UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING AND APPLIED SCIENCE

School of Electronics and Computer Science

Exploiting Metadata Links to Support Information Retrieval in Document Management Systems

By

Mirjana Andrija Andric

A thesis submitted for the degree of Doctor of Philosophy

March 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

EXPLOITING METADATA LINKS TO SUPPORT INFORMATION RETRIEVAL IN DOCUMENT MANAGEMENT SYSTEMS

by Mirjana Andrija Andric

Contemporary approaches to managing collections of documents typically lack support for flexible metadata definition, browsing and use for document retrieval. Furthermore, record of people's browsing and searching experiences is rarely utilised. Users are forced to resort to keyword searching without knowing which keywords exist in the domain and which were used by other searchers. In an organisational setting, in which documents share a common and related set of concepts, and where documents are used by a closed group of collaborators, these issues would have a strong impact.

A hypothesis of this dissertation is that utilising metadata linked into the ontological structure for the tasks of supporting information retrieval, offers an advantage over the traditional full text searching techniques. The "à la" (Associative Linking of Attributes) concept presented in this thesis demonstrates how a system for managing document collections in an organisational setting, could be enhanced to support the information delivery process. It can be considered as a pre-Semantic Web application, a recommender facility that provides assistance to locating items by utilising hypertextually linked metadata. The contribution of this thesis lies in three areas:

- Novel approach to metadata linking: using ontological zzstructures and ZigZag linking;
- Novel approach to using the metadata network for document retrieval: "Query by association";
- Evaluation that compares the usage of the "à la" prototype system with the traditional approach.

The "à la" prototype evaluation study compares free text querying using a full text search approach with the "à la" method for finding relevant documents. Precision (fraction of relevant search results) and serendipity (fraction of novelty or positively surprising items in a search result) were the chosen metrics. The study findings indicate that "à la" performs better in carrying out general-concept queries that fall within a project's knowledge domain. The evaluation concludes that "à la" can successfully enrich a typical document management system in order to improve the user's searching experience, by bringing not only the expected relevant search results, but also the items which are serendipitous. Moreover, browsing metadata connected in a Zigzag fashion can better aid understanding of the existing domain. This can in turn benefit knowledge engineers in their first step towards building a more formal organisational ontology. It is concluded that the "à la" method has the potential to be applicable towards establishing principles of the Semantic Web within a corporate environment.

Table of contents

List of figures

List of tables

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Dr Wendy Hall, for providing her support concisely and exactly when needed and for steering me from falling into numerous "PhD student traps". Grateful thanks are due to Dr Les Carr for his support and valuable discussions.

Special thanks go to the users from the company Finsoft Ltd. London, for participating in the study and providing the data set for it. Especially, I am thankful to Mr Zoran Zmajkovic who helped to make the experiment possible.

Although very difficult and constantly competing with my demanding job, my part time studies have greatly broadened my horizons. I owe a lot of gratitude to the members of the IAM group and colleagues from the University of Southampton, for our discussions and helpful reviews, and for keeping me in touch with the academic environment: Dr Panagiotis Melas, Dr Andrew Perkins, Vladimir Mircevski and numerous others. Dr. Gary Wills of IAM, provided me with valuable insights into the area of user evaluation, for which I am indebted.

On a more personal note, I must thank my family and friends for being patient during these years. I am finally able to answer their eternal question about my dissertation finishing date.

Definitions and abbreviations used

à la	Associative Linking of Attributes
AWOCADO	Adaptive Workflow Controller and Document Organiser
CM	Content Management
CMS	Content Management System
Dataset	A collection of computer-readable data records
DM	Document Management
DMS	Document Management System
HCI	Human Computer Interaction
HTML	Hypertext Mark up language: The programming language using a set of tags and rules to create Web pages which are later processed by Internet Browsers
IEEE	The Institute of Electrical and Electronics Engineers
intranet	An Internet type communication and information sharing system designed for the enterprise rather than for public use
IR	Information Retrieval
Java	A programming language designed for truly cross-platform applications
Java Script	A scripting language that utilises Java to permit more powerful HTML page functions
ONA	Ontology Network Analysis
TF-IDF	Term Frequency – Inverse Term Frequency
UNC	Universal Naming Convention is a way to identify a shared file in a computer in Windows and other operating systems. The UNC name format is: \\servername\sharename\path\filename.
URL	Uniform Resource Locator describes a location of a resource on a network
Web Server	A hardware and software platform that acts as an Internet server and or host creating and serving HTML pages and associated data for communication to the Internet
WWW	World Wide Web
XML	Extensible Mark-up Language: A programming language/data representation standard, enhanced in comparison to HTML

XSL	Extensible Stylesheet Language: Represents a stylesheet for the XML data

Chapter 1

Introduction

The first chapter of this thesis provides an overview of the research which is described in the following chapters. It identifies the importance of the research topic, explains the motivation, sets the scope and presents the expected contribution of this research.

1.1 Motivation

"If it was so very easy to look things up, how would our vocabulary develop, how would our habits of exploring the intellectual domain of others shift, how might the sophistication of practical organisation mature (if each person can so quickly and easily look up applicable rules), how would our education system change to take advantage of this new external symbol manipulation capability of students and teachers (and administrators)?" - Douglas Engelbart was wondering in the 'ancient' (for computer sciences) sixties (Engelbart 1962).

Things have changed considerably since Engelbart wrote about those issues in the 1960's, but many of the problems related to finding and retrieving in a timely manner a needed piece of information still remain. The fact is, an average individual is nowadays flooded with electronic information coming from various sources. Knowledge undoubtedly represents the greatest asset of a modern organisation (Butler 2004) and whether we realise it or not, search is becoming the *de facto* way of finding information (Olstad & Seres 2005). For today's organisations, such as companies or universities, the situation appears to be very similar – knowledge management is not extensively developed or adopted. A certain amount of knowledge typically resides in a number of office or departmental documents, usually in an unstructured form of various types and formats (Popkin & Cushman 1993). It is not always easy to locate those documents. Documents can be stored on a number of personal or shared disks, corporate intranets, repositories or in document management systems. Regardless of whether the organisation employs some kind of computer-based system for managing documents (i.e. a document management system) locating the desired documents represents a challenge. Documents are mostly unclassified or inconsistently classified, various versions are usually scattered around or circulated via email and quite a few documents are inaccessible by people

other then their authors (and even not by the authors in some cases!).

In order to find a particular item of knowledge within an organisational environment, people resort to keyword searching as in the case of the Web. The latest development in this area saw a proliferation of personal desktop search tools (Chirita *et al.* 2005) such as Google Desktop[1] or the announcements of search-aware capabilities of new operating systems such as Microsoft's Vista[2] (Dumais *et al.* 2003). Nevertheless, the keyword-based way of locating documents has the following disadvantages:

- Keyword locating can be inefficient as many relevant results can be missed for the reason that they did not contain the requested search term. Also, broad queries can bring too many results. A recent Ark survey found that, despite the significant amount of time knowledge-workers spend searching for relevant information, the overwhelming majority of searches return largely irrelevant or inaccurate results; 50% of the results is actually relevant or accurate[3];
- Typical office documents seldom have embedded links to connect them with each other, therefore finding interesting documents by following links is usually not available;
- People are not aware of other people's searching experiences and therefore can not reuse them. Searchers do not know which keywords their colleagues (communities of practice) used to search and which documents they found relevant;
- If people can not find an existing colleague's (or even their own) documents, they create new documents thus duplicating the work and risking using wrong versions later on.

Helping users to find relevant information using content similarities or suggesting search criteria as well as search results of similar users, opened the door to research in so called recommender systems (see Chapter 3: Recommender Systems). Introduction of metadata (see Chapter 2: Document Management and Information Retrieval) and use of ontologies and the Semantic Web (see Chapter 4: Knowledge

---

[1] <http://desktop.google.com>

[2] <http://www.microsoft.com/windowsvista>

[3] Ark Group keynote. *2nd annual optimising search and retrieval conference 2005.* London, UK

Management and ZigZag) promises to dramatically improve finding relevant documents. However the metadata by which the content on the Web or a document management system can be classified, is still largely lacking. Although there have been some advances to this end in the area of the Semantic Web, intermediary solutions that pave the path for the incoming Semantic Web are still needed.

1.2 Overview

The research presented in this thesis focuses on the issues that document management systems have in document retrieval. The hypothesis of this thesis is that using document metadata *(document attributes),* and introducing *associative adjacency links* between those attributes can facilitate finding and recommending relevant documents.

The "à la" (Associative Linking of Attributes) system described in this dissertation demonstrates how a system for managing document collections in an organisational setting could be enhanced to support the information delivery process. The presented solution attempts to alleviate the aforementioned disadvantages of keyword-based searching by pre-processing the document using text mining techniques and consequently saving the keywords and other metadata for aiding future queries. "à la" is a recommender facility that provides assistance to keyword-based search by utilising hypertextually linked metadata. It can be considered as a "pre-Semantic Web" application that aims to promote the Semantic Web principles in the corporate environment.

For this to be achieved "à la" first harvests the metadata, i.e. document attributes, and promotes them to be first class objects. Then it computes attribute links based on the statistical co-occurrences and heuristics. Metadata link weights are used to indicate the relationship strength. Associations or links between documents themselves can be dynamically and implicitly deduced from the links of their attributes.

Discovered attributes and their associations are stored in a high-dimensional informational space inspired by ZigZag (Nelson 1998), in the form of a somewhat extended zzstructure. ZigZag in its core can be defined as a space consisting of ordered lists of basic units (cells) intersecting at multiple points. Cells contain the

individual metadata instances concatenated in lists using a special kind of adjacency links which, in this thesis, we are going to refer to as *ZigZag links*[4]. The obtained metadata network is used for querying and it is finally visualised in the hypertextual zzstructure browser. The recommender facility employs this information in order to assist users in expressing their information-search queries as well as to automatically offer quality recommendations.

1.3 Contribution and Scope

The hypothesis of this dissertation is that utilising metadata ZigZag linked into an ontological structure (i.e. metadata vocabulary structure), for the tasks of supporting information retrieval, offers an advantage over traditional full text searching techniques.

Contributions of this thesis can be summarised in the following points:

- *Novel approach to metadata linking: using ontological zzstructures*
  The approach consists of:
  - extracting metadata from the document management system;
  - discovering attribute associations using statistical analysis, text data mining and recommender systems techniques;
  - and finally weaving documents, metadata and their ZigZag links into an ontological hypertext structure in a form of a somewhat extended zzstructure.
- *Novel approach to using the metadata network for document retrieval: "Query by association"*
  The contribution of this approach has two aspects. Firstly, traversing metadata network links is used by a search algorithm to locate associated metadata and documents of interest, in order to create a search result and recommendations for query expansion or modification. Secondly, the ontological metadata structure is visualised in a hypertext browser for navigating attributes and documents by the end-users.
- *Evaluation that compares usage of a prototype system with a traditional approach*

[4] The adjacency links in zzstructures will be called ZigZag links, while the term *link* will be used with a meaning of hypertext link (more details in section 2.3)

An initial system evaluation with users has been conducted in two areas, software engineering and education, in order to determine how the searching aspect of the system behaves compared to a classical solution of full text searching. Measuring serendipity of the system was introduced in addition to standard approaches to the evaluation of information retrieval and recommender systems.

The scope of this research is as follows:

- Text analysis and machine learning in document collections (introduced in Chapter 2 *Document Management and Information Retrieval* and Chapter 3 *Recommender Systems*; usage of adopted techniques presented in Chapter 6 *"à la": Associative Linking of Attributes*)
- Content, collaborative and knowledge-based recommender systems (detailed in Chapter 3 *Recommender Systems;* usage of adopted techniques presented in Chapter 5 *Initial Research in Document Management, Recommending and ZigZag* and Chapter 6 *"à la": Associative Linking of Attributes*)
- Knowledge representation, Web mining and text mining (introduced in Chapter 4, *Knowledge Management and ZigZag*; approach of this dissertation presented in Chapter 5 *Initial Research in Document Management, Recommending and ZigZag*, and Chapter 6 *"à la": Associative Linking of Attributes*)
- Searching by similarity and associations (approach of this dissertation presented in Chapter 6 *"à la": Associative Linking of Attributes*)

1.5 Thesis Structure

This thesis is presented in eight chapters. The subsequent seven chapters are as follows:

Chapter 2, Document Management and Information Retrieval, sets the scene and provides necessary background into the research area.

Chapter 3, Recommender Systems, introduces a class of systems that supports users in a collaborative environment.

Chapter 4, Knowledge Management and ZigZag, portrays techniques for knowledge

organisation and management. Background material on ontologies and the Semantic Web is presented. The idea of ZigZag as a paradigm for information storage and representation is also introduced.

After providing an overview of the theoretical context, the reminder of this thesis presents the work carried out in the course of the investigations.

Chapter 5, Initial Research in Document Management, Recommending and ZigZag, describes the first stage of this research. It discusses three systems that were built in order to investigate the aforementioned research areas:

- AWOCADO, an experimental document management system built with the aim of providing a test bed for investigating how is metadata used for searching;
- MAGENTA, an experimental recommender facility supporting guided tours for documents (Websites) where steps are suggested by utilising a recommender system;
- ZZDirectory, research into the use of ZigZag for storing and browsing metadata (ontologies), applied on Website Directories (Catalogues).

Chapter 6, "à la": Associative Linking of Attributes, elaborates the final stage of this thesis research. It presents an idea of creation and utilisation of Zigzag metadata links in order to support searching and browsing document collections. The chapter describes a prototype system named "à la", a step forward in adding a content-based recommendation and visualisation service to the document management system AWOCADO. The relationship between the systems described in chapter 5 and the "à la" system is shown in figure 1-1.

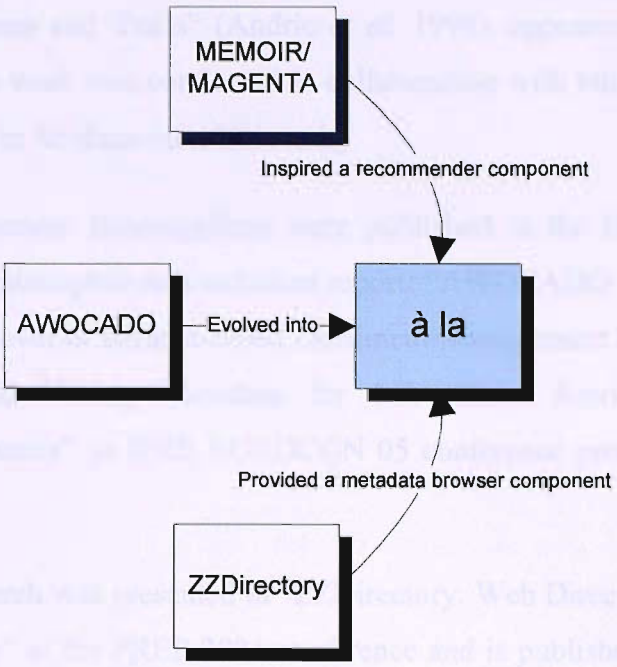


Figure 1-1: The "à la" system evolution

Chapter 7, "à la" Evaluation, provides an overview of evaluation methods and presents results obtained while evaluating the search aspect of the "à la" system. The method used in the "à la" system is compared to a reference method that uses full text search. Conclusions that will support the thesis hypothesis are drawn.

Chapter 8, Future Work and Conclusion, summarises the thesis contribution, discusses the findings, envisages a roadmap for future research and concludes the thesis.

Additional material for Chapters 2, 5 and 7, aimed to present an overview of the commercial content management systems, to illustrate the usage of initial prototype systems, and present the details on statistical tests, is presented in appendixes A-D.

1.6 Declaration

The work described in this thesis was carried out by the author between October 2000 and October 2005 (with some early work conducted in 1998).

The initial research in recommender systems was presented as a software demonstration at the ACM Hypertext 1998 conference, "MAGENTA - **M**EMOIR **A**ssisted **G**uided tours **EN**gineered from **T**rails using **A**gents" (Griffiths & Andric 1998) and as a paper in ETRAN 98, "Dynamically Generating Branching Guided

Tours using Agents and Trails" (Andric *et al.* 1998), appearing in the conference proceedings. The work was conducted in collaboration with other co-authors during the author's visit to Southampton University.

Metadata management investigations were published in the EPRINTS archive at University of Southampton as a technical report: "AWOCADO: Using Metadata for Information Retrieval in Intranet-based Document Management Systems" (Andric & Hall 2005b) and "Using Metadata for Information Retrieval in Document Management Systems" in IEEE EUROCON 05 conference proceedings (Andric & Hall 2005c).

The ZigZag research was presented in "ZZDirectory: Web Directory Browsing using ZigZag Paradigm" at the PREP 2004 conference and is published in the conference proceedings (Andric *et al.* 2004a).

The ideas of the initial "à la" research were presented as a poster and software demonstration at the ACM Hypertext 2003, "Implicit Document Recommending Using Associative Linking of Attributes" (Andric *et al.* 2003). The work is published in the poster proceedings.

