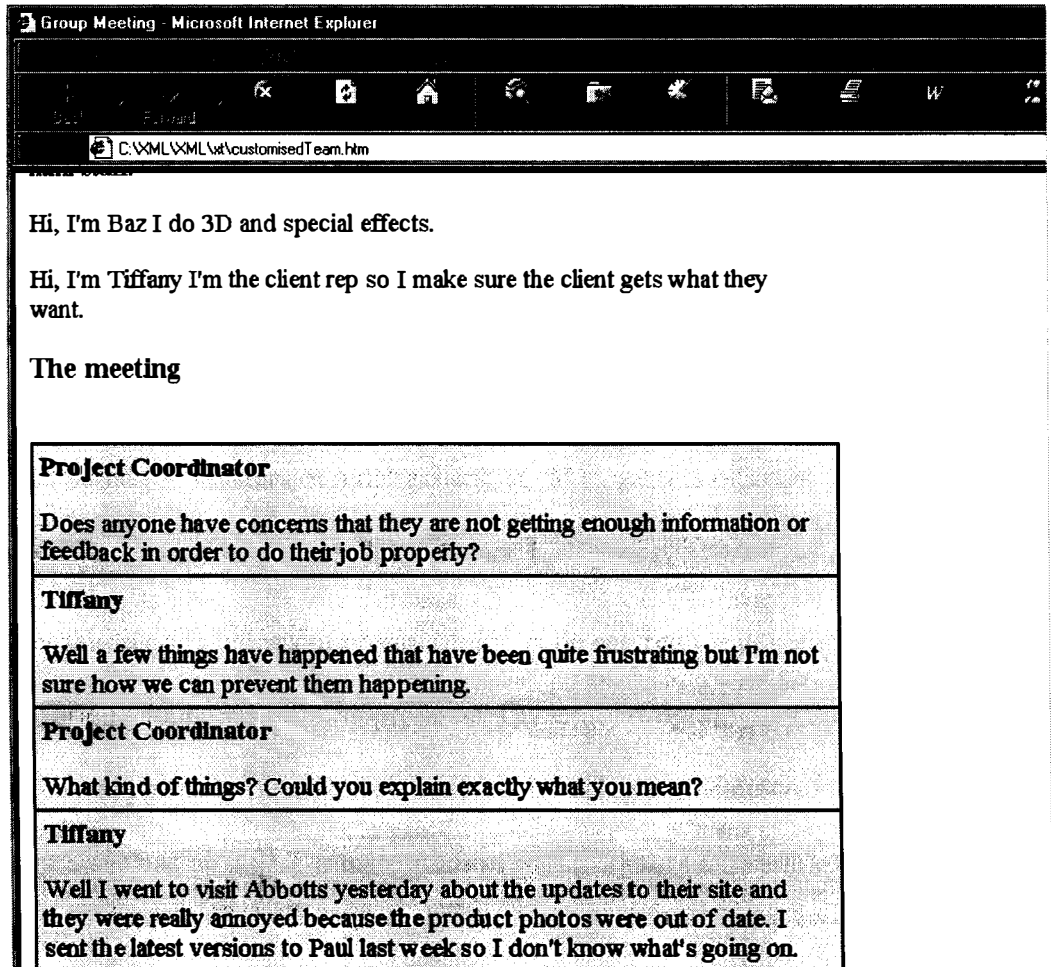


Figure 9: Output form Test 2.

The structure and the concept are the same as the original exercise, however the text has been changed to reflect the development of commercial web sites as opposed to learning material.

By using an XSLT style sheet and the XT transformation engine the objectives of both test 1 and 2 had been met. It is envisaged that all of the functionality of the XT engine will eventually be available through a user agent, further simplifying the process for end users. However XSLT is at the time of writing still a very new technology. Ladd, Odonnell et al (2000, p351) predict that:

“Increasingly, you can expect to see XSLT processors disappear, not in the sense that they won’t be there but they will likely be used from within some sort of XML Integrated Development Environment.”

Although it may be some time before the technology matures, these tests have shown that metadata encoded in XML will allow easy access to and customisability of learning knowledge components. Hence promoting Beard’s vision of “shared resources created by several instructors at several universities located around the world” and Schatz’s “clusters of independent, stand alone bits of knowledge that can be adapted and reused” as discussed in Chapter 1.

Reusability of sections of programming code can reduce development time and costs for developers of training products. Currently programmers could use a copy and paste approach to reuse such code but by using metadata encoded in XML and software such as an XSLT transformation engine a more extensible approach can be achieved.

Test 3

Test 3 involved the Simple Logic-based RDF Interpreter (SiRLI) software. This program is written in Java and works in conjunction with the Simple RDF Parser (SiRPAC). SiRPAC is a set of Java classes that can parse RDF/XML documents into the 3-tuples of the corresponding RDF data model. It uses an XML parser and the SAX API. To run test 3 it was therefore necessary to have the following software installed:

- the Java Development Kit (JDK)
- an XML parser – JamesClark’s XP parser was used
- SiRPAC, and
- SiRLI.

Before proceeding with SiRLI a test run of SiRPAC was necessary. SiRPAC was run from the command line with team.rdf as input and the following triples were created:

```
C:\code>java -Dorg.xml.sax.parser=com.jclark.xml.sax.Driver org.w3c.rdf.SiRPAC team.rdf
triple("http://learningDev.com/eDescribe","file:team.rdf#genid2","1. Working co-operatively with
others").
triple("http://learningDev.com/pcDescribe","file:team.rdf#genid3","1.1 team decisions.").
triple("http://learningDev.com/isPartOf","file:team.rdf#genid3","Working cooperatively with
others").
triple("http://learningDev.com/containsPC","file:team.rdf#genid2","file:team.rdf#genid3").
triple("http://learningDev.com/relatesToElement","http://learningDev.com/courses/MD05/team.xml
","file:team.rdf#genid2").
C:\code>
```

Since SiRLI is an inference engine it requires a rule file to define rules or axioms about the data and a query file. The objective of this test was to be able to search for and locate the data sample by the element of competency ‘Working

cooperatively with others’ from a search by the more abstract term ‘group work’. In order to test this with SiRLI a query was set using the theme (which contains the value *group work*) as follows:

```
FORALL learning_outcome <- theme(learning_outcome).
```

The rule was created, based on the ontology, to establish a relationship between the theme and the performance criteria *Work cooperatively with others*.

```
FORALL e1,t1 theme(t1) <- e1["http://learningDev.com/relatesTo"->>t1] <->
eDescribe(e1) <- e1["http://learningDev.com/eDescribe"->>t1].
```

Note that the elements are identified using their full URI (the prefix of *lo* seen in the rdf document is expressed in full (“http://learningDev.com”) followed by the element name). The rule file with this one rule was saved as *r.rule* and the query was saved as *q.query*.

SiRLI was run from the command line prompt with *team.rdf*, *theme.rdf*, *r.rule* and *q.query* as input files. The following result was obtained.

```
C:\code>java -Dorg.xml.sax.parser=com.jclark.xml.sax.Driver edu.unika.aifb.rdfie
.SiRLI r.rule -rdf team.rdf -rdf theme.rdf query.q

learning_outcome = literal("1. Working cooperatively with others")
learning_outcome = literal("group work")

C:\code>
```

Although the query only asked for the ‘theme’ element the search also returned the element of competency. This was as a result of the rule provided. The rule and query files could be enhanced to also retrieve the name of the resource, or to search by specific values contained within the elements. However this result was enough to verify the underlying principle of how inference mechanisms can be used to extract multiple views of meaning from digital resources.

No further tests were run to verify the effectiveness or otherwise of the software itself since the authors had demonstrated this in their article *A Query and Inference Service for RDF*.