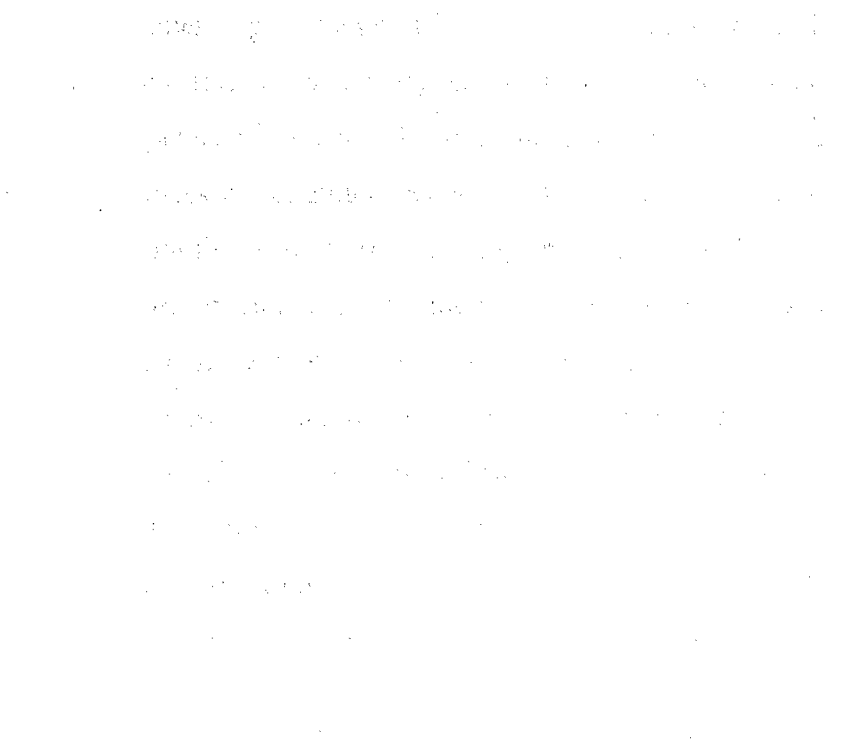
The "à la" system description and the preliminary evaluation results were published in the ACM Document Engineering (DocEng) Symposium 2004 proceedings, "Assisting Artifact Retrieval in Software Engineering Projects" (Andric *et al.* 2004b).

Applying the "à la" concept in the area of supporting the authoring of educational material and the evaluation results was presented at the Artificial Intelligence in Education (AIED) 2005 conference and published in the proceedings as: ""à la" in Education: Keywords Linking Method for Selecting Web Resources" (Andric *et al.* 2005a). The detailed work is to be published in the International Journal of Knowledge and Learning as: "Keywords Linking Method for Selecting Educational Web Resources à la ZigZag" (Andric *et al.* 2007).

This work reused part of the research conducted in the QuIC: Queries in Context

project[5] (supported by ESPRC GR/M77086).

"ZigZag" and "Xanadu" are trademarks of Dr T. Nelson, Fellow of the Oxford Internet Institute. This dissertation uses the name "ZigZag" and acknowledges that the U.S. trademark on the use of "ZigZag" for computer software has been registered. The U.S. patented process "Interactive connection, viewing, and manoeuvring system for complex data" in ZigZag, is also acknowledged (patent no 6,262,736, July 17, 2001).



[5] QUIC: Queries in Context project Website. Available from World Wide Web: <http://www.iam.ecs.soton.ac.uk/projects/quic/>

Chapter 2

Document Management and Information Retrieval

This chapter reviews the fundamental concepts related to document management, upon which this research is based. It also introduces the fundamentals of information retrieval. It then looks into hypertext and linking and their relationship to retrieval.

2.1 Introduction

In order to define what the term *document management* mean, we must first look into the concept of a *document* itself. The word *document* comes from the Late Latin (3rd to 6th century) *documentum* meaning *official paper*, and from the Latin *lesson* or *proof* (Gilheany 2001). In the past, a traditional document was commonly considered as a piece of paper, e.g. a memo, a letter, a plan or an invoice. That piece of paper presented some information, usually text, or text with graphics, arranged in a certain way over the physical surface, all with one objective: to provide communication. However, advances in computer technology have brought around radical changes in a concept of handling information: storing, presenting, and communicating it in so called *electronic* documents. A document, in today's broader sense, can be considered as an entity containing information, or an information container (Popkin & Cushman 1993). Contemporary electronic documents can be thought of as information composites coming from different sources and grouped for the purpose of communication and the development of information (Hendley 2005), while becoming active and mobile in addition (Dourish *et al.* 2000). Documents can then be defined as identifiable recordings of information (Gilheany 2001).

With the rise of the Web, the word document took on a new meaning. Web pages, emails, articles from Web newsgroups or usenet news (McLellan 1997), are nowadays all considered documents.

2.2 Document Management and Enterprise Content Management

There is, as yet, no officially accepted definition for the term *Document Management*, which is, nevertheless, used widely in practice. Why is this and will it ever become possible to finally define that term? The answer resides in the various

types of documents that can be managed by a document management system and upon which, to a great extent, the system functions themselves depend. There is no single list of features that best describe what a document management system should do. Briefly, document management systems manage creating, editing, versioning and accessing documents of various types through the whole document life cycle (Hendley & Broadhurst 2000; Hendley 2005). The phases of a typical life cycle include creating, classifying, revising, archiving, accessing and destroying documents. Document management systems should also provide flexible user management based on user roles such as readers, editors, administrators, etc. This functionality should cater for the access rights and permissions assigned to individuals or groups that can be overridden if necessary.

Document management systems experienced growth in popularity during the 1980s and 1990s (Hendley 2005), when their main aim was to organise documents such as spreadsheets, drawings, text processor documents and scanned material. At that time document management systems used to be called *Electronic Document Management (EDM) Systems*. However, since the 90s, the Internet has redefined the way organisations create and publish internal information. This has led to the development of the *Web Content Management Systems* and *intranet portals*. Web content management systems are responsible for creating, managing parts of documents or multimedia content, and delivering them, most commonly, as a part of a Website, as described in (Butler Group 2004). Intranet portals are company wide information delivery platforms, or to put it more simply: they are Websites that aggregate information from various sources. Both Web content management systems and intranet portals are very close to the document management systems. Digital Libraries, libraries whose content is managed using digital computers, can also be considered as a type of document management system.

Enterprise Content Management (*ECM*) is more recent terminology that has been adopted to denote a group of technologies related to document management (Butler Group 2004). The phrase was coined by AIIM[6], a major body in this field, in 2001, as pointed out by Harris-Jones (2002). Industry leaders such as Butler group predict

[6] AIIM – The Enterprise Content Management Association, collection of resources, Available from World Wide Web: <http://www.aiim.org and <http://www.aiim.org.uk/index1.asp>

convergence of technologies as a future trend in any content management related area (Butler 2004). The diagram in figure 2-1 shows the historical appearance and subsequent convergence of ECM related technologies (McGrath 2003).

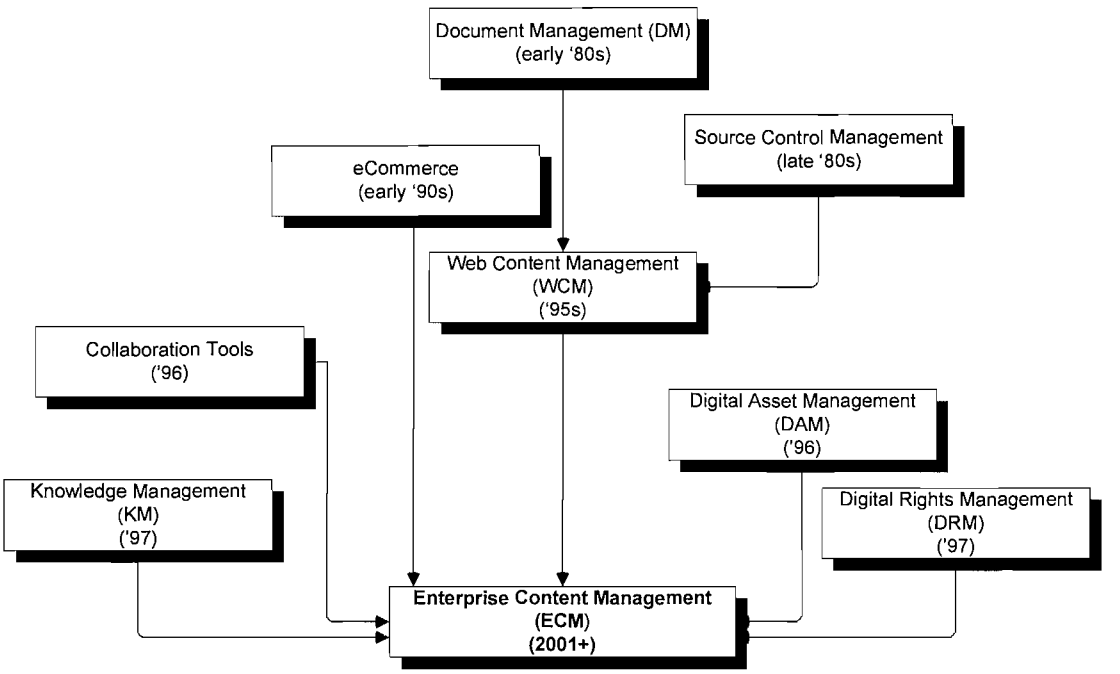


Figure 2-1: Enterprise Content Management related technologies, after (McGrath 2003)

We briefly describe the technologies mentioned in figure 2-1 here.

While document management is focused on 'traditional documents', the focal point of Web content management is on creation, management and maintenance of content on Websites. Also, document management systems usually manage files created by other applications, while Web content management systems control the creation of the content in addition to publication to various channels.

Source control systems manage files that represent source code or related documents. They keep track of the file versions and typically facilitate concurrency control by allowing only one person to edit a particular file at a time. *ECommerce* systems provide management of Websites focused on transaction-based systems.

Collaboration tools support teamwork. Typical functionality covered by collaboration tools, according to (McGrath 2003), includes:

- Enabling a group of people to work on a related content at the same time;

- Providing information exchange, preferably in real time, i.e. chat rooms, discussion threads, virtual meetings, application and desktop sharing white boarding;
- Utilising shared workspaces to support project teams;
- Providing a virtual workspace for peer-to-peer knowledge exchange for geographically dispersed teams.

Collaboration tools mostly include a *workflow* as the content or documents must pass through a defined set of steps during their lifecycle.

Document management systems typically lack abilities to efficiently capture and catalogue complex digital assets such as graphic file, design layout, video and streaming audio. These functions are provided by *Digital Asset Management* (*DAM*) systems. The DAM systems deliver the ability to store, register, index, analyse and retrieve multimedia content. *Digital Rights Management* (*DRM*) systems are similar to the DAM systems, with the additional role of regulating use of digital assets and ensuring that content intellectual property remains safe.

There are a number of intersection points between the mentioned enterprise content management technologies, while their boundaries are becoming increasingly blurred. Overall, ECM represents a holistic strategy for managing all types of information within an organisation, at all stages in its lifecycle, from creation and capture through to archiving and disposal (Jennings 2004). In other words, ECM manages a broad spectrum of electronic assets that are amassed across the enterprise. It is about providing tools to manage the creation, storage, editing and publication of information in a collaborative environment (McGrath 2003).

An overview of some prominent commercial ECM systems is given in Appendix A.

2.3 Hypertext

Usually, when we think about a document, we see it as a medium for sequential representation of information such as a text, laid down in a linear order. It does not have to be that way.

"Hypertext is a text that is not constrained to be linear and contains links to other

texts", as defined in (Girschweiler 1992). The term hypertext was coined by Ted Nelson in the 60s. He defined hypertext as "a body of written or pictorial material interconnected in a complex way that it could not be conveniently represented on paper" (Nelson 1965; 1981) or "a combination of natural languages text with the computer's capacity for branching, or dynamic display" (Nelson 1967). A term, closely related to hypertext, is hypermedia, also first used by Nelson (Nelson 1965). Hypermedia is a term used for hypertext that is not constrained to be text only; for example, it can include graphics, video, animation and sound. It can be defined in the following way: "Hypermedia: An application which uses associative relationships among information contained within multiple media for the purpose of facilitating access to, and manipulation of, the information encapsulated by the data" (Lowe & Hall 1999). "Multimedia hypertext", "Hypermedia" and "Hypertext" tend to be used loosely in place of each other (Nielsen 1993).

A hypertext system is made of nodes (concepts) and links (relationships) among them. A node usually represents a single concept or idea. The term *navigation* in hypertext and hypermedia systems refers to a way and order in which a user moves among documents or parts of documents that are of interest to be viewed. Links enable navigation between connected pieces of text or media. A fundamental hypertext characteristics, the link, possesses a source and a destination and it represents a relation of some sort between them (Ashman 2000).

2.4 Associative Links

An *associative link* can be defined as a type of link where two nodes are connected because there exists an association between them. The original idea behind hypertext and hypermedia applications was to allow navigation around a collection of material in the system using associative linking. The hypermedia application "uses a network of associatively related information" employing associative links, where an associative link represents "an instantiation of a semantic relationship between information elements" (Lowe & Hall 1999).

It can be considered that the inspiration for hypertext and associative linking comes from the Vannevar Bush's ground-breaking article "As we may think" (Bush 1945).

Bush's ideas of a device he called the memex[7] played a key role in a modern hypertext and multimedia development (Bush 1967). On the basis of observation that the human mind associates memories in accordance with some intricate web of trails, Bush introduced the concept of linking information stored in the form of documents, such as all individual's records, books and communication. The memex device represents a kind of mechanized private library which supports associative indexing and navigation, a sort of an information retrieval machine based on a microfilm technology. It was supposed to aid the user in associative recall of information by allowing initial creation and later browsing of associative links. Bush's idea was that the user links documents and joins them together to form a named trail for inclusion into the memex machine. His vision included the idea of trail branching as well. Bush anticipated the appearance of new kinds of encyclopaedias, with a ready-made trail network and even new vocation of "trail blazers".

Creating paths of associations through various documents envisaged by memex can be regarded as a knowledge building activity. The memex concept can be considered as not only "a repository of linkable resources, but a space for creating associative paths through information, and through this association, developing knowledge and understanding" (schraefel *et al.* 2004).

2.5 The Pivotal Hypertext Research

One of the first pioneering works in the area of hypertext systems in the early sixties, is Douglas Engelbart's ON Line (*NLS*) system (Engelbart 1963). NLS used links for cooperative linking and is considered to be the world's first implementation of what was to be called hypertext.

The memex vision strongly influenced Ted Nelson's project Xanadu[8] (Nelson 1981; Whitehead 1996). The aim of the Xanadu project was to create a unified literary environment on a global scale, a repository for everything that anybody has ever written, stored in computers. Documents are published into the *docuverse* in Xanadu, in much the same way that books or magazines are published. Documents can be

---

[7] The term "memex" could be an acronym for a "memory extender", although in the time it was coined this was not specifically indicated.

[8] Xanadu Home page, Available from World Wide Web: <http://www.xanadu.net/>

accessed by users through a distributed network of archival servers. When new documents are posted in the system, they can refer to and be connected to the already published documents. Xanadu comprised the concept of bi-directional links and a complex versioning system. Although never completed (Wolf 1995) Xanadu has influenced many subsequent systems, including the Web.

One of the best-known projects comprising the most successful hypertext system on the Internet is the World Wide Web, also known as WWW or simply the Web. The Web began as a networked information project at CERN in early 90's (Berners-Lee 2004). Documents in the Web are written in the HTML format, stored on distributed servers and addressed by URLs (Raggett *et al.* 1997). Web browsers then fetch the appropriate file(s) from those servers and interpret the format. HTML format consists of elements or tags which determine both the content and formatting of the Web document. The majority of the hypertext systems advocate separation of links from the content. However, the HTML format allows embedding the links in a form of the destination URL inside the content of the Web document itself, a feature known as *tagging*. Usually, Web pages are related to an entity, such as company, product or an individual, and managed as a group of connected pages called Website. However, a page in one Website can point to any other page on the Web. This can lead to an issue known as a broken or dangling link: a situation where a document is removed while other documents still point to it.

Closely related to tagging is the concept of semantic annotation. Annotations are, according to WordNet and Merriam-Webster, comments, usually added to the text. The markup languages such as HTML and XML can be considered as schemata for embedded annotations. Semantic annotations can be considered as information about the entities or concepts appearing in a part of document or its region. Semantic annotations can serve two purposes: to assist link services in providing hypertext links, or as a simple librarian for looking up resources about the specified entity or concept (Bechofer *et al.* 2003).

It is important for this thesis to note that the HTML format comprises an element called <META>, not fully utilised at the moment. Its original aim was to allow a document to be tagged with a comma-delimited list of keywords. The META element was meant to support the automated discovery of resources by search

engines. However, in practice, its abuse led to the situation where it is currently largely ignored by both Website designers and search engines (Himmelstein 2005; Doyle 2005).

It can be argued that Xanadu envisaged a richer, more robust and usable version of hypertext than today's Web (Yeates 1999). Although the Web suffers some inherent drawbacks such as dangling links, there is some kind of *docuverse* on the Internet, even though not in a way imagined by Nelson.

It is important to mention here "Microcosm", a system that manages hypermedia links in document management systems (Fountain *et al.* 1990; Davis *et al.* 1992). Its novel contribution lies in the introduction of navigation based on the content. Microcosm was initially developed in the 90s' at Southampton University as a distributed open hypermedia environment. It provides links generated on the fly and inserted in the existing applications. Microcosm uses generic links that link from a word or a phrase rather than from a specific source location. Links are deduced based on explicit or authored links stored in a database. The system can employ multiple databases of links called linkbases, which are user-configurable, for providing different paths through a set of multimedia information. Microcosm defines the three following types of links:

- The specific link (from a specific point in a source pointing to a specific point in a destination document)
- The local link (from a particular object, such as a word, anywhere in a source to a particular object anywhere in a destination)
- The generic or glossary link (connects particular objects in any place in any document)

In addition, there are two more types of links where destinations are not static:

- Text retrieval links (generated dynamically by computation using various types of text matching techniques)
- Relevance links (creating links to other documents that were clustered in advance by similarity)

Another Southampton University project, the Distributed Link Service (*DLS*) (Carr *et al.* 1995; 1996a; 1996b) builds on Microcosm experiences and provides link services for the Web. DLS redefines the term 'link' to be a specification of

relationship between two items – the source and the destination document. This allows the source of the link to expand into several offsets within a group of documents, or the destination of the link to resolve to a number of alternative documents. The aim of DLS was to bring the mentioned link related ideas to the Web, using experiences from the Microcosm project. It makes use of an external link database for storing and managing links thus abstracting the link service component as a third party service (De Roure *et al.* 1996). The set of applicable links for the document is obtained by sending a request to the link server(s) and the result is shown as a list of hyperlinks – destinations in a DLS HTML page.

The latest development in the area of hypertext includes the Web's successor – the Semantic Web. Initially the Web was created to enhance sharing of information between humans. The next generation of the Web, usually called the Semantic Web[9], originated from a vision of the Web's creator, Tim Berners-Lee (Berners-Lee *et al.* 1999; 2001; Heflin & Hendler 2001). Its goal is to establish the Web as a knowledge-rich information environment and to make it machine processable. Berners-Lee defines it as "an extension of the current Web in which information is given well-defined meaning, enabling computers and people to work in better cooperation" (Berners-Lee & Miller 2002).

2.6 Metadata

Document management systems typically store a number of documents' properties together with documents. Those document properties are known as *attributes* or *metadata*. The term *meta* comes from the Greek word that denotes *alongside*, *with*, *after*, *next*; while more recent Latin and English usage would employ *meta* to denote something transcendental, or beyond nature (Hillmann 2005).

"Metadata is structured information that describes, explains, locates or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information", as stated in (NISO 2004).

We can now redefine documents as "a discrete unit of content and its associated metadata" (Microsoft 2006a).

---

[9] Semantic Web Home page, Available from World Wide Web: <http://www.w3.org/2001/sw/>.

The term metadata is used differently in different communities, but in document management systems it is used in exactly the same way as in traditional library cataloguing – mostly for classification and taxonomies of the underlying documents. A popular example of metadata use is a library catalogue, which helps librarians manage their books and journals (Steinacker *et al.* 2001). According to (NISO 2004), metadata can be divided into three general types:

- **Descriptive metadata** provide means of identifying and discovering a resource, such as title, author or keyword. They describe when, how and by whom a particular piece of data was collected and what is it for (McGrath 2003). The descriptive metadata can also serve to summarise the meaning of the data.
- **Structural metadata** describe compound resources i.e. resources composed of several objects, for example scanned pages of a chapter. Also they can indicate relationships with other resources.
- **Administrative metadata** help manage a resource and contain technical information such as information about the file type/format. There are two subtypes of administrative metadata: rights management (intellectual property rights) and preservation metadata (archiving information).

Descriptive metadata are mostly used in locating resources they describe. They allow the description of the content to be shifted from the content-matching (or string-matching in most cases) level to a conceptual level, where the objective of searching can be semantically described (Steinacker *et al.* 2001).

From the location perspective, metadata can be either embedded in an object, i.e. as a part of the electronic resource content, or stored separately in some kind of a repository. The process of attaching metadata to content in order to categorise it is referred to as *meta tagging*.

2.6.1 Metadata Schemas

Metadata are usually organised into sets of metadata elements called *metadata schemas*. A metadata schema defines names of elements, their meaning (semantics), their syntax and content rules. Schema defines, for example, what types of values are given to an actual instance of a metadata element, or what values are allowed (coming from a controlled vocabulary).

There are numerous metadata schemas at the moment, but most of them for metadata representation use *Standard Generalized Mark-up Language (SGML)*[10] or its subset *Extensible Mark-up Language (XML)*[11] (Birbeck *et al.* 2001). XML was designed by the World Wide Web Consortium, (W3C[12]) as an initiative to define interchange of structured data on the Web.

Probably the most common schema used for Web originated resources is Dublin Core[13], established in 1995. Its aim is to provide semantic building blocks of Web metadata. Dublin Core contains 16 simple schema elements, which are best illustrated by the example given in figure 2-2:

ElementName	**Content**
Title	"Mirjana Andric's Home Page"
Creator	"Mirjana Andric"
Subject	"Personal home page"
Description	"Postgraduate student's presentation. Presents a list of research interests and publications."
Coverage	"1997-2005"
Publisher	"University of Southampton Website"
Date	"Jun-2005"
Type	"Web Page"
Format	"text/HTML"
Identifier	"http://www.ecs.soton.ac.uk/~ma00r/"
Language	"en"
Rights	"Unlimited access"

Figure 2-2: Dublin Core example

Dublin Core entry coded within HTML has the following syntax:

```
<META NAME = "DC.ElementName" CONTENT = "Value">
```

All Dublin Core elements can be repeated; they are optional and may be represented in any order. For example, a document can have more that one *Author* and the *Description* field can be omitted. Dublin Core elements fall into three categories, according to (Weibel *et al.* 1998):

---

[10] Overview of SGML Resources, Available from World Wide Web: <http://www.w3.org/MarkUp/SGML/>.

[11] Extensible Markup Language XML, World Wide Web Consortium, Available from World Wide Web: <http://www.w3.org/XML/>.

[12] <http://www.w3.org>

[13] Dublin Core metadata Initiative Home page, Available from World Wide Web: <http://dublincore.org>

- **Content,** comprising Coverage, Description, Type, Relation, Source, Subject, Title, Audience
- **Intellectual Property,** comprising Contributor, Creator, Publisher, Rights
- **Instantiation,** comprising Date, Format, Identifier, Language

Other examples of metadata schemas are:

- The Text Encoding Initiative (*TEI*) for marking-up novels, plays and poetry in order to support research in humanities[14].
- Metadata Encoding and Transmission Standard (*METS*) for describing structural aspects of scanned materials[15].
- Metadata Object Description Schema (*MODS*)[16], a subset of MARC 21[17] Library of Congress bibliography schema.
- The Learning Object Metadata (*LOM*), the IEEE standard for describing learning resources[18]. The LOM Schema uses almost every category of the Dublin core and extends it with suitable categories and attributes.
- MPEG Multimedia metadata for describing multimedia collections such as music and graphics[19].

2.6.2 The Role of Metadata in Document Management Systems

The development of digital libraries that can, in a broader sense, be considered as document management systems, has initiated much work in the area of metadata extraction and usage. Metadata in document management systems can be very basic. For example, the simplest way to manage documents is via a file management system and in this case at least the file name and the date of creation are kept. Usually, in document management systems, there is a kind of metadata repository where metadata are stored. Metadata associated with documents or in some case the

---

[14] The Text Encoding Initiative Home page, Available from World Wide Web: <http://www.tei-c.org>.

[15] Metadata Encoding and Transmission Standard Official Website, Available from World Wide Web: <http://www.loc.gov/standards/mets/>.

[16] Metadata Object Description Schema, Official Website, Available from World Wide Web: <http://www.loc.gov/standards/mods/>.

[17] Machine-Readable Cataloguing, Available from World Wide Web: <http://www.loc.gov/marc>.

[18] WG12: Learning Object Metadata Website, Available from World Wide Web: <http:/ltsc.ieee.org/wg12>.

[19] Moving Pictures Experts Group WebSite, Available from World Wide Web: <http://www.chiariglione.org/mpeg>.

documents themselves are stored and managed in repositories. Usually, documents are left in their original formats in the file system while metadata in repositories describes and points to them. A repository itself is typically implemented in either (McGrath 2003):

- Relational database (using tables);
- A pure XML database where even document content is stored uniformly with metadata in the XML format. XML databases usually provide better performance;
- A hybrid approach where relational database provides outside the wrapper of XML.

The use of database based repositories allows metadata to play a major role in a range of resource discovery functions, as listed in (NISO 2004):

- Identify documents
- Find documents by a relevant criteria
- Bring similar documents together and distinguish dissimilar ones
- Provide location and access information to the documents once they are found

There are initiatives to provide a common framework to standardize metadata repository exposing and extracting, such as OAI protocol for metadata harvesting (*OAI-PMH*) (Lagoze *et. al* 2002).

The document management system itself usually provides means to create metadata together with creation or managing documents, by using the following methods, according to (NISO 2004):

- Templates that allow users to fill in the values of metadata into pre-set forms that correspond to the schema elements used in a system. Then, a final metadata representation is generated based on the template;
- Mark-up tools that support embedded metadata creation;
- Extraction tools that automatically generate metadata based on the document analysis (usually limited to textual resources) and in that way support the process of manual determining metadata for documents;

- Conversion tools that translate the existing metadata to a final format. This type of tools can serve as an aid in completing the metadata;
- Combination tools, using two or more tools in different stages of metadata creation.

Besides resource discovery, metadata is important for organising electronic resources, promoting interoperability among different systems as well as both humans and machines understanding, digital identification, archiving and preservation. The resource discovery function in the context of document management systems is explained further in the following section dedicated to the information retrieval topic.

2.7 Information Retrieval

Information Retrieval (IR) is a field concerned with the structure, analysis, organisation, storage, searching, and retrieval of information (Salton 1989). It deals with how people find information and how tools can be constructed to help in that search. Since the advent of the Web, these tools have become known as search engines. Searching is the act of trying to find something or someone. IR systems identify and select a subset of large collection of information according to some criteria, known as a *query*. One can distinguish between two forms of search, for an item that is known to exist, with the intent to locate it, and for an item whose existence is uncertain, in order to ascertain whether it exists or not.

The issues concerning retrieval, according to Foraker Design[20] are:

- how the organisation (storage, design, relationships) of information affects its retrieval
- the types of searches people make
- the kinds of search queries people can make effectively
- what determines the relevance of retrieved information

A term related to IR is *parametric search.* According to (Aho *et al.* 1974) a

[20] Usability resources: a glossary of usability-related terms, usability methods, best practices, A free service of Foraker Design, Available from World Wide Web: <http://www.usabilityfirst.com>.

parametric search is a search that fits a number of simultaneous criteria (the parameters of the search). Sometimes, the term *attribute searching* is used instead. The information repository is searched by specifying values for attributes. The search mechanism returns the items that have the specified values on the attribute (Brown 1988). This method is mostly adopted by database engines.

On the other hand, the techniques of content searching are mostly applied by the Web search engines. In this case arbitrary search terms are submitted and the search mechanism returns the items where the requested terms occur. The mechanism is usually based on heuristics such as the frequency of the occurrence, and is only applicable to the text content.

The characteristic of an IR system is usually quantified using two measures, two quintessential IR metrics (Cleverdon & Kean 1968; King 1995):

- Precision, i.e. how well the retrieved documents match the search query, what percentage of documents are relevant to the query
- Recall, i.e. what fraction of the relevant documents are retrieved by a query

Recall measures the percentage of relevant texts that are correctly classified as relevant. Precision, on the other hand, measures the percentage of classified texts that are correctly relevant (Oxnard & Evans 2003). It would be the best if the IR system combined a high recall rate with a high precision rate. However, those two measures are generally inversely proportional – high precision is usually obtained at the expense of a relatively low recall. Precision and recall are usually measured by using relevance feedback, the process of tagging search results as relevant or non-relevant, conducted by the result recipient.

One classification of types of search is given by (Golshani & Dimitrova 1994):

- by key or identifier;
- by condition (given in the form of Boolean statement);
- finding similarities and appearance or absence of some pattern;
- by searching a content or semantics.

There are some additional kinds of searches as defined in (Puchinger & Raidl 2005;

Winston 1993; Russel & Norvig 1995). A search can be 'exact' where some algorithm is precisely followed; in contrast, a search can be heuristic. *Heuristic* as a term is always associated with the 'art of good guessing'. Therefore, heuristic search uses assumptions and 'shortcuts' that allows the search to be performed more efficiently; however the results are not always guaranteed. The term heuristic search is often used to describe domain knowledge-based search methods (Stefik 1995).

In order to perform searching an IR system has to collect the material to be searched, index it in order to prepare for the easier searching, and finally provide an interface/engine for querying.

Search engines are essentially text IR systems that index a vast amount of text on the Web. The architecture of a typical search engine on a Web comprises the following components (Pokorny 2004):

- *Crawler* (also called a spider or robot): a component that recursively fetches the Web pages by traversing the links, and then stores them in the Page repository;
- *Indexer*: a component that analyses the collected pages and builds necessary structures for holding terms and links in order to enable fast access;
- *Query engine*: a component that processes user queries, matches them to documents using the built indexes and subsequently ranks the result.

From the point of view of a document management system, the following search features are desirable:

- Search functionality that allows the standard parametric search by using metadata and Boolean operators such as AND or OR;
- More advanced intelligent searching such as natural language search, search by taxonomy concepts, ability to refine queries, associated word search, highlighting search terms in results and ranking results by relevancy.

It is interesting to note that the top ten research issues, identified a decade ago, still present today's challenges (Croft 1995):

1. Integrated Solutions: IR solutions integration with other systems within an organisation, thus aiming to solve part of the information management

problems;

2. Distributed IR: demand for the IR systems to work in distributed, wide-area network environments;

3. Efficient, Flexible Indexing and Retrieval: improving query response time and indexing speed;

4. "Magic": automatic expansion of queries to cater for the vocabulary mismatch (including synonyms or usage of automatic thesauri[21]);

5. Interfaces and Browsing: design of easy to use and intuitive interfaces for displaying and browsing search results;

6. Routing and Filtering: identifying relevant components in streams of information such as news feeds;

7. Effective Retrieval: Improving the IR measure and recovering from query imprecision and mistakes (using techniques such as stemming);

8. Multimedia Retrieval: accessing image, video and sound media and the multimedia indexing;

9. Information Extraction: identification of entities such as organisation and peoples' names from the textual IR results;

10. Relevance Feedback: case where user identify relevant items and the query is repeated based on those;

Surveys of more recent top research issues mention enhancing personalised results, improved multimedia searching, interactive question-and-answer methods, better visualisation techniques and searching on the non-PC devices (McLaughlin 2004).

IR related standards cover for example the Z39.50 search protocol (Kelly 1998), SQL-Multimedia proposals (SQL/MM[22]) and the DARPA TIPSTER architecture for

---

[21] See section 4.3.1 for more about thesauri.

[22] SQL Multimedia SQL/MM Home page, Available from World Wide Web: <http://www.jcc.com/SQLPages/SQL%20Multimedia.htm>.

integrating retrieval, routing and extraction systems[23].

Search can be *personalised*, when search results are tailored to the user and his/her preferences. The general idea is that different users will get different search results for the same query, depending on their profile saved in the system. A user profile can be built either explicitly or implicitly. Explicit preferences are based on information directly obtained from the user, for example asking them to rate suggested items, while the implicit ones are deducted from the user behaviour collected over time using unobtrusive monitoring.

2.8 Navigation and Retrieval

Navigation, as mentioned earlier, can be defined as a process of following a link or association between two pieces of information. The terms navigation and retrieval are closely related. Information retrieval basically represents bypassing the existing link structure and directly jumping somewhere in a specified information space. Searching the Web can be defined as a process of requesting a set of documents that match the query from a search engine. The documents are then navigated using the artificially created results page with links on it (Carr 2000c). It can be argued that navigation is equivalent to retrieval in a sense that retrieval can dynamically discover similar items and create links between them on the fly. However, Lewis and others (Lewis *et al.* 1999) point out that links in a hypertext can also be created through the external knowledge of the link's author instead out of similarity.

2.9 Document Management in an Organisational Setting

Like information systems in general, document management systems are influenced by hypermedia and especially Web technologies. Publishing documents either on organisational intranets or on the Internet itself, has become a reality nowadays. Bringing the knowledge from these documents to the people who will be accessing them hence becomes possible and highly advantageous.

Yet, the activities that involve finding, retrieving and presenting the relevant documents to their potential readers and editors, still present a challenge. In the case

[23] TIPSTER Text Program A multi-agency, multi-contractor program, Available from World Wide Web: <http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/>

of both intranets and the Internet, documents are frequently published in document management systems[24] without sufficient meta information. This is like putting pictures in a photo album without noting when or where they were taken (Maurer 1998). The problem is due to a lack of standardised means for metadata creation as well as a lack of discipline by the creators. <META> elements are rarely used, as discussed in section 2.5, and can not be sufficiently relied on. On top of this, content in a typical organisation contains much larger fraction of non-HTML documents than the general Web.

What are the typical document searches in the organisational environment? We can start considering this issue by first looking into the Web. Characteristic Web search types can be categorised as follows (Rose & Levinson 2004):

- Navigational: finding a specific known Website in a more convenient way than typing the URL;
- Informational: learning something by reading or viewing Web pages;
- Resource oriented: Obtain a specific resource from the Web, activities such as downloading, being entertained or interacting with it;

As concluded by Rose and Levinson, navigational searches are less prevalent in a general Web, while the resource-seeking goals may account for a large fraction of searches. It can be argued that searches in an organisation mimics the Web to a great degree – most of the time people are looking for a specific piece of content, such as a rulebook to view, a form to download, or the latest version of a document to edit.

Document management systems in organisations are characterised by the fact that they collect a highly-focused set of materials. Documents inside such system have been created or pre-selected and deposited there because they are relevant for the activities shared inside an organisation. As noticed in the works of Fagin and others (Fagin *et al.* 2003): intranet documents are often created for simple dissemination of information and are mostly spam free and search-engine unfriendly. Also a large percentage of queries tend to have a small set of correct answers.

---

[24] or content management systems in general

Another issue with document management systems in an organisational setting is that quite frequently documents in a document management system or on a corporate intranet, are not hyperlinked. The reasons for this are various, and include the following:

- Link authoring is mostly a manual and error-prone process that users avoid unless absolutely necessary;
- Links can easily become broken if documents are frequently changed (one of the common issues on the Web);
- Most of the current tools for creating documents do not offer natural and easy ways to create links.