As mentioned in Chapter 2 Tools such as *OntoEdit - Ontology Engineering Environment* under development by the Knowledge Management Group at the University of Karlsruhe, Institute AIFB will protect developers from the complexities of generating the frame logic axioms and queries, making this technology more appealing. Improved user agent support will also hide complexity from end users.

Summary

The RDF Model and Syntax Specification points out that “In a medium of global scale such as the World Wide Web it is not sufficient to rely on shared cultural understanding of concepts such as ‘creatorship’; it pays to be as precise as possible”(Lassila 1999). Ontologies and RDF Schema together provide a physical and logical model to add precision to metadata. The simple inference example shown here is only a small indication of the improved precision in resource discovery that could be obtained through machine understandable metadata.

CHAPTER 6

Conclusion

Applying metadata in XML format to the data sample facilitated the extraction and adaptation of the individual knowledge components it contained. By representing the learning outcomes of the sample in RDF format an inference mechanism could be used to source the sample from two forms of nomenclature. The tests carried out on this small sample of information indicate that metadata has more to offer in the digital world than in the print based medium. It can be applied to various levels of granularity and can provide multiple views of a single resource, thus providing the semantics necessary to convert raw data into meaningful chunks of knowledge.

Many organisations are understandably deterred from applying metadata. Its application is time consuming, costly and often confusing. Which vocabulary to choose? Which technology to use? Which standards to follow? Constant changes and updates in these areas also add to the dilemma. This project has attempted to demonstrate that the benefits can out-weigh the difficulties and that with careful planning metadata can improve productivity and leverage the knowledge in web based resources.

While experimenting with the Document Object Model (DOM) and Simple Interface for XML (SAX), it became obvious that being able to manipulate data at the smallest level of granularity opens up a whole range of opportunities for

developers and designers. Difficulties encountered during the project, such as the failure to produce results in Netscape 6 and the complexity of using DOS based software where browser support was not yet available, suggest that the true potential of the XML suite of technologies is yet to be harvested. However the technology is fundamental to the future direction of the Web and continued support for it by standards organisations and developers is guaranteed. For example on the 19th December 2000, the World Wide Web Consortium (W3C) released Extensible HTML (XHTML) Basic as a W3C Recommendation. XHTML Basic “is designed for Web clients that do not support the full set of XHTML features; for example, Web clients such as mobile phones, PDAs, pagers, and set top boxes” (XHTML Basic, 2000).

As this project demonstrated the Resource Description Framework (RDF) specification provides a structure for machine understandable metadata. It is expected that RDF will become a processing rules language for automated decision making about Web resources (Swik, 1999). This will pave the way for sophisticated data mining technologies developed in areas such as Knowledge Discovery in Databases (KDD) to be applied to Web Resources resulting in more precise knowledge retrieval and manipulation.

The XML family of technologies can provide the physical means for knowledge management in Berner-Lee’s semantic web. However, on a logical level, as this project has demonstrated the creation and use of meaningful metadata that addresses the multivalent nature of meaning and adheres to standards and schemas is required for optimal knowledge provisioning.

REFERENCES

- Altman, R.B., Badas, M., Xiaoqian, J.C., Carillo, M.W., Chen, R.O., & Abernethy, N.F. (1999). RiboWeb: An Ontology-based System for Collaborative Molecular Biology IEEE Intelligent Systems,14(5), 68-76.
- ARIADNE (2000). Alliance of Remote Instructional Authoring and Distribution Networks for Europe (on-line) Available WWW: <http://ariadne.unil.ch/Metadata/> (2000, June, 06).
- Beard, R. (2000). Faculty Initiative in Expanding Classroom Web usage. In Technology Source. (on-line) Available WWW: <http://horizon.unc.edu/TS/development/2000=05.asp> (2000, May, 05).
- Baxter, J. & Chua, W.I. (1999). Now and the Future. Australian Accounting Review, 9(3), 3-14.
- Berners-Lee, T. (1998). What the Semantic Web can represent (on-line) Available WWW: <http://www.w3.org/DesignIssues/RDFnot.html> (2000, March,11).
- Boehm, C. (1999). The Metadata Bear. Or: Bearing the weight of accessibility Journal of Educational Media,24(3), 177-189.
- Boye J. (1998) RDF – What’s in it for us? In International Related Technology (on-line). Available WWW: <http://www.irt.org/articles/js086/index.htm> (2000, June, 11).
- Brickley, D., & Guha, R. V. (2000). Resource Description Framework (RDF) Schema Specification 1.0. (on-line). Available WWW: <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>. (2000, July, 01).
- Chandrasekaran, B., Josephson, R., & Benjaminis, V.R. (1999). What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems,14(1), 20-25.
- Chen, W., & Mizoguchi, R. (1999). Communication Content Ontology for Learner Model Agent in Multi-agent Architecture. [on-line] Available WWW: <http://www/ei.sanken.osaka-u.ac.jp/aied99/a-papers/w-chen.pdf>. (2000, March, 12).
- Clark, J. (1998). XT (on-line). Available WWW: <http://www.jclark.com/xml/xt.html> (2000, May, 05).
- Colombo, C., Del Bimbo, A., & Pala, P. (1999). Semantics in Visual Information Retrieval IEEE Multimedia,6(3), 38-53.

- Currie, M. F. (2000). The EdNA Higher Education project (on-line) Available WWW: <http://www.dis.unimelb.edu.au/staff/mikec/edna/projects.html> (2000, November, 30)
- Decker, S. (2000). SiLRI (Simple Logic-based RDF Interpreter). (on-line). Available WWW: <http://www.aifb.uni-karlsruhe.de/~sde/rdf/> (2000, April, 17).
- Decker, S., Erdmann, M., Fensel, D., & Studer, R. (1999). Ontobroker: Ontology based Access to Distributed and Semi-Structured Information In R. Meersman et al (eds), Proceedings of DS-8 Semantic Issues in Multimedia Systems (pp. 351-369). Boston: Kluwer Academic Publishers.
- Detmer, W.M. & Shortliffe, E.H. (1997). Using the Internet to Improve Knowledge Diffusion in Medicine, Communications of the ACM,40(8), 101-108.
- DC (1998). Dublin Core Metadata Initiative (on-line). Available WWW: <http://www.purl.oclc.org/dc> (2000, March, 14).
- EdNA (2001). Education Network Australia (on-line). Available WWW: <http://standards.edna.edu.au/metadata/> (2000, October, 18).
- Erdmann, M., & Studer, R. (1999). Ontologies as Conceptual Models for XML Documents (on-line). Available WWW: <http://www.aifb.uni-karlsruhe.de/WBS/broker>, (2000, April, 15).
- DSTC (2000). Distributed Systems Technology Centre. (on-line). Available WWW: <http://www.dstc.edu.au/index.html> (2000, June, 07).
- Fallside, D.C. (2000). XML Schema Part 0: Primer W3C Candidate Recommendation 24th Oct 2000. (on-line). Available WWW: <http://www.w3.org/TR/xmlschema-0/>. (2000, November, 04).
- Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H., Staab, S., Studer, R., & Witt, A. (1999). On2broker: Semantic-Based Access to Information Sources at the WWW. Proceedings of the World Conference on the WWW and Internet (WebNet 99), Honolulu, USA.
- Flores-Mendez, R.A. (1999). Towards a Standardization of Multi-Agent System Frameworks. (on-line). Available WWW: <http://www.acm.org/crossroads/xrds5-4/multiagent.html> (2000, August, 22).
- Gilliland-Swetland, A.J. (2000). Introduction to Metadata: Setting the Stage (on-line). Available WWW: <http://www.getty.edu/gri/standard/intrometadata> (2000, October, 05).
- Gecsei, J. (1997). Adaption in Distributed Multimedia Systems. IEEE Multimedia,4(4), 58 –65.
- Goldberg, M. (2000). Publisher Beginnings. In Online Teaching and Learning Newsletter (on-line). Available WWW:

<http://www.webct.com/service/ViewContent?contentID=2339455&communityID=1464907> (2001, January, 20).