Consequently, the presence of links in document management systems is frequently not sufficient and can not be reliably utilised for the purpose of document retrieval.

Yet another challenge is the issue of supporting groups of people in the same organisation working on shared document collections. The fact that users are frequently working on the same documents and employing similar searches provides an opportunity for sharing member experiences within a group.

2.10 Summary

The objective of this chapter was to introduce the field of document management and to explore the main challenges presented by it. It also defined hypertext and covered the usage of metadata in document management systems. The topic of information retrieval was also covered, which is particularly relevant for this work as it focuses on the search aspect of document management systems. Some observations on particularities of document management systems in an organisation setting, such as lack of linking and the notion of collaborative work, were pointed out, which led us to the area of collaborative recommender systems.

Chapter 3

Recommender Systems

This chapter provides an introduction to recommender systems. It aims to show the importance of recommender systems, which unlike pure document management systems, take users more into account. The chapter considers the key issues in this field, such as document analysing, establishing similarity, user profiling and initialisation problems. This is followed by a survey of some important recommender systems.

3.1 Introduction

The recent decade has seen development of some solutions to the problems of information sharing and collaboration. Recommender systems are the most widespread systems used to facilitate group work. They can be thought of as members of the Computer Supported Cooperative Work (CSCW) (Greif 1988) family of systems, and ultimate successors of the Engelbart's work in augmentation of human intellect by computers (Engelbart 1963; 1995), as argued in (Twidale & Nichols 1998).

In the real world recommendation can be seen as inherently personal. Recommender systems attempt to emulate the human recommendation practices. Recommender systems assist users by providing appropriate suggestions, usually in the form of a proposed piece of information such as a document or a reference to it. In a typical recommender system people provide recommendations as input, which the system then aggregates and directs to appropriate recipients depending on their preferences (Resnick *et al.* 1994; Resnick & Varian 1997). The developers of the first recommender system, Tapestry (Goldberg *et al.* 1992), coined the phrase *collaborative filtering* and several others have adopted it. However, recipients of recommendations may not collaborate with the recommendation originators, as later recommenders systems have demonstrated, so *recommender system* may be a better term.

Recommender systems can be considered as a result of three pronounced shifts in information retrieval systems research. Table 3-1 depicts the flow of that evolution (Perugini & Goncalves 2002).

Table 3-1: Recommender systems evolution, after (Perugini & Goncalves 2002)

System	Design Matrix
Information Retrieval	terms × documents
Information Filtering	features × documents
Collaborative Filtering	people × documents
Recommender Systems	people × artifacts

The original objective of Information Retrieval systems is to select relevant information, while information filtering deals with the continuous removal of irrelevant information. IR systems use terms to distinguish relevant documents while information filtering systems use additional document characteristics (features) to accomplish the filtering task. Therefore IR systems look into matching terms and documents while information filtering matches features to documents. In the early days of systems like Tapestry, collaborative filtering meant using people's experiences in the filtering process, which means that the system design shifts from features to people: people get matched with documents. Finally, contemporary recommender systems deal with various items or artifacts that can be recommended to their users, therefore ultimately people get matched with artifacts.

3.2 Recommender System Types

The main categorisation of recommender systems is based on the techniques used for recommending. Recommender systems can utilise:

- **Content-Based** approach, where the contents of an item is analysed and recommendations created mainly based on items similar by content;
- **Collaborative** approach, where the pattern of other users' behaviours is analysed and recommendations formed mainly based on similarity of users;
- Advanced, **hybrid** approaches that combine various methods.

3.2.1 Content-based Recommender Systems

Content-based (CB) recommender systems (Balabanovic & Shoham 1997; Mladenic 1999) typically analyse the content of textual items from a given collection. This class of systems then creates a summary representation of each item analysed using frequent terms analysis techniques in case of textual content or other methods for other types of content. Users of the system are represented by their profiles which

correspond to an approximation of the interests of a given user. Users are modelled in a manner similar to the way documents are modelled. This allows the content-based system to calculate the similarity between users and also user-document and document-document similarities. Based on these similarities, items from an analysed set can be suggested to users. Statistical text analysis and similarity calculation methods will be further detailed later on in this chapter.

3.2.2 Collaborative Filters

The other kind of recommender systems, collaborative filters (*CF*), introduced in (Goldberg *et al.* 1992), and described in seminal work of Resnick & Varian (1997)[25], utilise the natural recommendation process, e.g. accepting an advice from a friend who has similar preferences. Collaborative filtering systems have human recommenders who explicitly rate items, such as Web pages, and then the system prepares recommendations for the other users based on the overlapping areas of interest. User models, containing the user's behavioural patterns, are stored. A collaborative filtering recommender system is generally based on recording a person-specific set of choices/preferences or the usage history. This is used as a behaviour-based profile to compare with other user profiles in the appropriate context. Collaborative filtering recommender systems usually perform the initial user profiling by asking a new user to rate some preliminary items. Recommendations are then created by using statistics over the most similar profiles in order to predict individual needs and preferences.

The typical algorithm for generating a recommendation is given in (Herlocker *et al.* 1999):

Step 1: Calculate a degree of similarity between the current user and other users

Step 2: Identify a group of users who appear to share common interests with the current user. Their evaluations are to be used for generating recommendations.

Step 3: Calculate estimated evaluations for items that the current user has not seen (or evaluated). An estimated evaluation predicts the current user's

---

[25] Additional resources can be found on <http://www.iota.org/Winter99/recommend.html>.

evaluation of an unseen item.

Step 4: Rank order the items according to the estimated evaluations and select the top *n* items to recommend.

3.2.3 Hybrid Recommender Systems

There are some hybrid approaches combining the collaborative and content-based methods as in (Balabanovic & Shoham 1997) and a technique used in (Baudisch 1999) where content based descriptors are used as additional users for collaborative filtering. A content-based predictor for user preferences, described in (Melville *et al.* 2001), is another example of a hybrid system. Hybrid recommender systems using knowledge-based approach are going to be described in section 4.2.4.

3.2.4 Other Categorisations

Recommender systems can also be categorised and observed in the following four dimensions (Resnick & Varian 1997):

- **By the complexity of the given evaluation,** i.e. the user evaluation or rating can be anything from a single bit, item recommended or not, to unstructured textual annotations;
- **By the system of gathering recommendations** i.e. explicitly or implicitly;
- **By the origin of recommendation** i.e. anonymous, tagged with the source's identity, or tagged with a pseudonym;
- **By the method of aggregating evaluations,** for example: variants of weighted voting, content analysis, combining suggested links into referral chains, filtering out negative recommendations etc.

The other way of categorising recommender systems would be according to the way recommendations are made:

- By recommending what similar people would prefer;
- By recommending documents/items sharing as many attributes with an item the user identifies desirable;
- By the individuals' patterns.

In addition to content-based and collaborative approaches, two more recommender systems types are mentioned in the literature (Terveen & Hill 2001): recommendation support tools and social data mining systems for mining and visualising records of social activity.

Also, recommender systems can be classified by what they recommend, e.g.: articles from Web newsgroups, Web pages, documents, people or products/items such as videos, movies, music, books or other products in the e-commerce domain.

3.3 Key Architectural Issues

3.3.1 Content-based Issues

One of the key issues in content-based and hybrid recommender systems is analysing the content and determining the similarity of items based on that analysis. In this thesis we are focusing on text analysis. Recommender systems use a variety of content processing techniques based on statistical document analysis and machine learning methodologies (Dhar & Stein 1997; Turban 1995), in a similar way to classical information retrieval systems.

The goal of text analysis in CB recommender systems is indexing documents. *Document indexing* involves transforming full text into a shorter representation. It can be defined as an activity for determining the distinguishing properties of document in order to improve the ability of the retrieval systems to locate the relevant documents while processing a query (Baeza-Yates & Ribeiro-Neto 1999; Evans & Zhai 1996). The indexing process needs to extract or determine the *features* that best distinguish particular documents. It is quite common for indexing methods to assign variable importance to such features, expressed as numerical weights. Those features, also known as *index terms*, are later used in establishing the relevancy of a document: a document is asserted to be relevant if the document and the query share several index terms (Arampatzis *et al.* 1998). If multiple documents are found to be relevant, the degree of the term sharing with a query determines the ranking of the results.

One of the most used models for representing documents, queries and user profiles is known as the *Vector-Space Model* (Salton 1975). It is a way of representing

documents through the words they contain. This is often referred to as the *bag of words* approach to document representation and involves a so-called *linearisation*. Linearisation is a process of ignoring markup tags from a document so its content is reinterpreted as a string of characters (Garcia 2005). Document indexing then continues with the tokenisation (imposing lowercase everywhere, removing punctuation), filtration (removing stopwords) and stemming (reducing terms to their roots). Stopwords is a list containing the most frequent language words such as "and" and "the". Words are then stemmed, as in (Porter 1980), a process in which various grammatical endings are removed and word roots retained. The count of term frequency is kept (tf). The process is illustrated in figure 3-1.

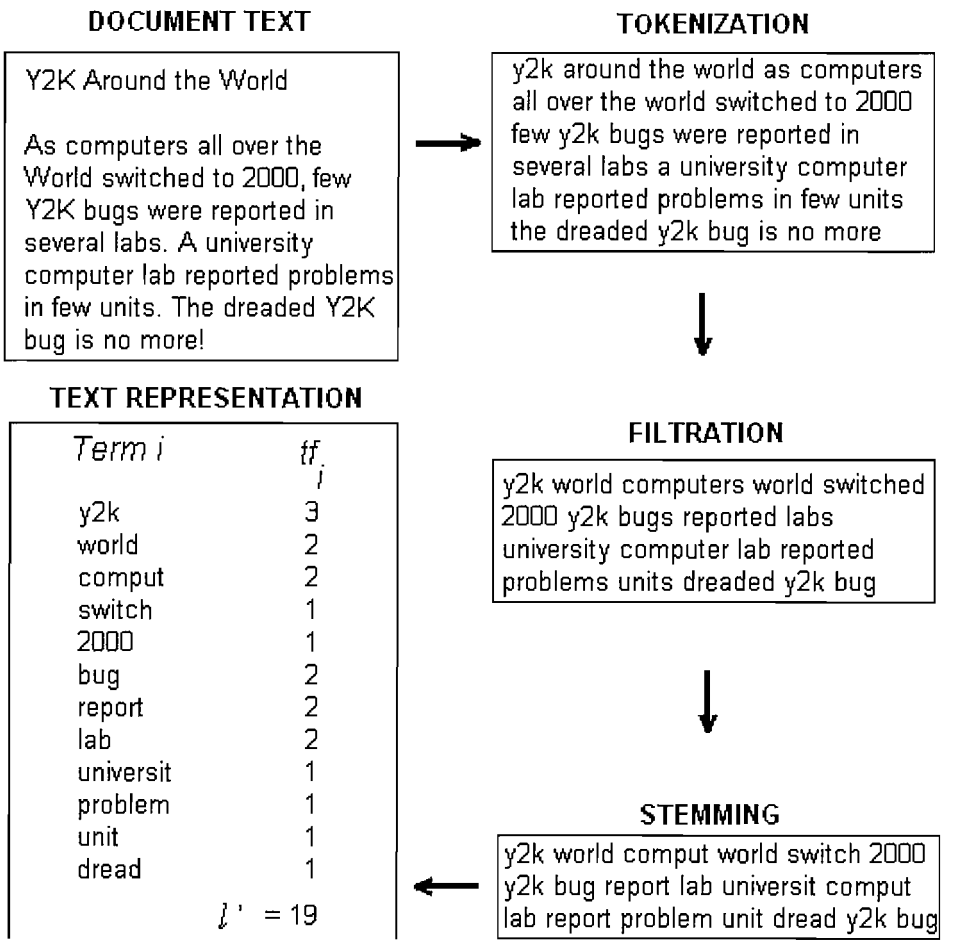


Figure 3-1: Steps of document indexing: tokenization, filtration, and stemming, after (Garcia 2005)

In the vector space model, each document or query is represented as a vector of terms and associated weights. In other words, documents or queries are points (or vectors from the point zero) in the high dimensional space of terms where each term corresponds to an axes and weights are numerical marks on the axes themselves.

This represents a step ahead of so called binary or *Boolean Vector-Space Model* that has only two possible weights: zero when term does not correspond to a document and one if it does. The information retrieval system SMART, a research project initiated in the early 60ties by Salton, described in (Salton 1968; Rocchio 1971), embodied many of the information retrieval techniques found in the modern vector space systems.

One of the most successful and well-established weighting schemes in information retrieval is based on the vector-space model and is called *Term Frequency Inverse Document Frequency* method (TF-IDF) (Salton & Buckley 1988; Salton 1989; Mladenic 1999). In TF-IDF algorithm terms are assigned weights that reflect frequency of term occurrences within a document that are further modified (multiplied) by the inverse document frequency of that term in the selected corpus of documents. This means that favoured terms are those that repeat frequently inside a document but which are not so common in a wider collection of documents. Terms that in general occur too often are thus "punished".

The weights are calculated as in the following equation (Salton 1989):

$$W(t_k, d_i) = (1 + \log(f_k(t_k, d_i))) \cdot \log \frac{n}{r}$$

where $W(t_k, d_i)$ and $f_k(t_k, d_i)$ represent respectively the weight and the frequency of the kth term (t_k) in the ith document (d_k). n is the total number of terms in the whole document collection and r is the number of documents which contain the term t_k. The TF-IDF algorithm usually normalises the vectors as this takes into account different document lengths and makes weights in different documents comparable. Normalised weights are calculated according to the following formula:

$$W'(t_k, d_i) = \frac{W(t_k, d_i)}{\sqrt{\sum_1^n (W(t_k, d_i))^2}}$$

where $W(t_k, d_i)$ represents the original weight, while $W'(t_k, d_i)$ stands for the normalised weight.

Then, when documents are represented as vectors, similarity between pairs of documents or between query and individual documents can be calculated. The vector-space model allows for the simple product of vectors to determine how close they are in the high dimensional space, i.e. how similar they are. The cosine of the angle between two vectors is measured and the obtained result falls between 0 and 1. Zero means that vectors do not share any terms, and one means that all the terms and their weights are the same. The equation that illustrates computing the *similarity factor* between document d and a query q is given below.

$$sim(d,q) = \cos(d,q) = \frac{d \times q}{|d| \cdot |q|} = \frac{\sum_{1}^{n} d_i \cdot q_i}{\sqrt{\sum_{1}^{n} d_i^{\,2} \cdot \sum_{1}^{n} q_i^{\,2}}}$$

This method is quite popular in the IR research community and works well in the majority of situations, mostly on a relatively static document corpus. However, it has some disadvantages. The key problems are coping with the volume of extracted terms and also the requirement that a document collection is present beforehand, which typically is not possible on the Web. Related to this, there are the issues of the time needed to calculate similarity as well as the memory needed for large term-document matrices. There are many techniques for reducing the dimensionality, e.g. feature reduction (Bloom & Langley 1997). The most common is removing apriori terms from a stopword list.

A more recent technique of Latent Semantic Analysis (*LSA*), or Latent Semantic Indexing, (*LSI*) (Foltz & Dumais 1992; Laundauer & Dumais 1997; Papadimitriou *et al.* 1998) is now gaining wider acceptance in the text analysis research community. LSA extends the vector-space model by modelling term-document relationships using an approximation of the typical document-term matrix. Typical document-term matrices are sparsely populated and LSA is focused on making them denser. This is achieved by a well established algorithm involving matrix operations called Singular Value Decomposition (*SVD*) (Furnas *et al.* 1998). Increasing density and reducing dimensionality, apart from solving some scalability issues, has a very interesting consequence: it allows similar terms to be identified. Some documents become related by terms that do not physically occur in the document content. If for example

a document *X* possesses occurrences of a term "cat", the document-term matrix would have a number representing the frequency of the term in the cross section of the term "cat" and the document *X*'s identifier. Let's say that for example some other document *Y* in the same collection mentions the term "feline". If the documents are sufficiently similar, the LSA algorithm would create a corrected document-term matrices in such as way that the document *X* is now related to the term "feline" with the non-null factor in the cross section. This enables future searches on "feline" to find the document *X* where this term does not appear at all. Effectively, a synonymy relationship has been established between those two terms and latent associations between terms and documents discovered. Besides word-to-word similarity, other types of relationships can be mathematically derived: document-to-document similarity and word-to-document similarity. Document-to-document similarity is especially important for the content-based recommender systems as they typically support *Find more like this* or *Get similar documents* features.

Terms can be also related or connected if they appear together in the same document on many occurrences. According to (Baeza-Yates & Ribeiro-Neto 1999), "the degree of term co-ocurrence in a database is a measure of semantic connectivity (*SM*) and can be used to build thesaurus[26]".

Content-based recommender systems are usually criticized for two weaknesses (Shahabi & Chen 2003): content limitation (extracted features capture only certain aspects of the limited range of content) and over-specialisation (suggesting items not strictly similar to the user profile is not possible). These issues can be overcome by using a collaborative approach, although collaborative-based recommender systems have their own issues.

3.3.2 Collaborative Filtering Issues

Collaborative filtering is quite advantageous in cases when the recommended items can not be easily analysed by using text extraction techniques, for example audio or pictures. Generally, recommender systems work best when a large number of people are involved and they have been building their profiles for some time. However

---

[26] See section 4.3.1 for more details about thesauri.

scalability is always an issue with such systems. The key architectural issue here is to select a method of identifying similar users. Also, such systems have the disadvantage that they can not recommend an item that no user has rated and have to design methods of incorporating new items into the recommendation set. As content has not been analysed, an association with a similar item can not be established. Users with unusual or highly specific tastes present another problem. How to identify similar users who have rated non-overlapping items is yet another issue CF systems have to face.