- Goldfarb, C.F., & Prescod, P. (2000). The XML Handbook. USA: Prentice Hall
- Gray M. (2000). Discussion Paper: Web technologies report to the Steering Committee of the VET Preferred Standards project describing the findings of DSTC Pty Ltd regarding web technologies for VET online training. Draft 0.6. This discussion will inform the VET Preferred Standards Document. (on-line). Available WWW: http://vetpst.dstc.edu.au/3WebProtocols/docs/_Toc496943796. (2000, December, 16).
- Gruber, T. (2000). What is an Ontology (on-line). Available WWW: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> (2000, November, 27).
- Guarino, N. (1997). Understanding, Building and Using Ontologies (on-line). Available WWW: <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/guarino/guarino.html> (2000, April, 14).
- Guarino, N., Masolo, C., & Vetere, G. (1999). OntoSeek: Content-Based Access to the Web IEEE Intelligent Systems,14(3), 70 –76.
- Heflin, J., Hendler, J., & Luke, S. (1999). SHOE: A Knowledge Representation Language for Internet Applications Technical Report CS-TR-4078 Department of Computer Science, University of Maryland at College Park.(on-line). Available WWW: <http://www.cs.umd.edu/projects/plus/SHOE/pubs/#aiweb00> (2000, April, 05).
- Hendler, J., & Stoffel, K. (1999). Back-End technology for High-performance Knowledge Representation Systems IEEE Intelligent Systems 14,(3), 63-69.
- Hunter, D. (2000). Beginning XML. Birmingham UK: Wrox Press.
- Iannella, R. (1999). An Idiot's Guide to the Resource Description Framework. (on-line). Available WWW: <http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/> (2000, June, 22).
- IEEE (2000). Institute of Electrical and Electronics Engineers (IEEE). (on-line). Available WWW: <http://www.ieee.org/> (2000, August, 22).
- IMS (2000). Global Learning Consortium (on-line). Available WWW: <http://www.imsproject.org/metadata/index.html> (2000, September, 03).
- Karvounarakis, G., Christophides, V., & Plexousakis, D. (2000). Querying Semistructured (Meta)Data and Schemas on the Web: The case of RDF & RDFS. (on-line). Available WWW: <http://citeseer.nj.nec.com/322145.html> (2000, October, 09).

- Kifer, M., Lausen, G., & Wu, J. (1995) Logical Foundations of Object-Oriented and Frame-Based Languages. Communications of the ACM,42(4), 741-843.
- Kushmerick, N. (1999). Gleaning the Web IEEE Intelligent Systems 14,(2), 20-22.
- Ladd, E., O'Donnell, J., Morgan, M., & Watt, A.H. (2000). Using XHTML, XML, and Java 2. USA: Que Corporation.
- Lassila, O. (1999). Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation 22 February 1999. (on-line). Available WWW: <http://www.w3.org/TR/REC-rdf-syntax/> (2000, June, 02).
- LTSC. (2000) Learning Technology Standards Committee. (on-line). Available WWW: <http://ltsc.ieee.org> (2000, October, 19).
- Le Hégarret, P. (2000). Document Object Model (DOM). (on-line). Available WWW: <http://www.w3.org/DOM/> (2000, November, 10).
- LOM (2000). Learning Object Metadata Working Group (on-line). Available WWW: <http://ltsc.ieee.org/wg12/> (2000, September, 06).
- Lopez, M.F., Gomez-Perez, A. & Sierra, J.P. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. IEEE Intelligent Systems,14(1), 37-45.
- Miller, J., Tsatalos, O.G., & Williams, J.H. (1997). DataWeb: Customizable Database Publishing for the Web IEEE Intelligent Systems,4(4), 14-21.
- MSXML 3.0 (2000). MSXML Parser 3.0 Release. (on-line). Available WWW: <http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/msdn-files/027/000/541/msdncompositedoc.xml> (2000, October, 17).
- O'Leary, D.E. (1998). Using AI in Knowledge Management: Knowledge Bases and Ontologies IEEE Intelligent Systems,13(3), 34-39.
- OntoEdit (2000). Ontology Engineering Environment. (on-line). Available WWW: <http://ontoserver.aifb.uni-karlsruhe.de/ontoedit/> (2000, August, 31).
- Ostyn, C. (1998). IEEE P1484.20 Competency Definitions Working Group. (on-line). Available WWW: <http://ltsc.ieee.org/wg20/index.html> (2000, November, 14).
- Pedersen, J. (1999) Intelligent Information Retrieval IEEE Intelligent Systems,14(4), 30-31.
- Peereboom, M. (1998). DONOR WP1 Granularity and the use of DC.Relation, (on-line). Available WWW: <http://www.kb.nl/coop/donor/project-en-index.html?coop/donor/rapporten/granularity.html> (2000, December, 02).
- Saarela, J. (1999). SiRPAC - Simple RDF Parser & Compiler. (on-line). Available WWW:<http://www.w3.org/RDF/Implementations/SiRPAC/> (2000, July, 16).

- SAX 2.0. (2000). SAX 2.0: The Simple API for XML. (on-line). Available WWW: <http://www.megginson.com/SAX/> (2000, July, 16).
- Schatz, S. (2000). Paradigm Shifts & Challenges for Instructional Designers. (on-line) Available WWW: <http://www.imsproject.com/feature/kb/knowledgebits.html> (2000, December, 03).
- Schoening, J. (2000). IEEE Standard Upper Ontology (SUO) Working Group. (on-line). Available WWW:<http://suo.ieee.org/> (2000, December, 07).
- Ram, A., Catrambone, R., Guzdial, M.J., Kehoe, C.M., McCrickard, D.S., & Stasko, T. (1999). PML: Adding Flexibility to Multimedia Presentations IEEE Multimedia, 6(2), 40-51.
- Skonnard, A. (2000). The XML Files. (on-line). Available WWW: <http://msdn.microsoft.com/msdnmag/issues/0900/XML/print.asp> (2000, October, 17).
- Swartout, W. (1999). Ontologies IEEE Intelligent Systems,14(1), 18-19.
- St. Laurent, S., & Biggar, R.J. (1999). Inside XML DTDs: Scientific and Technical. USA: McGraw-Hill.
- Staab, S., Erdmann, M., Maedche, A., & Decker, S. (2000). An Extensible Approach for Modeling Ontologies in RDF(S). Proceedings of the ECDL-2000 Workshop Semantic Web: Models, Architectures and Management (pp. 1-16). Lisbon.
- Suzuki, D. (1998). Earth Time. Australia: Allen & Unwin.
- Swick, R. R. (1999). Frequently Asked Questions about RDF. (on-line). Available WWW: <http://www.w3.org/RDF/FAQ> (2000, December, 16).
- Talmacs, K. (2000). Metadata in the University Environment Paper presented at the VALA 2000 Conference, Melbourne, 18/2/00.
- Tsichritzis, D. (1999). Reengineering the University Communications of the ACM,42 (6). 93-100.
- Uschold, M. (1998). Where are the Killer Apps? Paper presented at 13th European Conference on Artificial Intelligence. (on-line). Available WWW: <http://www.lia.dia.fi.upm.es/WORKSHOP/ECAI98/accepted-papers.html> (2000, August, 02).
- Ward, N. (2000). Information Ecology. (on-line). Available WWW: <http://www.dstc.edu.au/Research/infoecology-ov.html#reports> (2001, January, 11).
- Waterson, A. & Preece, A. (1999). Verifying ontological commitment in knowledge-based systems Knowledge-Based Systems 12, (1-2), 45-54.

- WebCT (2000). Get WebCT-ready publisher content here and spend more time teaching and less time preparing your online course with e-Packs. (on-line). Available WWW: <http://www.webct.com/content/> (2000, December, 20).
- Wise, J.A. (1999). The Ecological Approach to Text Visualisation. Journal of the American Society for Information Science, 50 (13), 1224 - 1233.
- XML 1.0 (2000). Extensible Markup Language (XML) 1.0 (Second Edition). (on-line). Available WWW: <http://www.w3.org/TR/2000/REC-xml-20001006> (2000, November, 02).
- XML Notepad (2000). Microsoft XML Notepad (on-line). Available WWW: <http://msdn.microsoft.com/xml/notepad/intro.asp> (2000, October, 10).
- XMLwriter (2000) Welcome to XMLwriter. (on-line). Available WWW: <http://www.xmlwriter.com> (2000, December, 12).
- XHTML (2000). XHTML1.0: The Extensible HyperText Markup Language. (on-line). Available WWW: <http://www.w3.org/TR/xhtml1/> (2000, December, 09).
- XHTML Basic (2000) XHTML™ Basic W3C Recommendation 19 December 2000. (on-line). Available WWW: <http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/> (2001, January, 07).
- Yang, J., & Papazoglou, M.P. (2000). Interoperation Support for Electronic Business Communications of the ACM, 43 (6).39-47.

APPENDIX 1

Data Sample

The following is problem based scenario from a typical online course. It is shown as it appears in the browser followed by the HTML version.

Team Dynamics

As the Project Manager of a multimedia development team you are concerned that the current project is well behind schedule. You can't understand why this should be as you have plenty of resources and all the necessary equipment. You're wondering if there might be a communications problem. You've just arrived back from a conference and some comments that Steve Bamsworth mentioned in his paper on Team Dynamics has got you thinking. You have called a team meeting to try and determine the level of communication between team members.

Meet the team

Hi I'm Jasmine. I'm a graphic designer and I like to draw characters and create interesting interface designs.

Hi I'm Paul and I do a bit of everything but mostly programming stuff using Director and Flash.

I'm Phil and JavaScript is my specialty but my main job is doing all the html stuff.

I'm Tiffany and I'm the instructional designer. I work from my own office and use email, fax and phone calls to communicate with rest of the team. Once a month I come in to check on how things are going.

Hi, I'm Baz I do graphics and some Flash stuff. I'm best at technical type drawing, doing perspectives and stuff like that.

I'm Tiffany and I'm the instructional designer. I work from my own office and use email, fax and phone calls to communicate with rest of the team. Once a month I come in to check on how things are going.

The meeting

Project Manager

Does anyone have concerns that they are not getting enough information or feedback in order to do their job properly?

Tiffany

Well a few things have happened that have been quite frustrating but I'm not sure how we can prevent them happening.

Project Manager

What kind of things? Could you explain exactly what you mean?

Tiffany

Well one time I sent over some photos I had taken for unit 2. I emailed them to Paul and he put them on your shared drive. The following week I get this message from Baz saying could I check the graphics he had drawn for unit 2. It turned out that he had spent 3 days creating graphics for the stuff I had sent photos for.

Baz

How was I supposed to know there were photos I just read the storyboard and it said picture of girl in wheelchair or whatever and so I drew them. Nobody said anything to me about photos.

Project Manager

So is this the main reason why there are delays with the graphics?

Baz

Well not really, the main problem was having to re-do all the graphics in unit 2 because Tiffany wasn't happy with them.

Tiffany

You make it sound like I'm being fussy Baz but the whole unit is about body language and facial expressions and there just wasn't enough detail in the graphics to show the points I was trying to get across.

Baz

Well that's not my area. I hate drawing faces.

Project Manager

Would it have been better if you had done those Jasmine?

Jasmine

Yeah I guess so but I had a lot to do with the graphics for that big animation in unit 3.

What seems to be the main problem here?

- The team don't get along together**
- Lack of communication**
- A lack of formal procedures**

The team probably do get along well but they may need help in organising structured communication that will assist them with their work.

Task

There seems to be a lot of room for improvement here. Now that the meeting is finished you can start work on your report. Think about what the developers have said and refer to the notes you took during the meeting. You might like to have a look at Barnsworth's paper for some ideas. You can also refer to the Project Management guide or ask advice from one of your colleagues. The format of your report should follow the instructions set out in the Office Policy and Procedures document. You should include:

- List of problems identified.
- List of strategies to address these problems.
- A suggested implementation plan.

HTML version of the data sample.

```
<html>
<head>
<title>teamScenario</title>

<SCRIPT LANGUAGE="JavaScript1.2">

var fbk = new Array();

fbk[1] = "The team probably do get along well but they may need help in organising structured
communication that will assist them with their work.";

fbk[2] = "Yes, it seems that they could work more efficiently if they communicated more with
respect to their roles and responsibilities";

fbk[3] = "There may be a problem in this area but some of the issues raised indicate a need for
communication on a less formal level too.";

</SCRIPT>

</head>
<body>
<table width="550">

<h3>Team Dynamics</h3>

<p>As the Project Manager of a multimedia development team you are concerned that the
current project is well behind schedule. You can't understand why this should be as you have
plenty of resources and all the necessary equipment. Your wondering if there might be a
communications problem. You've just arrived back from a conference and some comments that
Steve Barnsworth mentioned in his paper on Team Dynamics has got you thinking. You have
called a team meeting to try and determine the level of communication between team
members.</p>

<h3>Meet the team</h3>

<p>Hi I'm Jasmine. I'm a graphic designer and I like to draw characters and create interesting
interface designs.</p>

<p>Hi I'm Paul and I do a bit of everything but mostly programming stuff using Director and
Flash.</p>

<p>I'm Phil and JavaScript is my specialty but my main job is doing all the html stuff.</p>

<p>I'm Tiffany and I'm the instructional designer. I work form my own office and use email, fax
and phone calls to communicate with rest of the team. Once a month I come in to check on how
things are going.</p>
<p>Hi, I'm Baz I do graphics and some Flash stuff. I'm best at technical type drawing, doing
perspectives and stuff like that.</p>

<h3>The meeting</h3>

<table WIDTH="60%" BORDER="1" CELSPACING="0" CELLPADDING="5"
BORDERCOLOR="#000066">
  <tr>
    <td BGCOLOR="#BFDFFF">
      <p><b>Project Manager</b></p>
      <p>Does anyone have concerns that they are not getting enough information or feedback in
order to do their job properly? </p>
    </td>
  </tr>
</table>
```

```
|  |
| --- |
| Tiffany  Well a few things have happened that have been quite frustrating but I'm not sure how we can prevent them happening. |
| Project Manager  What kind of things? Could you explain exactly what you mean? |
| Tiffany  Well one time I sent over some photos I had taken for unit 2. I emailed them to Paul and he put them on your shared drive. The following week I get this message from Baz saying could I check the graphics he had drawn for unit 2. It turned out that he had spent 3 days creating graphics for the stuff I had sent photos for. |
| Baz  How was I supposed to know there were photos I just read the storyboard and it said picture of girl in wheelchair or whatever and so I drew them. Nobody said anything to me about photos. |
| Project Manager  So is this the main reason why there are delays with the graphics? |
| Baz  Well not really, the main problem was having to re-do all the graphics in unit 2 because Tiffany wasn't happy with them. |
| Tiffany  You make it sound like I'm being fussy Baz but the whole unit is about body language and facial expressions and there just wasn't enough detail in the graphics to show the points I was trying to get across. |
| Baz  Well that's not my area. I hate drawing faces. |
|  |