Collaborative based recommender systems also suffer from the initialisation problem, sometimes called the *incentive problem* (Resnick & Varian 1997) and also known as the *cold start problem* (Maltz & Ehrlich 1995; Schein *et al.* 2002). When the system is first used there is no data on which to base recommendations. Gradually by usage the basis for recommendations is built but it can take some time for the system to start providing any recommendations. Cold start is the main problem in recommender systems that use a statistical approach. Introducing explicit, usually static, models of users and items in a knowledge-based approach, is known to overcome the initial cold start problem in the best way.

3.4 Evaluating Recommender Systems

Evaluating recommender systems is very difficult because of the diverse nature of their algorithms and data sets, the goals for which the evaluation is performed and the measures selected for comparison (Herlocker at al. 2004). There are different ways to evaluate different aspects of recommender system (Karypis 2001; Swearingen & Sinha 2001). General Human Computer Interaction (*HCI*) methodology for evaluation can be used (see more details in Chapter 7).

The evaluation process should start with an understanding what tasks the users are trying to perform with the aid of the system. Herlocker and others. (Herlocker *et al.* 2004) identified a list of key user tasks in collaborative recommender systems that should be taken into account during the evaluation. These are:

- Annotation in Context: predicting which items should be viewed in context and thus filtering out the undesired content;
- Find Good Items: suggesting the best items;

- Find All Good Items: exhaustive search, not a usual type of task;
- Recommend Sequence: ordered list of suggestions, a kind of guided tour;
- Just Browsing: where the goal is "sightseeing" of available items;
- Find Credible Recommender: verifying that systems recommendations match users' preferences;
- Improve Profile: when users contribute ratings to make their profile more accurate;
- Express Self: enabling users to express their opinions;
- Help Others: entering ratings for benefiting the user community;
- Influence Others: entering rates for biasing opinions of the user community.

Most recommender systems support the first two tasks and most of the evaluations focus on accuracy. An accuracy metric empirically measures how close a recommender system's predicted ranking of items differs from the user's ranking of preference (Herlocker *et al.* 2004). A standard approach to evaluating the effectiveness of recommender systems is to use traditional IR performance indicators (Salton & McGill 1983): *precision*, being defined as the proportion of recommended items correctly matching the query out of all items offered, and *recall*, being defined as the proportion of correctly matched items offered out of all the "correct" items that exist in the system. This is usually considered as a problem of properly classifying items into "correct" and "incorrect" for the given query (Yang 1999). In the case of recommender systems the query is typically a query by example, i.e. get similar items to the current one, or a query using user profiles, for example find highly recommended items suggested by similar users. In contrast to this, in case of search engines the query is typically a string, i.e. a word or a phrase.

One can argue that relevancy is sometimes an entirely subjective measure. Users can regard some item, e.g. a document, relevant and/or interesting regardless of it matching the expressed search query (Keen 1971). Also, users may not be able to accurately enough specify their information needs, or to formulate it in such a way that it retrieves the desired items (Baeza-Yates & Ribeiro-Neto 1999). All this poses a difficult task of interpreting the underlying meaning of the query for the recommendation engine or search engine. Therefore, the effectiveness of the retrieval algorithm should be best evaluated talking into account the actual number

of retrieved documents, as this indicates the user's implicit interest. Alternatively, explicit user feedback on the relevancy should be employed.

3.5 Examples of Recommender Systems

Tapestry (Goldberg *et al.* 1992), the earliest collaborative-based recommender system in the domain of email filtering, relied on collaboration of a closely-knit group of users. The Tapestry users provide explicit opinions by annotating emails. Similar users are identified and simple filtering rules are set manually.

One of the first collaborative filtering systems applied to usenet news, was GroupLens (Resnick *et al.* 1994). In this system readers rate news and their votes are stored on a separate rating server(s). At the same time, rating servers predict votes of the unseen news for the user, based on the votes of similar users. Recommending is based on the heuristic that people who agreed in the past would agree in the future as well. News items are not totally automatically recommended; they are firstly categorised into newsgroups and secondly moderated by a moderator who can provide the initial ratings as well.

Many recommender systems use *software agents* for performing their tasks. Software agents are software systems capable of flexible autonomous actions having key features such as reactive and/or proactive behaviour and social activity (Wooldridge 1998). Fab (Balabanovic & Shoham 1997; Balabanovic 1997) and People Helping One Another To Find Stuff (PHOAKS) (Terveen *et al.* 1997) are systems that recommend Web resources. Fab uses agents to build profiles for each user and finds the profiles of a set of "nearest neighbours" for that user. It recommends pages by comparing their keywords to those stored with the user profile and the nearest neighbours' profiles. Fab utilises both collaborative and content-based approaches, using TF-IDF techniques for analysing documents and cosine similarity for matching them with the user profile. PHOAKS looks for recommendations implicitly expressed by other people in usenet news and re-uses their recommendations.

IBM's Web Browser Intelligence (Barret *et al.* 1997) personalises Web data using agents to process only those pages previously viewed by the user in order to suggest new ones. However, Web Browser Intelligence is not collaborative: users can search

for topics of interest only in their personal history of viewed pages, and are restricted to following paths through their previously trawled Web pages.

Similarly, WebWatcher (Joachims *et al.* 1997) and Personal Web Watcher (PWW) (Mladenic 1996) provide support for the individual users who are using Web search to find information on the Web. WebWatcher is a browsing assistant helping users on a Web "tour". The user profile is built on a set of keywords specified at the beginning of a tour. PWW builds a user profile with time and uses agents to suggest interesting links on a currently viewed page, based on matching the destination documents with the current dynamic user interests.

A recommendation system Knowledge Pump (Glance *et al.* 1998; Glance & Grasso 2000) uses the community-based collaborative filtering techniques to proactively suggest interesting information to its users. In the recommendation process users' profiles are utilised to derive a flow of knowledge from the repositories.

The OWL system, "A Recommender system for Organization-Wide Learning" (Linton 1998; Linton *et al.* 2000), claims to be the first recommender system that was applied in pursuing the goal of organisational learning, defined in (Cyert & March 1963). The OWL system observes a large number of users for a long period of time. OWL utilises its users as passive sources of knowledge and provides recommendations at the same time for individual users to learn selected items of knowledge. The system monitors usage of applied commands/functions inside observed (specially prepared, instrumented) applications. OWL provides a current peer-based answer to the question: Given all the functionality of this application, which is the next-most useful function for me to learn?

Towle & Quinn (2000) present a system which addresses the problem of bootstrapping in recommender systems. Authors propose that the solution for gaining leverage for knowledge-based recommender systems would be to use explicit models both for users and for items i.e. products being recommended, rather that the implicit ones which are inferred from the ratings given throughout the usage of the system.

Spertus describes a hyperlink-based recommender system called ParaSite (Spertus & Stein 1998a; 1998b). The system mines the existing (manually authored) links on the

Web pages and recommends related documents using the inter-document structure. If a user requests a page similar to a set of pages $\{P_1, \ldots P_n\}$ the ParaSite system will use the reference set $\{R_1, \ldots R_n\}$ of pages that point to a maximal subset of pages P (common "parents") and then return as a result pages that are referenced by the R pages. The system does not understand the content of pages; it only presumes that if a person likes one page then, he/she may find interesting the pages referenced by the same "parent". The main assumption for recommendations is that co-referencing implies some sort of human-defined similarity (Spertus & Stein 1998a), e.g. two pages are similar if they are pointed by the same page. However, as discovered in (Carr *et al.* 2000a; 2000b) most of the hyperlinks between documents on the Web are created for purely navigational purposes, not for true associative linking, which potentially does not carry too much semantic meaning. Although the pointed pages can be of interest, they are probably not very similar in many cases.

Claypool and others discuss the usage of the implicit interest indicators for gathering implicit ratings from users (Claypool *et al.* 2001). They developed a "Curious Browser" that monitored users' behaviour while browsing the Web. The authors found that the time spent on the page, and the amount of scrolling on the page, or the combination of the two, have a strong correlation with the explicit interest. The authors note that gathering implicit interest indicators can be achieved by mining Web server logs, thus strengthening the predictions that can be made by recommender systems.

Probably one of the best known commercial systems for recommending is Amazon[27], the on-line books, music and other products store (Schafer *et al.* 1999; Sarwar *et al.* 2000; Linden *et al.* 2003). Amazon offers both assistance for searching their product database as well as suggestions for the currently viewed items such as *Customers who bought/viewed this book also bought/viewed* and *Look for similar items by category/subject*. A range of numerical (e.g. number of stars) and textual ratings are also provided.

Research conducted in the Intelligence, Agents, Multimedia (IAM) group at the University of Southampton comprises a range of systems aiming to connect the

[27] <http://www.amazon.com>

agents, knowledge management and hypermedia domains.

The MEMOIR (Managing Enterprise-scale Multimedia using an Open Framework for Information Re-use) project (MEMOIR) (Hill *et al.* 1997; De Roure *et al.* 1998a; 1998b; 2001), also found in IT-Innovation's Website[28], is a multifaceted system that can be best described as a recommender system. It utilises user trails (sequences of visited intranet documents) to create a recommended reading list for users who have similar interests. QuIC (El-Beltagy *et al.* 2001) enriches the Web browsing experiences of users by suggesting new contextual multi-destination hyperlinks from a "most similar" linkbase. The Quickstep recommender system (Middleton *et al.* 2001) assists researchers in finding on-line research papers. It proposes daily recommendations using a range of collaborative and content-based technologies. Its successor Foxtrot (Middleton *et al.* 2004) supports a searchable research paper database using a knowledge-based approach.

In the next section, we discuss the MEMOIR project in more detail because it is particularly relevant to the work described in this thesis.

3.5.1 The MEMOIR Project

A sensible approach for mining information on the Web is to exploit knowledge acquired by other users who have previously surfed the Internet or an intranet exploring similar or related material. This is the method undertaken by the MEMOIR project and recommender systems (especially CF recommenders) in general.

The MEMOIR was a joint project, which involved partners from academia and industry, sponsored by the European Union's ESPRIT Programme. The MEMOIR system manages diverse sources of information within an intranet environment and assists users in finding both relevant documents and researchers with related interests. MEMOIR can thus be best described as a collaborative recommender system and navigation assistant, but it can also be characterised as a knowledge management system addressing the so-called corporate memory problem especially visible in geographically dispersed organisations. It also represents the evolution of

[28] Technologies and systems for knowledge management and collaborative working, Available from World Wide Web: <http://www.it-innovation.soton.ac.uk/research/knowledge.shtml>

the Distributed Link Service (DLS) (Carr *et al.* 1995) architecture (presented in section 2.5). But, while the DLS, like many other hypermedia systems, treats hypermedia links as first class objects, MEMOIR promotes another kind of object: *the trail* (De Roure *et al.* 1998a), which we now describe in more detail.

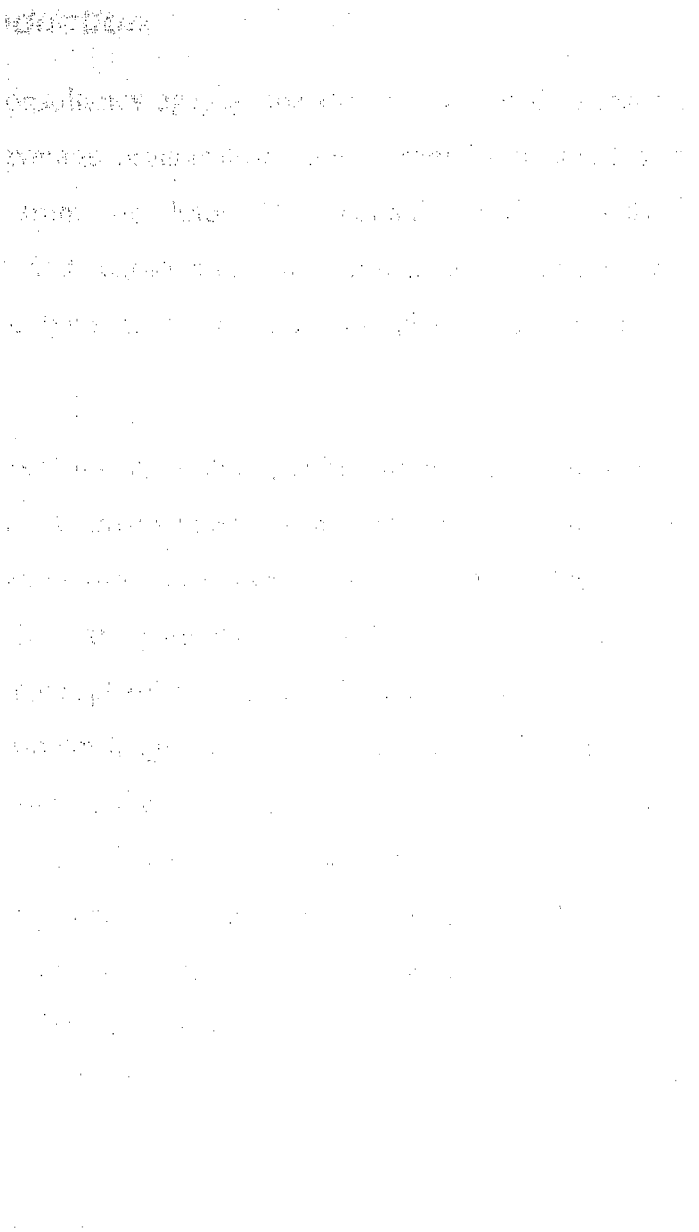
MEMOIR monitors usage of the intranet/Internet and stores locations, in the form of URLs, for the documents (pages) that a specific user has visited: browsed or the links he/she followed. Those sequences of visited sites represent trails in MEMOIR. MEMOIR uses a proxy server for detecting trails and stores them in an object database, thereby creating additional sequential connections for Web pages. This additional linking information is not actually stored within the document, but in an external database, called a trailbase, leaving the original page unaffected. Each trailbase contains collections of links to closely related documents based on a topic of interest to the user who defined the trails. What if another user shares a similar interest? Would they not also be interested in exploring the contents of those trails? MEMOIR could answer questions such as "Who else has seen this document?" and "What other documents did they read?".

MEMOIR comprises an open framework of interchangeable components, among which are: interface manager (user interface enables access to MEMOIR functions via a standard Web browser), databases for links (linkbases) and trails (trailbase) and a set of software agents, such as *Suggest further reading, Extract keywords, Find document by keyword, Suggest relevant people, Find people by keyword* and agent server performing MEMOIR functions.

3.7 Summary

In this chapter we have described recommender systems and reviewed the research literature on the subject. The main recommender systems types, content and collaborative based, and the key architectural issues of text analysis, similarity, user profiling and cold start were introduced. This chapter also discussed some key components for the evaluation of recommender systems, such as the measures and metrics to be used.

The next chapter tackles in more detail the framework needed for advanced knowledge-based recommender systems, including a review of the field of knowledge management.



Chapter 4

Knowledge Management and ZigZag

This chapter provides an overview of the knowledge management area. The aim of the Semantic Web and its use of ontologies as a framework for knowledge management are introduced. Then special attention is given to ZigZag, a novel paradigm for managing information which is going to be adopted for the knowledge manipulation later in this thesis. ZigZag's comparison to ontologies is also discussed.

4.1 Introduction

A renowned consultancy agency, the Butler group, claims that only one third of the value of the average organisation comes from its material assets. The other two thirds come from its knowledge capital (Butler 2004). Therefore, better understanding and exploitation of knowledge management presents a great opportunity for both academic and industrial communities. But what exactly is knowledge?

Choo and others (Choo *et al.* 2000) define *knowledge* as the evolution of information resulting from the involvement of human cognition. It is assumed that human knowledge is expressible in words, able to be constructed using networks of rules, and recognisable in the patterns of the rules (Clancey 1997). Knowledge is closely related to the concept of semantics. The term semantics is used to describe both natural and computer languages. It is often contrasted with syntax[29] and is popularly understood to refer to the meaning of symbols and expressions in languages (Stefik 1995). The term knowledge is often used synonymously with the word information, which is not very precise. According to Davenport (1997) while *data* is considered as a set of discrete facts/simple observations, *information* is defined as data with relevance and purpose. Finally, *knowledge* is regarded by Davenport as valuable information from the human mind. It is our belief that involvement of human interpretation is a distinguishing factor between information and knowledge.

Knowledge management is a discipline dedicated to promoting knowledge growth,

[29] Syntax can be defined as a set of rules or patterns according to which words are combined into sentences (Cherry 2002)

knowledge communication and knowledge preservation in an organisation (Steels 1993). In simpler terms, knowledge management seeks to make the best use of the knowledge that is available to an organisation[30]. A discipline, known as *knowledge engineering* involves acquisition, representation, reasoning (inference) and explanation of knowledge (Turban 1995).

The industrial/corporate community regards knowledge management as more than a technology or a set of methodologies, but truly a practice or discipline that involves people and processes as well as technology (Tobin 2003). Often knowledge practices can be defined as knowledge-powered problem resolution, i.e. using a knowledge base, knowledge sharing, collaboration and knowledge reuse to efficiently resolve business issues.

Closely related to the meaning of knowledge management is the concept of organisational or corporate memory. According to (van Heijst *et al.* 1996) "a corporate memory is an explicit, disembodied, persistent representation of knowledge and information in an organisation".

4.2 Ontologies and the Semantic Web

Capturing, creating, representing and querying knowledge are the main research issues in knowledge management. One of the latest advances in this direction is the development of ontologies and their usage in the Semantic Web.