```

```

        <p><b>Project Manager</b></p>
        <p>Would it have been better if you had done those Jasmine?</p>
    </td>
</tr>

<tr>
    <td BGCOLOR="#BFDFFF">
        <p><b>Jasmine</b></p>
        <p>Yeah I guess so but I had a lot to do with the graphics for that big animation in unit
3.</p>
    </td>
</tr>

</table>

<form name="f1">
<table WIDTH="100%" BORDER="0" CELSPACING="0" CELLPADDING="5">

    <tr VALIGN="TOP">
        <td>
            <p>What seems to be the main problem here?</p>
        </td>
    </tr>
</table>

<table WIDTH="450" BORDER="0" CELSPACING="0" CELLPADDING="2"
BGCOLOR="#BFDFFF">

    <tr VALIGN="TOP">
        <td><INPUT TYPE="radio" NAME="r1" VALUE="1" onClick="f1.ta1.value=fbk[1]"></td><td>
            <p>The team don't get along together</p>
        </td>
    </tr>

    <tr VALIGN="TOP">
        <td><INPUT TYPE="radio" NAME="r1" VALUE="2" onClick="f1.ta1.value=fbk[2]"></td><td>
            <p>Lack of communication </p>
        </td>
    </tr>

    <tr VALIGN="TOP">
        <td><INPUT TYPE="radio" NAME="r1" VALUE="3" onClick="f1.ta1.value=fbk[3]"></td><td>
            <p>A lack of formal procedures</p>
        </td>
    </tr>

    <tr VALIGN="TOP" ALIGN="CENTER">
        <td COLSPAN="2"><TEXTAREA NAME="ta1" COLS="40" ROWS="7"
WRAP="PHYSICAL"></TEXTAREA>
        </td>
    </tr>

</table>

</form>

<h3>Task</h3>

<table WIDTH="60%" BORDER="0" CELSPACING="0" CELLPADDING="5">
    <tr VALIGN="TOP">
        <td><p>There seems to be a lot of room for improvement here. Now that the meeting is finished
you can start work on your report. Think about what the developers have said and refer to the
notes you took during the meeting. You might like to have a look at Barnsworth's paper for some
ideas. You can also refer to the Project Management guide or ask advice from one of your
colleagues. The format of your report should follow the instructions set out in the Office Policy and
Procedures document. You should include:</p>

```

```
<ul>
  <p><li>List of problems identified.</li></p>
  <p><li>List of strategies to address these problems.</li></p>
  <p><li>A suggested implementation plan.</li></p>
</ul>
</td>
</tr>
</table>
</table>
</body>
</html>
```

APPENDIX 2

Metadata at the highest level of granularity

This metadata is in XML format and based on the Dublin Core (DC) and the Learning Object Metadata (LOM) scheme Draft Document v3.6.

```
<dataSample
  xmlns:dc="http://purl.org/metadata/dublin_core#"
  xmlns:lo="http://ltsc.ieee.org/wg12/index.html#">

  <dc:comment>Can be adapted to other industry settings</dc:comment>

  <lo:general>
    <lo:identifier>MD05</lo:identifier>
    <lo:title>Team Scenario</lo:title>
    <lo:language>en</lo:language>
    <lo:description>Problem based scenario involving a team meeting in a multimedia
      environment</lo:description>
    <lo:aggregationLevel>1</lo:aggregationLevel>
  </lo:general>

  <lo:lifeCycle>
    <lo:contribute>
      <lo:role>Instructional Designer</lo:role>
      <lo:entity>Joe Smith</lo:entity>
      <lo:date>2000-05-11</lo:date>
    </lo:contribute>
  </lo:lifeCycle>

  <lo:technical>
    <lo:format>XML</lo:format>
    <lo:size>5865</lo:size>
    <lo:requirements>
      <lo:type>Browser</lo:type>
      <lo:name>Internet Explorer</lo:name>
      <lo:minimumVersion>5</lo:minimumVersion>
    </lo:requirements>
    <lo:requirements>
      <lo:requirements>
        <type>Browser</type>
        <name>Netscape</name>
        <minimumVersion>6</minimumVersion>
      </lo:requirements>
    </lo:requirements>
  </lo:technical>

  <lo:educational>
    <lo:interactivityType>expositive</lo:interactivityType>
    <lo:learningResourceType>problemStatement</lo:learningResourceType>
    <lo:interactivityLevel>Very Low</lo:interactivityLevel>
    <lo:semanticDensity>Medium</lo:semanticDensity>
    <lo:context>Industry training</lo:Level>
    <lo:difficulty>Medium</lo:difficulty>
    <lo:strategy>Problem based learning</lo:strategy>
  </lo:educational>
```



```
< lo:relation>
  < lo:kind>IsPartOf</ lo:kind>
  < lo:resource>Certificate 2 in Medial Studies</ lo:resource>
</ lo:relation>
< lo:relation>
  < lo:kind>Relation</ lo:kind>
  < lo:resource>Team Dynamics Article</ lo:resource>
</ lo:relation>

</dataSample>
```

APPENDIX 3

XML Schema - Scenario.xsd

This schema was developed specifically for the data sample and is based on the *XML Schema Candidate Recommendation* (Fallside, 2000) defined by the W3C.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">

  <!--identify the schema-->

  <xsd:annotation>
    <xsd:documentation>
      Team Scenario schema for a learning component using a team meeting scenario and
      a question.
    </xsd:documentation>
  </xsd:annotation>

  <!--General note - complexType elements contain subelements and attributes. SimpleType
  elements contain only a value such as a number or a string. Global elements can be referenced
  in other elements throughout the schema.-->

  <!--global elements-->

  <xsd:element name="teamScenario" type="xsd:teamScenarioType"/>
  <xsd:element name="comment" type="xsd:string"/>
  <xsd:complexType name="listItem" type="item"/>
  <xsd:complexType name="item" type="xsd:string"/>
  <xsd:element name="sectionHeading" type="xsd:string"/>

  <!--teamScenarioType describes the structure of the scenario. It contains five subelements. The
  question element has 'minOccurs' set to 0 and is therefore optional within an instance. the other
  four subelements use the default of 1 and are therefore compulsory. The compulsory elements
  must be used in the order specified and be of the types specified-->

  <xsd:complexType name="teamScenarioType">
    <xsd:sequence>
      <xsd:element name="introduction" type="xsd:string"/>
      <xsd:element name="team" type="characterType"/>
      <xsd:element name="script" type="dialogue"/>
      <xsd:element name="question" type="xsd:multipleChoice" minOccurs="0"/>
      <xsd:element name="task" type="taskType"/>
    </xsd:sequence>
    <xsd:attribute name="creationDate" type="xsd:date"/>
  </xsd:complexType>

  <!--characterType describes the team members. It contains 2 subelements and has a minimum of
  2 since 2 chars is the smallest team. Any number of team members are allowed but more than 5
  will become confusing for the user-->

  <xsd:complexType name="characterType" minOccurs="2" maxOccurs="unbounded">
    <xsd:sequence>
      <xsd:element name="charName" type="xsd:string"/>
```

```

        <xsd:element name="charRole" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

```

<!--dialogue describes the dialogue between team members - who is talking and what they are saying-->

```

<xsd:complexType name="dialogue">
    <xsd:sequence>
        <xsd:element name="charName" type="xsd:string"/>
        <xsd:element name="charSpeech" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

```

<!--task describes a set activity for the user. It can be used in conjunction with the global elements sectionHeading and listitem-->

```

<xsd:complexType name="taskType">
    <xsd:sequence>
        <xsd:element name="taskDescription" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

```

<!--multipleChoice contains 2 elements one for the question or Statement and one for the question options. 'questionOption' then contains 3 sub elements for the number, option and feedback information. The minimum number of options is 2 and the maximum is 5-->

```

<xsd:complexType name="multipleChoice">
    <xsd:sequence>
        <xsd:element name="questOrStatement" minOccurs="1" maxOccurs="1"/>
        <xsd:complexType name="questionOption">
            <xsd:sequence>
                <xsd:element ref="comment" minOccurs="2" maxOccurs="5"/>
                <xsd:element name="optionNumber" type="xsd:optNo"/>
                <xsd:element name="option" type="xsd:string"/>
                <xsd:element name="feedback" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:sequence>
</xsd:complexType>

```

<!--this simple type is used to restrict the number of valid options to 5. This is done by restricting an existing xml schema type namely positiveInteger-->

```

<xsd:simpleType name="optNo">
    <xsd:restriction base="xsd:positiveInteger">
        <xsd:maxInclusive value="5"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

</xsd:schema>

```

APPENDIX 4

XML version of the data sample – team.xml

The XML metadata used here is based on the Scenario.xsd schema in

Appendix 3.

```
<?xml version="1.0" ?>
<teamScenario>

<sectionHeading>Team dynamics</sectionHeading>

<introduction>
  As the Project Manager of a multimedia development team you are concerned that the
  current project is well behind schedule. You can't understand why this should be as you
  have plenty of resources and all the necessary equipment. Your wondering if there might be
  a communications problem. You've just arrived back from a conference and some
  comments that Steve Barnsworth mentioned in his paper on Team Dynamics has got you
  thinking. You have called a team meeting to try and determine the level of communication
  between team members.
</introduction>

<sectionHeading>Meet the team</sectionHeading>

<team>
  <name>Jasmine</name>
  <role>
    Hi I'm Jasmine. I'm a graphic designer and I like to draw characters and create
    interesting interface designs.
  </role>
</team>

<team>
  <name>Paul</name>
  <role>
    Hi I'm Paul and I do a bit of everything but mostly programming stuff using Director and
    Flash.
  </role>
</team>

<team>
  <name>Phil</name>
  <role>
    I'm Phil and JavaScript is my specialty but my main job is doing all the html stuff.
  </role>
</team>

<team>
  <name>Tiffany</name>
  <role>
    I'm Tiffany and I'm the instructional designer. I work form my own office and use email, fax and
    phone calls to communicate with rest of the team. Once a month I come in to check on how
    things are going.
  </role>
</team>
```

```

<team>
  <name>Baz</name>
  <role>
    Hi, I'm Baz I do graphics and some Flash stuff. I'm best at technical type drawing, doing
    perspectives and stuff like that.
  </role>
</team>

<sectionHeading>The Meeting</sectionHeading>

<meetingScript>
  <dialogue>
    <speakemame>Project Manager</speakemame>
  <speech>
    Does anyone have concerns that they are not getting enough information or feedback in order to
    do their job properly?
  </speech>
</dialogue>

  <dialogue>
    <speakemame>Tiffany</speakemame>
  <speech>
    Well a few things have happened that have been quite frustrating but I'm not sure how we can
    prevent them happening.
  </speech>
</dialogue>

  <dialogue>
    <speakemame>Project Manager</speakemame>
  <speech>
    What kind of things? Could you explain exactly what you mean?
  </speech>
</dialogue>

  <dialogue>
    <speakemame>Tiffany</speakemame>
  <speech>
    Well one time I sent over some photos I had taken for unit 2. I emailed them to Paul and he put
    them on your shared drive. The following week I get this message from Baz saying could I check
    the graphics he had drawn for unit 2. It turned out that he had spent 3 days creating graphics for
    the stuff I had sent photos for.
  </speech>
</dialogue>

  <dialogue>
    <speakemame>Baz</speakemame>
  <speech>
    How was I supposed to know there were photos I just read the storyboard and it said picture of
    girl in wheelchair or whatever and so I drew them. Nobody said anything to me about photos.
  </speech>
</dialogue>

  <dialogue>
    <speakemame>Project Manager</speakemame>
  <speech>
    So is this the main reason why there are delays with the graphics?
  </speech>
</dialogue>

  <dialogue>
    <speakemame>Baz</speakemame>
  <speech>
    Well not really, the main problem was having to re-do all the garphics in unit 2 because Tiffany
    wasn't happy with them.
  </speech>
</dialogue>

```

```

    <dialogue>
      <speakername>Tiffany</speakername>
      <speech>
You make it sound like I'm being fussy Baz but the whole unit is about body language and facial
expressions and there just wasn't enough detail in the graphics to show the points I was trying to
get across.
      </speech>
    </dialogue>

<dialogue>
  <speakername>Baz</speakername>
  <speech>Well that's not my area. I hate drawing faces.</speech>
</dialogue>

<dialogue>
  <speakername>Project Manager</speakername>
  <speech>Would it have been better if you had done those Jasmine?</speech>
</dialogue>

<dialogue>
  <speakername>Jasmine</speakername>
  <speech>
Yeah I guess so but I had a lot to do with the graphics for that big animation in unit 3.
  </speech>
</dialogue>

</meetingScript>

<question>
  <multipleChoice>
    <statement>What seems to be the main problem here?</statement>
    <questionOption>
      <optionNumber>1</optionNumber>
      <option>The team don't get along together</option>
      <feedback>
The team probably do get along well but they may need help in organising structured
communication that will assist them with their work.
      </feedback>
    </questionOption>

    <questionOption>
      <optionNumber>2</optionNumber>
      <option>Lack of communication change</option>
      <feedback>
Yes, it seems that they could work more efficiently if they communicated more with respect to
their roles and responsibilities.
      </feedback>
    </questionOption>

    <questionOption>
      <optionNumber>3</optionNumber>
      <option>A lack of formal procedures</option>
      <feedback>
There may be a problem in this area but some of the issues raised indicate a need for
communication on a less formal level too.
      </feedback>
    </questionOption>
  </multipleChoice>
</question>

```

```
<sectionHeading>Task</sectionHeading>

<task>
  <taskDescription>
    There seems to be a lot of room for improvement here. Now that the meeting is finished you can start work on your report. Think about what the developers have said and refer to the notes you took during the meeting. You might like to have a look at Barnsworth's paper for some ideas. You can also refer to the Project Management guide or ask advice from one of your colleagues. The format of your report should follow the instructions set out in the Office Policy and Procedures document. You should include:
  </taskDescription>

  <orderedList>
    <item>List of problems identified.</item>
  </orderedList>

  <orderedList>
    <item>List of strategies to address these problems.</item>
  </orderedList>

  <orderedList>
    <item>A suggested implementation plan.</item>
  </orderedList>
</task>

</teamScenario>
```

APPENDIX 5

RDF Scheme

This schema was created to identify the learning outcomes associated with the data sample and is based on the related ontology.

```
<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <!--note that this RDF schema would be referenced from an instance document using a
  namespace declaration and an abbreviation such as lo. This would allow unambiguous reference
  to the class "LearningOutcome". The abbreviated form of syntax has been used.-->

  <!--there are four classes representing the four concepts in the ontology-->

  <rdfs:Class ID="LearningOutcome">
    <rdfs:subClassOf
      rdf resource=" http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdfs:Class>

  <rdf:Class rdf:ID="ElementCompetency">
    <rdfs:subClassOf rdf:resource="#LearningOutcome"/>
  </rdf:Class>

  <rdf:Class rdf:ID="Theme">
    <rdfs:subClassOf rdf:resource="#LearningOutcome"/>
  </rdf:Class>

  <rdf:Class rdf:ID="PerformanceCriteria">
    <rdfs:subClassOf rdf:resource="#ElementCompetency"/>
  </rdf:Class>

  <!--there are four properties with range Literal that are used to identify text descriptions of
  resources-->

  <rdf:Property ID="loDescribe">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  </rdf:Property>

  <rdf:Property ID="eDescribe">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  </rdf:Property>

  <rdf:Property ID="pcDescribe">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  </rdf:Property>

  <rdf:Property ID="tDescribe">
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  </rdf:Property>

  <!--there are four properties constrained by both a range and a domain to identify relationships
  between the classes-->
```