4.2.1 Definition of Ontologies

Ontologies are defined as "the science or study of being" by the Oxford dictionary. Ontologies - specifications of what exists, or what we can say about the world - have been around at least since Aristotle, as noticed in (Brewster & O'Harra 2004). In the modern sense, an ontology can be defined as a formal, explicit specification of a shared conceptualisation of a domain of interest (Gruber 1993). Ontologies therefore provide a shared and common understanding of a domain that can be communicated between people and application systems as well (Fensel 2000). Alternatively, an

[30] <http://en.wikipedia.org/wiki/Knowledge_management#Definition>

ontology can be defined as a specification of concepts to be used for expressing knowledge which comprises types of entities, attributes and properties, relations and functions, and constraints, a definition by the Knowledge Systems Laboratory, Stanford (Dekkers 2000). To put it more simply, ontologies represent networks of concepts and relationships between them, which are formally defined.

Having a formal nature, ontologies are suitable for use by humans and machine alike and for conveying that domain understanding, i.e. knowledge, among participants in the information exchange. They provide the vocabulary or names for referring to the terms in the given subject area and the logical statements that describe what the terms are and how they relate to each other.

4.2.2 The Semantic Web

The key to reaching the goals of the Semantic Web lies in the use of ontologies as a way to assign meaning to the information in Web pages. The idea of a Semantic Web consisting of ontologies and controlled vocabularies is gaining momentum. A collaboration of a large number of organisations, both academic and industrial, is congregated in the body called the Semantic Web Coordination Group[31], which represents the main Semantic Web task force, among other bodies such as the Semantic Web Agreement group (*SWAG*[32]).

The essential components of Semantic Web technology are according to (Preece & Decker 2002):

- A common data model, currently Resource Description Framework (*RDF*)[33];
- Ontologies of standardised terminology for the domain representation;
- Languages based on RDF such as DARPA Agent Markup Language plus Ontology Inference Layer (*DAML+OIL*)[34], for developing ontologies for marking up Web resources.

---

[31] Semantic Web Coordination Group Home page, Available from World Wide Web: <http://www.w3.org/2001/sw/CG/>.

[32] Semantic Web Agreement Group Home page, Available from World Wide Web: <http://swag.webns.net>.

[33] Resource Description Framework (RDF), Collection of resources at the W3C Semantic Web Activity Website, Available from World Wide Web: <http://www.w3.org/RDF/>.

RDF is a framework or language for representing metadata. The RDF model uses object-property-value triplets to represent relationships. An object (i.e. a Web resource, such as "http://www.example.org/index/html") has a property (for example *creator*) which possesses a value (another resource or a literal such as "John Smith"). It builds on the XML syntax and imposes structural constraints to express semantics. RDF is based on a formal model of directed graphs used for the representation of the semantics of metadata, i.e. their relationships. RDF offers a means of publishing and sharing well defined machine-processable vocabularies among different information communities. The RDF model can be also considered as an extension of the classical Entity-Relationship (*ER*) model (Chen 1976) adapted for the Web, in such a way that relationships are promoted to first class objects that can be created independently (Berners-Lee 1998a). Ontology can be expressed as a collection of related RDF statements, which together specify a variety of relationships among data elements and ways of making logical inferences among them (Cherry 2002).

The RDF Schema (Brickley & Guha 2004) introduces a basic vocabulary for a statement's meaning in a metadata description and for the relation between two metadata descriptions (Steinacker *et al.* 2001). The RDF Schema currently serves to build a bridge between simpler metadata descriptions schemas such as Dublin Core and formal domain ontologies. Ontologies can be developed using description languages such as the Ontology Interchange Language (*OIL*) (Horrocks *et al.* 2000). The concepts of the ontology can be encoded with the RDF Schema while a Web-based resource's metadata descriptions can also be encoded in the RDF Schema. The combination of semantic networks or ontologies with descriptions of Web-based resources will eventually lead to the Semantic Web (Decker *et al.* 2000).

As a part of the Semantic Web initiative, the latest research is focused on development of so called *Topic Maps*[35], presented in (Pepper 2000). Topic maps are smart indices for improving search capabilities by categorising subjects in topics, and using associations and occurrences. The idea of TopicMaps is not new, but

---

[34] DAML: The DARPA Agent Markup Language Home page, Available from World Wide Web: <http://www.daml.org/>.

[35] <http://esw.w3.org/topic/TopicMaps>

recently that research has been revived in the context of the Semantic Web. OASIS[36], is a non profit consortium that promotes the development, convergence, and adoption of e-business standards, that is exploring ways to standardise various aspects of topic maps, as stated in (Paulson 2005).

4.2.3 Hypermedia and the Semantic Web

The focus of hypermedia is about expressing relationships between things, as discussed in the 1st International Workshop on Hypermedia and the Semantic Web[37] and as such, it can be thought of as closely related to the Semantic Web. Hypermedia links can be thought of as ontological relationships that can be navigated. It can be considered that the Semantic Web paradigm of providing another semantic layer on top of the Web pages by separately keeping relationships in ontologies, more closely comply with hypermedia principles than the current Web. Thus, it can be argued that the Semantic Web initiative could move the Web towards becoming a 'proper' hypermedia system by providing better management of links between Web objects.

4.2.4 Recommender Systems and Ontologies

Some hybrid recommender systems include augmenting the recommender systems by using ontologies as described in (Middleton *et al.* 2002). Ontologies can be used to bootstrap the recommender system by supporting classification and enabling explicit user modelling using ontological user profiling (Middleton *et al.* 2004). Ontologies can also be used to enable knowledge sharing among agents that recommend items, as in the lesson plan sequencing system (Yang *et al.* 2002). In the example of the collaborative filtering recommender system described in (Mobasher *et al.* 2003), knowledge extraction based on ontologies serves to supplement user ratings. Bootstrapping the collaborative filtering ratings based on a knowledge-based approach and explicit models of recommendation items, is undertaken in the FindMe project (Burke 1999).

This approach, in general, can be considered as a knowledge-based approach to recommender systems.

---

[36] OASIS Home, Available from World Wide Web: <http://www.oasis-open.org/home>.

[37] August, Nottingham, UK, Available from World Wide Web: <http://www.ecs.soton.ac.uk/~dem/workshops/htsw2003/>.

4.2.5 Web Mining and Association Rules

Web mining is the process of discovering potentially useful and previously unknown information and knowledge from Web data (Cooley *et al.* 1997). It seeks to extract knowledge from the ever growing Web data.

There are three main categories of Web mining (Kolari & Joshi 2004):

- content mining (analysing the content of Web resources),
- structure mining (analysing the graph of Web links)
- and usage mining (mining Web logs and interaction databases).

The area of Web mining most relevant to this thesis is called *text mining* and represents a subcategory of Web content mining that does not use the Web structure. It is closely related to both recommender systems and the Semantic Web. For instance, a recommender system WebWatcher (Joachims *et al.* 1997) uses content and, to a degree, structure mining techniques to provide guided tours on the Web.

Data mining includes link analysis in order to find associations. Association rules, first introduced in (Agrawal *et al.* 1993), are used to identify associations or correlation relationships across items mined in a large set of data, and then formulate rules. Recommender system in works of (Mobasher *et al.* 2001) is mining user sessions from Web logs using association-rule algorithms.

Seminal research in the area of Web structure mining is Kleinberg's HITS (Hyperlink-Induced Topic Search) algorithm (Kleinberg 1999). By using network traversal and a weighting schema the HITS algorithm identifies two categories of Web pages: a) authorities i.e. quality pages about certain topic topics, and b) hubs, i.e. pages that link to many good authorities.

Finally, Berendt *et al.* (2002) claim that Web mining enables the Semantic Web vision, and, at the same time, the Semantic Web infrastructure improves Web mining effectiveness.

4.3 Structures for Representing Complex Connected Information

There is a wealth of techniques for representing knowledge, as investigated in the

artificial intelligence area over the years (Sowa 1984). Ontologies can range from a flat set of metadata, such as Dublin Core, via hierarchical structures, to the most complex graph-like structures i.e. networks. There exist numerous structures that can be used for knowledge representation. As this thesis focuses on representing knowledge in so called zzstructures, here we shall briefly present only the subset of structures that are similar or particularly relevant to zzstructures: hierarchical structures, semantic nets, mSpaces and finally zzstructures themselves. The listed structures are briefly described and considered with regard to their suitability for knowledge representation.

4.3.1 Hierarchies, Taxonomies and Thesauri

A simple tree structures with single parent and multiple children nodes, are the most commonly used to organise information for representing knowledge categories and their subcategories. Hierarchies are very common in contemporary computing. Most information management systems employ hierarchies as the dominant paradigm and encourage categorising information by putting them into a particular place (Dourish *et al.* 2000). For example, a directory, also known as a folder or a catalogue, binds a group of files together.

Taxonomies are used for classifying any kinds of things or objects. A good example would be a taxonomy in the animal kingdom where we have class, order, family, genus and species hierarchical levels. Taxonomies are usually hierarchical by nature, however there are so called faceted taxonomies which consist of multiple tree taxonomies called facets (Tzitzikas *et al.* 2002). Faceted taxonomies perform classification by different aspects (facets) and must obey a restriction that the individual taxonomies must be exclusive. Using a combination of levels from different facets the taxonomy can be queried in a more effective way. However, some combinations of levels are incompatible and do not bring any results.

Sometimes in real life things can not be categorised to exclusively belong to one category only. Therefore some taxonomies, as in examples of Website directories such as the Open Directory Project[38], must allow for either horizontal links between

[38] ODP home page, Available from World Wide Web: <http://www.dmoz.org>.

nodes (thus breaking the single parent rule) or for the duplication of nodes. Also there exists a kind of multiple intersecting hierarchies, sharing at least one node, that are called Polyarchies. Polyarchies combine multiple trees that intersect at one or more nodes and therefore have common sub-hierarchies. They are very difficult to visualise (Robertson *et al.* 2002a; 2002b).

The oldest and most wide known systems for representing semantic relationships are thesauri, tracing back to the Dewey Decimal Classification system first published in the 19th century. A thesaurus can be defined as a structure that holds semantic relationships between index terms (Aitchison & Gilchrist 1987). In other words, a thesaurus is a set of terms connected by a set of relations (Jones *et al.* 1994). Thesauri are usually multi-faceted complex structures which aim to support three main types of relationships: *equivalence* (equivalent terms, synonyms), *hierarchical* (broader/narrower terms) and *associative* (related terms). Additionally thesauri typically support the *preferred term* relationship as well.

4.3.2 Semantic Nets

Semantic nets can be thought of as a classical family of knowledge representational schemes dating from the early days of the artificial intelligence research. The objective for semantic net was to become the "representational format [that would] permit the `meanings' of words to be stored, so that humanlike use of these meanings is possible" (Quillian 1968). A semantic network, or a semantic net, is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs (Shapiro 1987). Semantic nets represent a way to model the domain knowledge by showing relationships between objects, classes or concepts. Alternatively, a semantic net can be defined as a directed graph consisting of vertices which represent objects (concepts), and edges which describe semantic (usually hierarchical) relations between the objects (Turban 1995). In principle, there are no restrictions on the number of edges, and the structure can become quite a complex network.

Semantic nets consists of *nodes*, denoting objects, *links*, denoting relationships between objects, and *link labels* that denote particular relationships (Winston 1993). The term link is used in a similar meaning as in hypertext. One example of nodes, links and link labels is shown in figure 4-1. One of the most common relationships

between objects is a relationship of instance and its class, known as *is-a-member-of-the-class*, or shortened *isA*.

4.3.3 mSpace

"mSpace is an interaction model which exposes relationships within an information space and which provides a set of manipulations on that space to assist the exploration of those relationships" (Gibbins *et al.* 2004; schraefel *et al.* 2005a; 2005b). It can be viewed as a specialized kind of polyarchy where the levels of hierarchy trees are shown and manipulated in a multipane browser (McGuffin & schraefel 2004; Wilson *et al.* 2005). mSpaces and the mSpace browser have been developed at University of Southampton in recent years.

The mSpace browser enables users to browse the multidimensional space of data points (usually strings, i.e. text or images) in such a way that certain dimensions can be fixed and later on changed in a flexible way. The user is presented with a tool to organise multiple hierarchies in such a way that many possible trees can be browsed. The mSpace interface contains a number of columns where each column represents one dimension in the information space.

In the example in figure 4-2 the level of the dimension *Era* is fixed to *Classical* while the rest of the panes in the browser show other possible trees from that point on. The information visible in the other columns is restricted by the selection of the level on the column more to the left. Similar logic is applied to the other columns and users can keep adding, removing or reordering columns representing a desired dimension e.g. the type of categorisation such as Composer, Album etc.

The mSpace browser represents a generic Semantic Web browser that can assist users in surfing a complex linked information space. It moves beyond the Web hypertext by enabling a what-if exploration and showing associations between parts of different domains, promptly on user's request (schraefel *et al.* 2004).

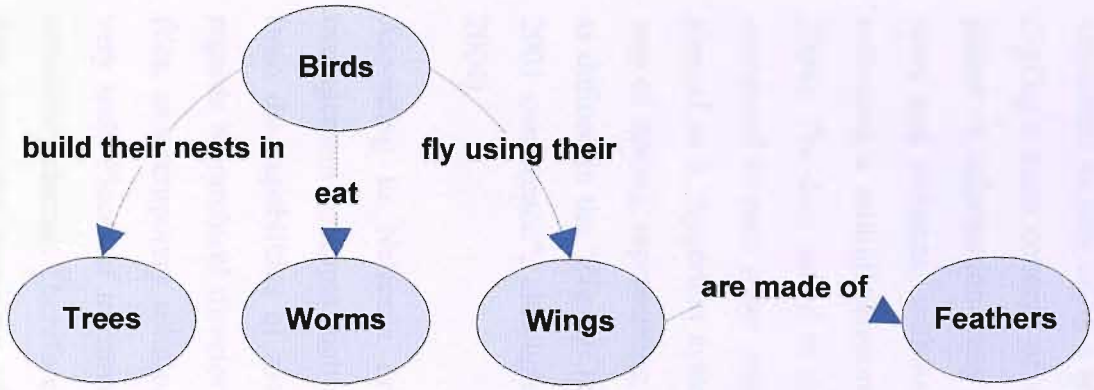


Figure 4-1: An example of a semantic net: bird related concepts, after (Poget 1999)

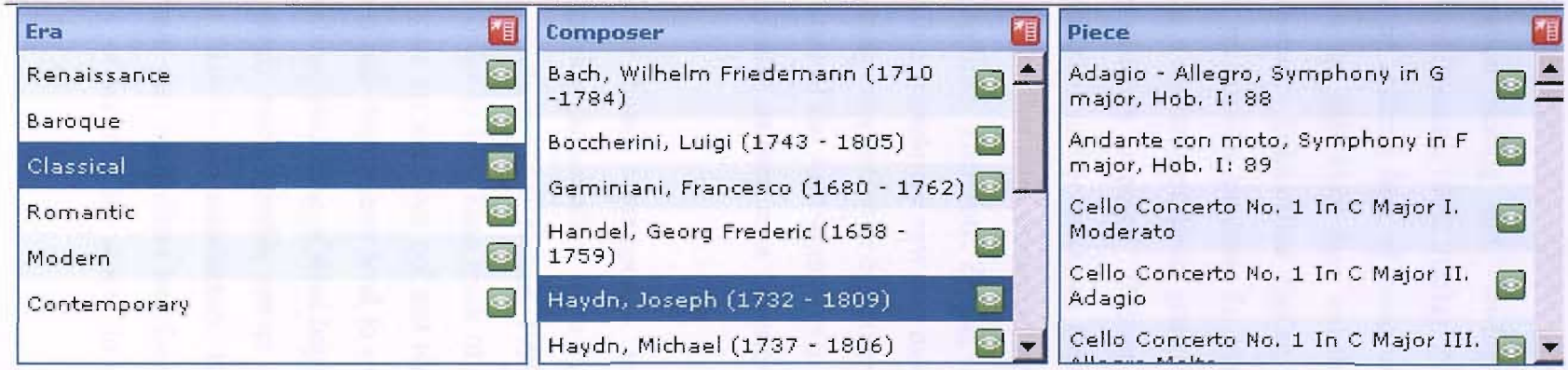


Figure 4-2: mSpaces browser example, after mSpaces demo, Available from World Wide Web: <http://demo.mspace.fm/>

4.4 ZigZag

4.4.1 ZigZag and zzstructures

ZigZag[39] represents an innovative information storing and viewing paradigm introduced by the hypertext pioneer Ted Nelson (Nelson, 1998; 2003a). ZigZag in its core can be defined as a space consisting of ordered lists of basic units (cells) intersecting at multiple points. Cells contain the individual data and they can be associated in lists using a special kind of adjacency links called ZigZag links here. ZigZag's main concept of using a complex matrix-like structure for manipulating pieces of information "may be thought of as a multidimensional generalization of rows and columns, without any shape or structure imposed" (Nelson 1998). It embodies a multidimensional lattice of principled interconnections (Moore *et al.* 2004). The data stored in the elementary cells of this interconnective structure are connected to each other using untyped ZigZag links. Therefore, ZigZag can also be viewed as a "hypertext system that deals with a completely new and more flexible way of storing, representing, arranging and using information in computer systems", as defined in the "Zigzag: Introduction and State of the Art" workshop in Hypertext 2001 conference[40], although it was not originally defined as a hypertext (Nelson 2004).