```
<rdf:Property ID="relatesToElement">
  <rdfs:range rdf:resource="#ElementCompetency"/>
  <rdfs:domain rdf:resource="#LearningOutcome"/>
</rdf:Property>

</rdf:Property ID="containsPC">
  <rdfs:range rdf:resource="#PerformanceCriteria"/>
  <rdfs:domain rdf:resource="#ElementCompetencie"/>
</rdf:Property>

</rdf:Property ID="relatesTo">
  <rdfs:range rdf:resource="#Theme"/>
  <rdfs:domain rdf:resource="#ElementCompetency"/>
</rdf:Property>

</rdf:Property ID="isAbout">
  <rdfs:range rdf:resource="#ElementCompetencie"/>
  <rdfs:domain rdf:resource="#Theme"/>
</rdf:Property>
```

APPENDIX 6

Java Program (using SAX)

```
//This is java program that extracts data from the team.xml documnet by using the SAX 1.0
parser.
//The objective of this program is to print out each team members details as stored in the <role>
elements of the team.xml document.
//written by Alison Anderson oct 2000

//import the Sax classes required

import org.xml.sax.*;

//the HandlerBaswe class of SAX1 contains a standard implementation of methods from the
//DocumentHandler interface

public class teamFinder extends HandlerBase
{

    //declare variables for a flag to track the current element and string to hold the data found

    private StringBuffer teamRole = new StringBuffer();
    private boolean isRole = false;

    //invoke the read method

    public static void main(String[] args) throws Exception
    {
        teamFinder readerObj = new teamFinder();
        readerObj.read();
    }

    //create a parser object

    public void read () throws Exception
    {
        Parser parserObj = new com.jclark.xml.sax.Driver();
        parserObj.setDocumentHandler (this);
        parserObj.parse ("file:///c:/code/team.xml");
    }

    //use the startDocument method to print a start message

    public void startDocument() throws SAXException
    {
        System.out.println("\nmeet the team...");
    }

    //use the endDocument method to print an end message

    public void endDocument() throws SAXException
    {
        System.out.println("\n...hope you guys have fun");
    }
}
```

//the startElement method catches the start of each element.
//if the element = 'role' the contents is stored in the teamRole variable

```
public void startElement(String name, AttributeList atts) throws SAXException
{
    if (name.equals("role"))
    {
        isRole = true;
        teamRole.setLength(0);
    }
    else
    {
        isRole = false;
    }
}
```

//after the element is read the contents of teamRole is output to the screen

```
public void endElement (String name) throws SAXException
{
    if (isRole)
    {
        System.out.println("\n" + teamRole.toString());
        isRole = false;
    }
}
```

```
public void characters(char[] chars, int start, int len)
    throws SAXException
{
    if (isRole)
        teamRole.append(chars, start, len);
}
```

When executed the Java program gives the following results.

```
C:\code>javac teamFinder.java
```

```
C:\code>java teamFinder
```

meet the team...

Hi I'm Jasmine. I'm a graphic designer and I like to draw characters and create interesting interface designs.

Hi I'm Paul and I do a bit of everything but mostly programming stuff using Director and Flash

I'm Phil and Javascript is my specialty but my main job is doing all the html stuff.

I'm Tiffany and I'm the instructional designer. I work from my own office and use email, fax and phone calls to communicate with rest of the team. Once a month I come in to check on how things are going.

Hi, I'm Baz I do graphics and some Flash stuff. I'm best at technical type drawing, doing perspectives and stuff like that.

...hope you guys have fun

```
C:\code>
```

APPENDIX 7

DOM Example

This HTML document (chooseElement.html) demonstrates how DOM methods can be used to extract and manipulate individual elements from a resource.

```
<html>
<head><title>Team DOM</title>

<script language="JavaScript">

    //create a DOM object
    var objDOM;
    objDOM = new ActiveXObject("MSXML.DOMDocument");

    //set asynchronous download to false. If set to true (the default setting),
    //the load method returns control to the caller before the download is finished.
    objDOM.async = false;

    //load an xml document to work with, team.xml
    objDOM.load("team.xml");

    //extract elements from team.xml

    //extract introduction element
    var objIntroNode = objDOM.selectSingleNode("/teamScenario/introduction");
    var introElement=(objIntroNode.firstChild.nodeValue);

    //extract team 'role' elements
    var objNodeList = objDOM.getElementsByTagName("role")
    var teamMembersNumber=(objNodeList.length);

    //extract team 'dialogue' element and child nodes
    var objNodeList2 = objDOM.getElementsByTagName("dialogue")
    var dialogueNumber=(objNodeList2.length);

    //extract question options
    var objNodeList3 = objDOM.getElementsByTagName("option")
    var questNumber=(objNodeList3.length);

    //extract task element
    var objTaskNode = objDOM.selectSingleNode("/teamScenario/task/taskDescription");
    var taskElement=(objTaskNode.firstChild.nodeValue);

function showIntro() //this function will display the intro from team.xml
{
    var introRadio = window.document.f1.r1;
```

```

        if (introRadio.checked == true) {
            window.document.f1.r2.checked = false;
            window.document.f1.r3.checked = false;
            window.document.f1.r4.checked = false;
            window.document.f1.r5.checked = false;
            window.document.f1.ta1.value=(introElement);
        }
    }

function showTeam() //this function will show the team roles from team.xml
{
    var teamRadio = window.document.f1.r2;

    if (teamRadio.checked == true) {
        window.document.f1.r1.checked = false;
        window.document.f1.r3.checked = false;
        window.document.f1.r4.checked = false;
        window.document.f1.r5.checked = false;
        var member = new Array();
        var i
        for (i = 0; i < (teamMembersNumber); i++){
            member[i] = objNodeList.item(i).text;

            window.document.f1.ta1.value=(member[0] + "\n" + "\n" + member[1] + "\n" + "\n" +
            member[2] + "\n" + "\n" + member[3] + "\n" + "\n" + member[4]);
        }
    }
}

function showMeeting() //this function shows the dialogue from team.xml
{
    var meetingRadio = window.document.f1.r3;

    if (meetingRadio.checked == true) {
        window.document.f1.r1.checked = false;
        window.document.f1.r2.checked = false;
        window.document.f1.r4.checked = false;
        window.document.f1.r5.checked = false;
        var dialogue = new Array();
        var i
        for (i = 0; i < (dialogueNumber); i++){
            dialogue[i] = objNodeList2.item(i).text;

            window.document.f1.ta1.value=(dialogue[0] + "\n" + "\n" + dialogue[1] + "\n" + "\n" +
            dialogue[2] + "\n" + "\n" + dialogue[3] + "\n" + "\n" + dialogue[4]+ "\n" + "\n" + dialogue[5]+
            "\n" + "\n" + dialogue[6]+ "\n" + "\n" + dialogue[7]+ "\n" + "\n" + dialogue[8]+ "\n" + "\n" +
            dialogue[9]+ "\n" + "\n" + dialogue[10]);
        }
    }
}

function showTask() //this function shows the task element from team.xml
{
    var taskRadio = window.document.f1.r4;

    if (taskRadio.checked == true) {
        window.document.f1.r1.checked = false;
        window.document.f1.r2.checked = false;
        window.document.f1.r3.checked = false;
    }
}

```

```

        window.document.f1.r5.checked = false;
        window.document.f1.ta1.value=(taskElement);
    }
}

function showQuestOptions() //this function shows the multiple choice question options
{
    var questRadio = window.document.f1.r5;

    if (questRadio.checked == true) {
        window.document.f1.r1.checked = false;
        window.document.f1.r2.checked = false;
        window.document.f1.r3.checked = false;
        window.document.f1.r4.checked = false;

        var quest = new Array();
        var i
        for (i = 0; i < (questNumber); i++){
            quest[i] = objNodeList3.item(i).text;
            window.document.f1.ta1.value=(quest[0] + "\n" + "\n" + quest[1] + "\n" + "\n" + quest[2]);