According to Nelson's original ideas, some basic contemporary information management concepts such as files and directories, do not allow us to truly benefit from the capabilities of modern computers (Nelson 1965; Nelson 1999). Nelson regards hierarchical directories, invented in order to help keep track of the list of files, as a temporary solution that does not scale up and does not suit real projects very well. Pieces of information from various real-life projects tend to overlap and constantly change. Therefore, according to Nelson, modelling of what happens to the data using the paradigm of files with relatively unchangeable names and using directories that hierarchically organise those files, is not satisfactory. It has been noted in the frontier IEEE magazine Spectrum that the folders interface does not cope sufficiently well with today's quantity of information stored in computers

---

[39] ZigZag Home, Collection of resources, Available from World Wide Web: <http://xanadu.com/zigzag/>.

[40] Available from World Wide Web: <http://www.sigweb.org/conferences/ht-conferences-archive/ht01/workshops.html>#w4>

(Wohl 2005).

Nelson proposes a new concept of presenting information in computers, a new data layer called ZigZag (Nelson 2001; 2003a). Ted Nelson sees ZigZag as the kernel of a completely new approach to computing (Ziff-Davis 1998). It can be argued that by adding a notion of a network to an associative file system, Nelson indirectly extends the concept of Bush's memex (schraefel *et al.* 2004).

The smallest unit of information in ZigZag is called a cell. The cell, or *zzcell*, is a first class object and a principal unit of the system (Nelson 2004). The cell can contain something as simple as a character or it can contain a picture or some other object. Cells of type text are especially relevant to this thesis. Cells are connected to each other along an unlimited number of dimensions, which effectively represent types of relationships. In order to best visualise such a structure, as a starting point we can imagine a simple two-dimensional spreadsheet with rows and columns (see figure 4-3).

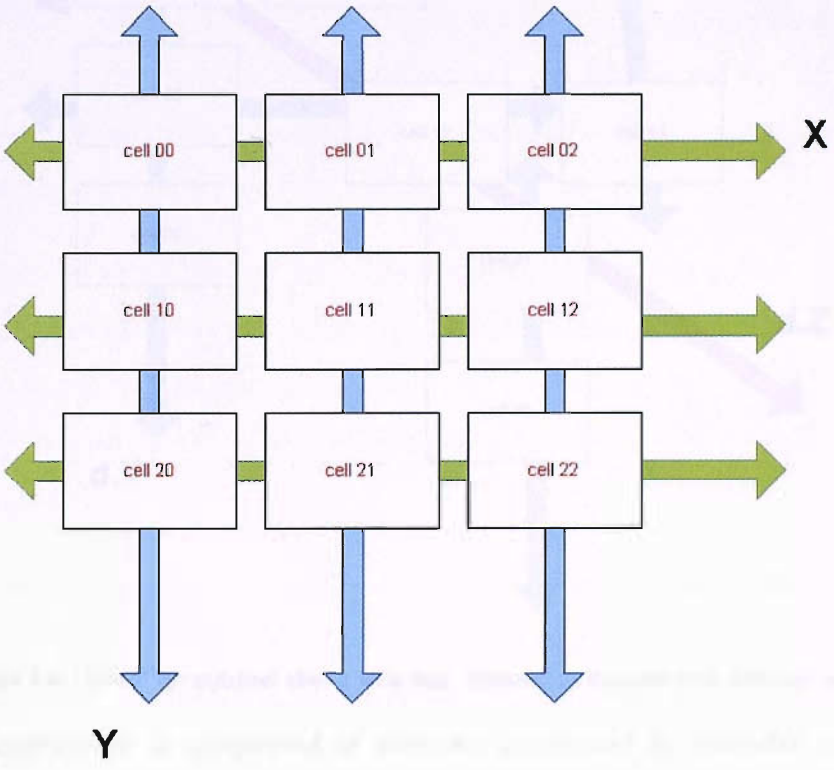


Figure 4-3: A typical two dimensional spreadsheet

Cells on the spreadsheet are connected to each other by a horizontal dimension

represented by rows and a vertical dimension represented by columns. The horizontal dimension denotes the relationship in which are cells ordered in such a way that they might have a left and a right neighbour. Similarly, a cell participates in the vertical dimension that determines which cell is under or above the given cell using the rules of the vertical relationship. ZigZag allows the introduction of multiple dimensions. If we imagine that a cell is freed from the matrix and enable it to connect with any other cell in any new dimension, we get a ZigZag structure (see figure 4-4 for a very crude sketch of the structure).

However, this way of connecting cells in the structure has some limitations. The ZigZag structures conforming to ZigZag rules of connecting are usually referred to as zzstructures. This rule can be formally expressed with the following definition:

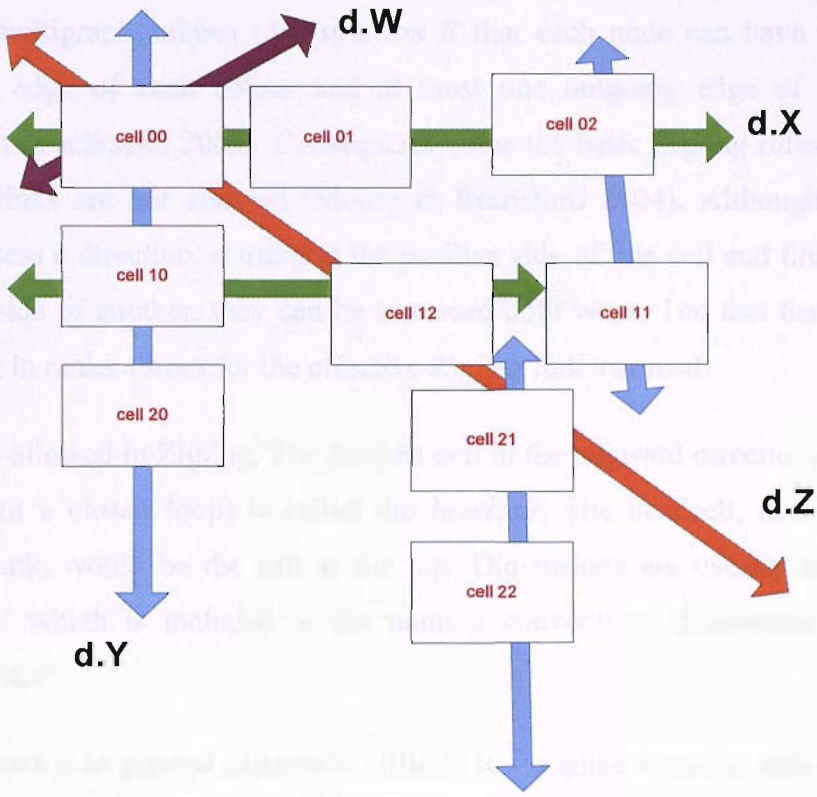


Figure 4-4: 'Freed' spreadsheet shown with four dimensions denoted with different colours

Rule R: zzstructure is comprised of elements connected by pairwise symmetrical untyped links having a nominal ordering where a>b or b>a (exceptional case: loop of two elements where both a>b and b>a) (Nelson 2000).

A cell can participate in a dimension by connecting to the other cell(s) (or in the special case to itself) via two sides, positive and negative. For a particular dimension, a cell can be connected in such a way that none, one or both of the sides are used. Therefore, there exists a constraint that a cell can have at most one neighbour on each side, positive (pos) or negative (neg) and up to two directions for ZigZag links: *posward* and *negward*, all this in one dimension. Unlike cells, ZigZag links are not first-class objects - they can not be identified and addressed independently.

Cells are connected in a series of sequences or lists if only one dimension is observed at a time. Some cells do not have any ZigZag links; some can have both positive and negative neighbours. It is convenient to colour the ZigZag links belonging to the same dimension with the same colour. This collection of strands or paths, in zzstructure called ranks, forms a network, effectively, an edge-coloured directed multigraph subject to *restriction R* that each node can have at most one incoming edge of each colour and at most one outgoing edge of each colour (McGuffin & schraefel 2004). Consequently, one the basic ZigZag rules is that one-to-many links are not allowed (Moore & Brailsford 2004). Although zzstructure links possess a direction, starting at the positive side of one cell and finishing in the negative side of another, they can be traversed both ways. The fact that there is no branching in ranks allows for the effective ZigZag link traversal.

Loops are allowed in ZigZag. The furthest cell in the negward direction of each rank (that is not a closed loop) is called the *headcell*. The headcell, in a case of the vertical rank, would be the cell at the top. Dimensions are usually denoted by a prefix 'd.' which is included in the naming convention '**d.**dimensionname', for example "d.x".

The structure is in general extremely difficult to visualise. Usually, only a portion of the whole structure is shown at a time, typically revealing cells and ZigZag links in 2 or 3 dimensions. A selected set of cells called *Raster*, presented in a certain way, is called a *View*.

4.4.2 zzstructure Clones

A very important notion in ZigZag, a potentially crucial unique feature in comparison to previously mentioned information representation structures, is a notion of *cloning*. Cloning (live copying) represents an implementation of transclusion (the knowable identity of more than one thing) at the cell level (Moore *et al.* 2004; Nelson 2004). The zzstructure principle allows for an interesting effect of criss-crossing lists, where a cell can exist on many lists at the same time (Nelson 1998), and effectively enable transclusion of the content. Therefore, cloning is considered to be one of the basic structural mechanisms in ZigZag (Lukka 2001).

Cells cannot repeat within the same structure, but if necessary, a cell can be cloned in order to for example participate in one-to-many relationship. In this case, a virtual repetition of the cell content is achieved, in order to enable the cell to be connected to many other cells via the same dimension, without breaking the ZigZag restrictions.

Clones are special cells with the following two features:

- They possess a dynamic reference to the original, source cell, which prevents duplication of data. Changing the original or one of the cloned cells changes them all.
- They are knitted with each other and with their head cell containing the original cell in a special dimension called Clone dimension.

As regular cells, clone cells respect the ZigZag link limitations, meaning that they have only up to two neighbours in a Clone dimension. Traversing ranks in clone dimension allows for identification of the original cell and all its clones. One clone rank could look like this example: Original – Clone1 – Clone2 – Clone 3.

There are other ways in which one-to-many (or many-to-one/many-to-many) relationships can be represented in zzstructures, such as use of additional dimensions or mimicking connections by other structures. This thesis adopted the approach of using clones.

4.4.3 zzstructure Examples

An excellent example of a zzstructure is the system of London underground train lines and stations. Stations represent cells while the train lines can be considered as dimensions. Some stations can belong to more than one line, where different ranks intersect. Moreover, in the example of the London tube system given in figure 4-5, each line is given a name and a specific colour. A traveller on the network can follow some route (rank) or change the line (dimension) on a certain station (cell), providing that such cell offers a choice of interconnection.

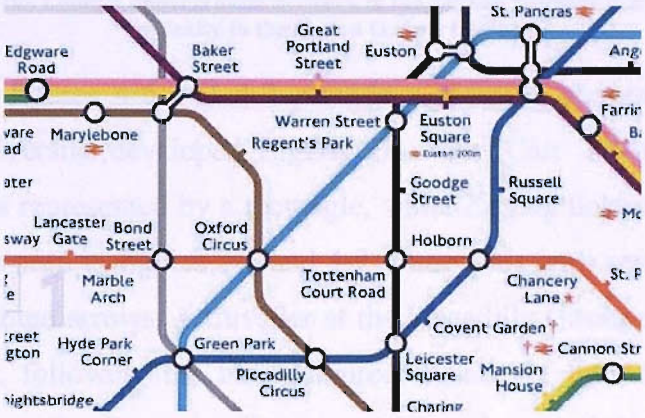


Figure 4-5: Portion of the London underground network on a map, after <http://www.tfl.gov.uk/tube/maps/>

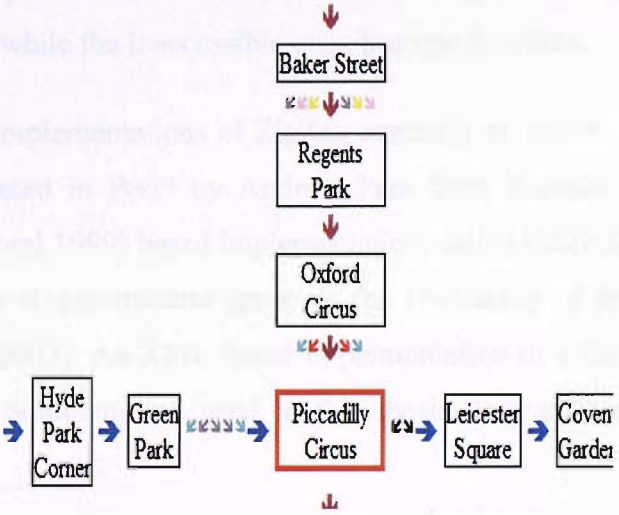


Figure 4-6: Portion of the London underground network in the ZigZag Browser (Carr 2001a) – current station Piccadilly Circus

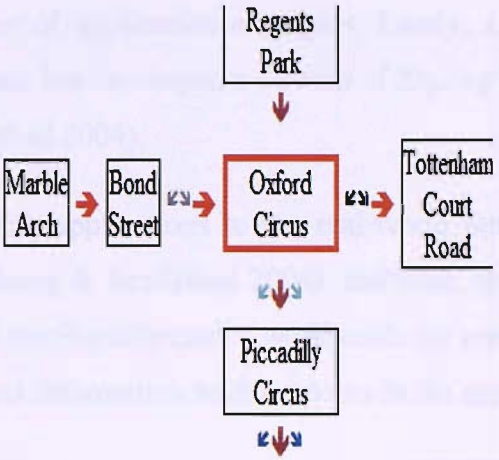


Figure 4-7: Portion of the London underground network in the ZigZag Browser (Carr 2001a) – navigating vertically to the station Oxford Circus

The diagrams in figures 4-6 and 4-7 provide a view on the zzstructure using a Southampton University developed ZigZag Browser (Carr 2001a). The cell in a ZigZag browser is represented by a rectangle, while ZigZag links are represented as arrows. As can be seen in figures 4-6 and 4-7, some cells have several ZigZag links indicated with slanted arrows. A traveller at the Piccadilly Circus station can decide to continue right, following the blue-coloured, Piccadilly line, towards Leicester Square, or to change the dimension/line to a purple-coloured, Bakerloo Line, and go up to Oxford Circus as shown in figure 4-7. As the user moves in the multidimensional space of the train lines, the currently accessible cells and ZigZag links are revealed, while the inaccessible ones become invisible.

There are several implementations of ZigZag currently available. The initial ZigZag prototype was created in Perl[41] by Andrew Pam from Xanadu Australia (ZigZag 1999). A Java[42] (Goraj 1999) based implementation, called GZZ/GZigZag, originated as a project in the Hyperstructure group at the University of Jyvaskyla, Finland[43] (Lukka & Ervasti 2003). An XML based implementation of a ZigZag browser from the University of Southampton, used in this thesis, is available at (Carr 2001a),

---

[41] The Source for Perl -- perl development, perl conferences, Available from World Wide Web: <http://www.perl.com/>.

[42] Java Technology, Available from World Wide Web: <http://java.sun.com/>.

[43] Collection of resources, Available from World Wide Web: <http://gzigzag.sourceforge.net/> and <http://www.nongnu.org/gzz/>.

together with a number of application examples. Lately, a research group at the University of Nottingham has developed a version of ZigZag known as ZZZ (Nelson 2003b; Moore & Brailsford 2004).

Two examples of ZigZag applications to the real-world domains are described in (Moore *et al.* 2004; Moore & Brailsford 2004): zzPhone, the information manager for mobile phones, and the Bioinformatics workbench for creating and manipulating interconnected biological information such as atoms in the metabolic Krebs Cycle.

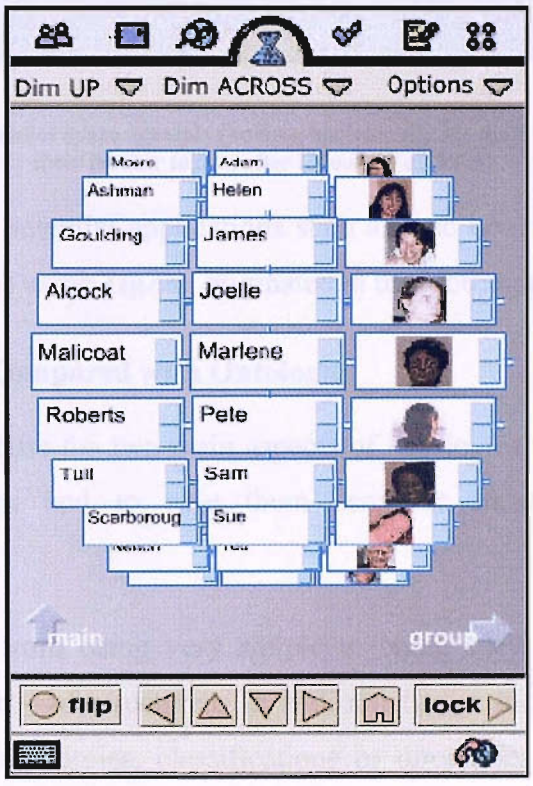


Figure 4-8: The zzPhone example showing the Contacts, Firstname and Photo ranks, after (Moore *et al.* 2004)

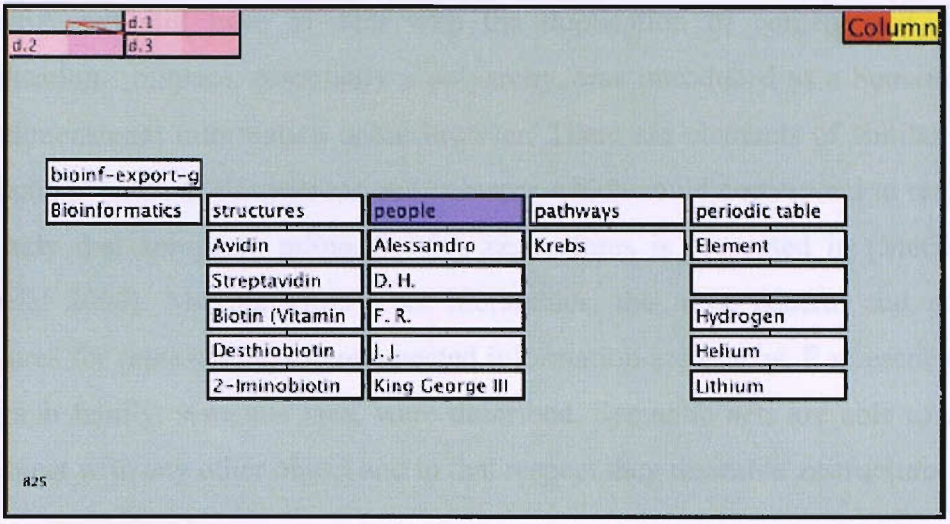


Figure 4-9: The Bioinformatics space example showing horizontally the main categories and vertically their further topics, after (Moore *et al.* 2004)

More ideas about the potential applications such as web bookmark management or browsing website maps using ZigZag originated at the Nottingham University.