        }
    }
}
</script>

```

<p>This HTML page extracts data from XML elements in the team.xml document using JavaScript and the DOM API</p>
 <P>Choose an element to see the corresponding data.</p>

```

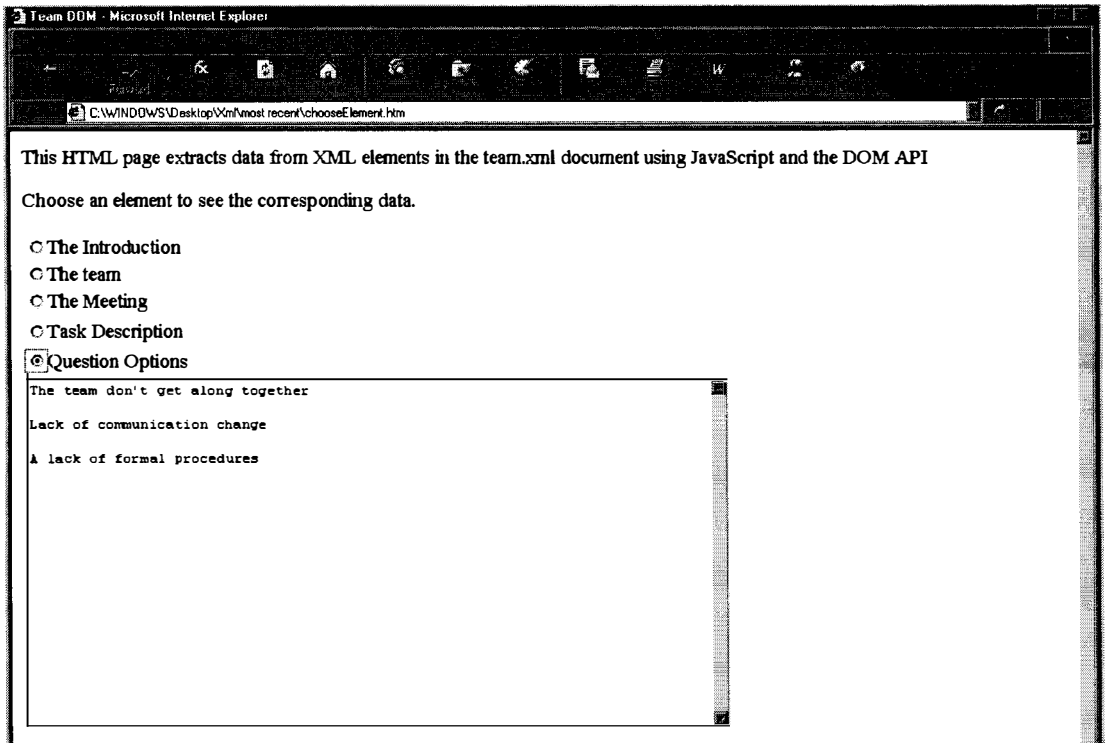
<form name="f1">
<table>
  <tr>
    <input type="radio" name ="r1" onClick="showIntro();">The Introduction
  </tr>
  <tr>
    <input type="radio" name ="r2" onClick="showTeam();">The team
  </tr>
  <tr>
    <input type="radio" name ="r3" onClick="showMeeting();">The Meeting
  </tr>
  <tr>
    <input type="radio" name ="r4" onClick="showTask();">Task Description
  </tr>
  <tr>
    <input type="radio" name ="r5" onClick="showQuestOptions();">Question Options
  </tr>
  <textarea name="ta1" cols=80 rows=20></textarea>
</tr>
</table>

</form>

</body>
</html>

```

This screen capture shows chooseElement.html as displayed by IE 5.5.



APPENDIX 8

XSLT style sheet

This style sheet displays information from a source XML document that contains any of the elements from the data sample (team.xml).

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">

<html>
<head>
<title>Group Meeting</title>

    <!-- Javascript starts here-->

<SCRIPT LANGUAGE="JavaScript1.2">

var fbk = new Array();

<xsl:for-each select="/teamScenario/question/multipleChoice/questionOption">
fbk[ <xsl:value-of select="optionNumber" /> ] = " <xsl:value-of select="feedBack" /> ";
</xsl:for-each>

</SCRIPT>

</head>
<body>
<table width="550">
<h3>Team Dynamics</h3>

<p>
<xsl:value-of select="/teamScenario/introduction" />
</p>

<h3>Meet the team</h3>
<xsl:for-each select="/teamScenario/team">
<p>
<xsl:value-of select="role" />
</p>
</xsl:for-each>

<h3>The meeting</h3>
<br />
<table WIDTH="60%" BORDER="1" CELSPACING="0" CELLPADDING="5"
BORDERCOLOR="#000066">
<xsl:for-each select="/teamScenario/meetingScript/dialogue">
<tr>
    <td BGCOLOR="#BFDFFF">
        <p> <b> <xsl:value-of select="speakername" /> </b> </p>
        <p> <xsl:value-of select="speech" /> </P>
    </td>
</tr>
</xsl:for-each>
```

```

</table>
<form name="f1">
<table WIDTH="100%" BORDER="0" CELSPACING="0" CELLPADDING="5">
    <tr VALIGN="TOP">
        <td><p>
            <xsl:value-of select="/teamScenario/question/multipleChoice/Statement/">
        </p>
        </td>
    </tr>
</table>
<br />
<table WIDTH="450" BORDER="0" CELSPACING="0" CELLPADDING="2"
BGCOLOR="#BFDFFF">
    <xsl:for-each select="/teamScenario/question/multipleChoice/questionOption">
        <tr VALIGN="TOP">
            <xsl:if test="position() = 1">
                <td>
                    <INPUT TYPE="radio" NAME="r1" VALUE="1" onClick="f1.ta1.value=fbk[1] />
                </td>
            </xsl:if>
            <p><xsl:value-of select="option"/></p>
        </tr>
        <xsl:if test="position() = 2">
            <td>
                <INPUT TYPE="radio" NAME="r1" VALUE="2" onClick="f1.ta1.value=fbk[2]" />
            </td>
            <p><xsl:value-of select="option"/></p>
        </td>
        </xsl:if>
        <xsl:if test="position() = 3">
            <td>
                <INPUT TYPE="radio" NAME="r1" VALUE="3" onClick="f1.ta1.value=fbk[3]" />
            </td>
            <p><xsl:value-of select="option"/></p>
        </td>
        </xsl:if>
        <xsl:if test="position() = 4">
            <td>
                <INPUT TYPE="radio" NAME="r1" VALUE="4" onClick="f1.ta1.value=fbk[4]" />
            </td>
            <p><xsl:value-of select="option"/></p>
        </td>
        </xsl:if>
        <xsl:if test="position() = 5">
            <td>
                <INPUT TYPE="radio" NAME="r1" VALUE="5" onClick="f1.ta1.value=fbk[5]" />
            </td>
            <p><xsl:value-of select="option"/></p>
        </td>
        </xsl:if>
    </tr>
</xsl:for-each>
    <tr VALIGN="TOP" ALIGN="CENTER">
        <td COLSPAN="2">

```

```

<TEXTAREA NAME="ta1" COLS="40" ROWS="7" WRAP="PHYSICAL" />
</td>
      </tr>
    </table>
  </form>
  <h3>Task</h3>
  <table WIDTH="60%" BORDER="0" CELLSPACING="0" CELLPADDING="5">
    <tr VALIGN="TOP">
      <td>
        <p><xsl:value-of select="/teamScenario/task /></p>
        <xsl:for-each select="/teamScenario/listItem">
          <ul>
            <p><li><xsl:value-of select="item" /></li></p>
          </ul>
        </xsl:for-each>
      </td>
    </tr>
  </table>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```