4.4.4 zzstructures Compared with Ontologies

In this section we discuss the two main aspects of ontologies, namely the ability to represent relationships and to give them semantic meaning, in relation to zzstructures.

Ontologies can vary from being very simple to being very complex and can be represented in a variety of structures. In addition to the hierarchical relationship structure of typical taxonomies, classifications or directories, complex ontologies must make possible cross node relationships between entities, in order to better model real-word associations.

Earlier in this chapter we briefly presented selected information-representation structures which are used for ontologies and are relevant to zzstructures. Firstly simple hierarchies (classifications/taxonomies) are shown not to be able to represent any other than parent-child relationship. Lack of expressiveness of hierarchies was the main motivator for Ted Nelson to envisage such multidimensional structures as zzstructure. Complex hierarchies: faceted taxonomies (non-intersecting set of hierarchies) and polyarchies (intersecting set of hierarchies) were shown to be more

powerful, but still have to deal with the duplication of content and complex visualisation. mSpace, essentially a polyarchy, was introduced as a Semantic Web multidimensional information space browser. There are elements of similarity with zzstructures (such as dimensions and columns which could correspond to ranks) and the study that compares mSpaces with zzstructures is presented in (McGuffin & schraefel 2004). Moving away from hierarchies, the most general and powerful structures for representing interconnected information are graphs. Representatives of the graph family, semantic nets, were described. Semantic nets are able to connect any object with any other object and in that respect they resemble zzstructures.

It has been shown in (McGuffin & schraefel 2004; McGuffin 2004) that zzstructures are more powerful and more general (meaning that they can subsume) than edge-coloured directed multigraphs, in particular more general than lists (1D arrays), hierarchies and even polyarchies. Semantic Web ontologies are most commonly represented in RDF, which is itself a directed graph. According to the findings in the mentioned study any given edge-coloured directed multigraph can be converted to a zzstructure with no loss of information and also these kinds of graphs subsume any directed graphs. Therefore, we conclude that it is possible to store in a zzstructure any complexity of interconnections required from an ontological structure that can be represented by RDF.

Secondly, formal ontologies must be more robust than for example faceted classification because they allow for richer semantic relationships and especially restrictions on those relationships. The question here is can zzstructures provide the means to define and adhere to relationship restrictions? We argue that this is possible in principle. Definitions of rules for relationships can be added as new cells or dimensions into a standard zzstructure. The introduction of zzcell programming and so called *applitudes* (a kind of application in ZigZag) (ZigZag 1999; Nelson 2004) indicates that a dynamic component can be added to zzstructures as well. Applitudes can be used to execute the ontology rules on the relevant cells. However, the practicality of this approach still remains an open research issue.

In this research, a zzstructure implementation is chosen as a mechanism for storing

and presenting ontologies containing metadata and their ZigZag links. We decided to explore creating ontological structures using zzstructure because of the following reasons:

- Simplicity, elegance of implementation and easy traversal: there are only three concepts used, cell, dimension and ZigZag link;
- Transclusion of content is enabling easy maintenance of entities as they are stored only once;
- Ability to clearly express and easily create new types of associations between any entities allows for accurate modelling of the real world relationships;
- Accommodating low level restrictions of ZigZag links does not present serious limitation and is easily overcome.

This dissertation looks in more detail into the suitability of a zzstructure for such a task, which shall be described in section 5.4 (ZigZag for Browsing Simple Ontologies: ZZDirectory) and Chapter 6 ("à la": Associative Linking of Attributes).

4.5 Knowledge-based Information Retrieval

We shall take a closer look here at the possibilities of using interconnected knowledge in information retrieval.

Various studies on the Web confirmed that users mostly post very short queries, one or two words (Silverstein 1998; 1999; Wen *et al.* 2000). Research in (Pinkerton 1994; Anick 1994) states that the majority of users submit to search engines on the Internet, either a single key word query or up to 3 words query. Short queries are usually insufficient to describe the user's interest. Therefore if IR engines then try to exactly match queries to document terms, both precision and recall of a result set will suffer. To overcome this problem a *query expansion* has been introduced to assist users on formulating a refined or expanded query (Cui *et al.* 2002). Query expansion is "a process of adding new terms to a given user query in an attempt to provide better contextualization (and hopefully retrieve documents which are more useful to the user)" (Baeza-Yates & Ribeiro-Neto 1999). In this process a query is augmented with or without the user's participation, i.e. automatically or manually.

Recommender systems can provide help to users in order find the correct words for a successful search by supporting iterations of query reformulation (Belkin 2000). Techniques such as LSA can be used to offer similar terms (Alaniz *et al.* 2002; Stenmark 2005). Associating query terms with all document terms for the document user has marked as an acceptable result for the given query (Hang *et al.* 2002), allows for an alternative solution of recommending connected terms.

In addition, techniques for using knowledge of how the terms are interconnected in ontologies or other network-connected information spaces, are applied to improve the query reformulation and the retrieval itself, advancing in the direction of more intelligent, *semantic search.* Some IR engines use term co-occurrence in the query expansion algorithms, which enables using keywords that are semantically connected. Other, like for example Froogle[44] attempt to connect query terms with the other metadata describing items in the collection.

The most common technique used in semantic networks is called *spread activation* (Cohen & Kjeldsen 1987; Crestani 1997). In essence, the spread activation algorithm starts from a set of nodes and some restriction rules, and then the activation flows through the network reaching other connected nodes (Rocha *et al.* 2004). A good example of the spread activation for automatic query expansion applied on a thesaurus, is the work of Alani and others in the Ontologically Augmented Spatial Information System (*OASIS*) project (Alani *et al.* 2000). Semantic searchers using the activation method are described in (Guha *et al.* 2003; Stojanovic *et al.* 2003; Rocha *et al.* 2004). In the Semantic Web, ontologies can be used as the underlying framework for the ontology network analysis using spread activation (Alani *et al.* 2002; Middleton *et al.* 2002).

4.6 Summary

The objectives of this chapter were to introduce the knowledge management field and present the knowledge management background relevant to this thesis. Firstly, ontologies and the Semantic Web were defined in order to illustrate the

[44] <http://froogle.google.com>

contemporary approach taken in knowledge management. Then the relationship between the Semantic Web and hypermedia was considered because this thesis looks into hypertext systems used for knowledge manipulation. At the same time, this research is highly related to recommender systems and therefore the relationship between ontologies and recommender systems was also examined.

This chapter also described various structures used to organise pieces of interconnected knowledge: simple and complex hierarchies, and mSpaces as user controlled polyarchies. ZigZag, the author's chosen method for information management, was presented in more detail in the context of knowledge management. Finally, some aspects of using information spaces for semantically backed information retrieval were discussed.

Three background chapters (Chapters 2, 3 and 4) have covered the relevant research fields following how the scope of this thesis' work has been historically widened. The following chapter will present the author's early work in document management, recommender service and knowledge representation and visualisation.

Chapter 5

Initial Research in Document Management, Recommending and ZigZag

Our initial investigation in the field of document management is described in this section. The aim was to consider information retrieval using metadata. The goal of this piece of work was to create a document management system that could be easily extended to include new research ideas of introducing flexible metadata types and controlled vocabulary and to serve as a 'proof-of-concept system' for future work. A prototype system, called "AWOCADO", is presented.

The initial recommender system research work, a guided tour builder called "MAGENTA" is presented in this chapter as well, and finally, "ZZDirectory", an investigation into an application of ZigZag for Web taxonomies, is described.

5.1 AWOCADO, MAGENTA and ZZDirectory Overview

This chapter discusses three separate systems that were built in order to investigate the aforementioned research areas:

- AWOCADO, a document management system;
- MAGENTA, a recommender facility extending the MEMOIR system developed by other authors;
- ZZDirectory, a system based on ZigZag for storing and browsing taxonomies.

The reason for presenting those systems here the following: the first system AWOCADO is in our second phase of the research extended with a recommender facility inspired by MEMOIR/MAGENTA, and a ZigZag-inspired knowledge browser developed in ZZDirectory.

5.2 Metadata in Document Management Systems: AWOCADO

5.2.1 Research Motivation

Locating digital documents in modern organisations with the aid of metadata, is a challenging area of research in document management/content management systems.

Using metadata, i.e. document attributes, associated with the documents substantially improves the efficiency and accuracy of location. The use of metadata, however, is still not easy mainly because of the difficulty in defining and then effectively using such metadata. Section 5.1 will firstly elaborate on the current limitations of metadata management in document management systems and then present our initial research conducted in order to overcome those limitations.

The majority of contemporary document management systems are somewhat restricted, sharing the same kind of limitations related to metadata specification and usage (Hendley 2005). In some cases the attribute set is fixed and limited to only basic document properties such as *Author* or *Title*. Slightly better solutions allow for a set of attributes to be extended, however the same set of attributes is applied to all types of documents. Ideally, different classes of documents would use different sets of attributes; in the context of one document type only certain attributes might be applicable. Moreover, in some systems it is not possible to specify a controlled vocabulary for attributes. This limitation is significant since with attribute searching even a tiny inaccuracy in the query can result in no items being found.

Having reviewed these limitations, we attempted to develop a document management system that overcomes them. AWOCADO (Adaptive WOrkflow Controller And Document Organiser) represents an attempt to introduce adaptable metadata management into document management systems. The AWOCADO prototype system provides a novel framework for defining and managing document attributes and locating documents using those attributes. Another important goal of this work was to provide a test bed for investigating the use of metadata for searching.

5.2.2 System Overview

The acronym AWOCADO stands for the Adaptive WOrkflow Controller And Document Organiser. The idea of AWOCADO represents a continuation of the author's early research on general-purpose document management systems (Damjanovic & Andric 1997; 1998).

AWOCADO handles documents and meta information about documents. It also facilitates the exchange of documents and messages among participants. The AWOCADO system can be best described as an enriched internal mailing system combined with a searchable "source control"-like repository, benefiting end-users: creators, editors and consumers of the documents in the repository.

An experimental AWOCADO system prototype was developed with the following aims in mind:

- To serve as a Web-based (intranet) document and metadata repository,
- To operate in a multi-user environment,
- To allow flexible attribute definitions based on a defined document class,
- To allow flexible searching by adapting a set of search attributes based on a document class context,
- To introduce controlled vocabulary for selected attribute types,
- To incorporate a simple workflow system and, most importantly,
- To provide a test bed for observing how users manage and search their documents in a document management system environment.

The concept that AWOCADO tries to prove is that introduction of the extendable attribute types definitions and controlled vocabularies for the chosen types, could improve the document retrieval in a document management system.

5.2.3 AWOCADO Architecture

The architecture of AWOCADO is given in figure 5-1.

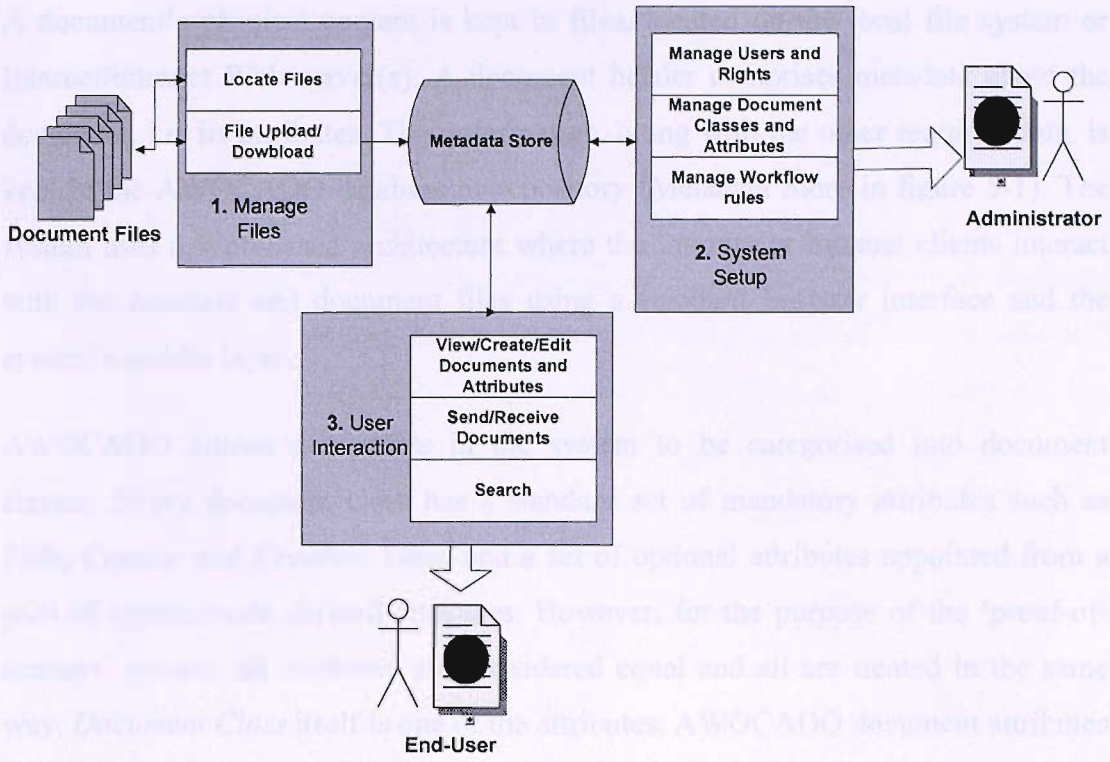


Figure 5-1: The AWOCADO system architecture

The AWOCADO system aims to support three high level groups of functions:

- Document archive management (storing, locating etc) and a workflow control (figure 5-1, part 1, Manage Files)
- System administration (such as definition of document classes, their attributes, and access control) (figure 5-1, part 2, System Setup)
- Providing an interface for end-users and enabling them to access and manipulate document collection (figure 5-1, part 3, User Interaction)

The main components of the system include:

- File manipulation component called *Manage Files*,
- Database (storage) component called *Metadata Store*,
- System initialisation component called *System Setup* module,
- *User Interaction* module, responsible for interfacing the system user.

The fundamental object in the AWOCADO system is a *document*. A document conceptually consists of a document's physical content and a document's header.

A document's physical content is kept in files, located on the local file system or Internet/intranet Web server(s). A document header comprises metadata about the document, i.e. its attributes. This information, along with the other required data, is kept in the AWOCADO database or repository (Metadata Store in figure 5-1). The system uses a Web-based architecture where the Internet or intranet clients interact with the database and document files using a standard browser interface and the system's middle layer.

AWOCADO allows documents in the system to be categorised into document classes. Every document class has a standard set of mandatory attributes such as *Title*, *Creator* and *Creation Time*, and a set of optional attributes appointed from a pool of system-wide defined attributes. However, for the purpose of the 'proof-of-concept' system, all attributes are considered equal and all are treated in the same way. *Document Class* itself is one of the attributes. AWOCADO document attributes can also serve to describe or summarise the content of documents. This is accomplished by introducing a type of attribute called *Keywords*. A document in the system is modelled by using a vector of attribute-value pairs, as in a standard vector-space model. An example is shown in figure 5-2. It should be noted that the examples in this thesis are taken from a documentation set supporting a real software development project.

Document 4242
{Document Class, "CSDB Document"}
{Title, "Formal Issues for Interface Workshop"}
{Creator, "Chris Walker"}
{Creation Date, "jan 7 2003"}
{Maj, 1}
{Min, null}
{Doc Maturity, "Draft"}
{Location, "WorkZone"}

Figure 5-2: Document attributes example in AWOCADO

Attributes could be either internal to the document management system (i.e. having a local meaning inside a system) or they could represent a link to the other legacy system, usually containing an external record identification in other databases. In other words, an attribute can hold a primary key of the object from another system.

Additional opportunities are encountered if the repository is implemented in the same database as the legacy system.

The concept of a predefined domain (controlled vocabulary) for attributes is supported. Controlled vocabulary can be considered as a simple ontology. Attributes can be set to be "discrete", in which case they can only take a value from the given set. Privileges and access control are defined and controlled at a document class or at an individual document level. Each document within AWOCADO can contain zero, one or more files with the same or different file format. Different physical files represent various document versions that are kept in a system history. If required, every change to the document (version) can be stored. Changes to the archived document are also kept and considered to be *revisions* with their own versions. A document without a file is called a record; in which case only its attributes are of interest. A Web document is an example of such a record, its duplicate is not kept in a system, only its reference. In those cases, the AWOCADO system serves as an external Web classification and bookmarking tool.

A document's physical location can be represented either using a Universal Naming Convention (*UNC*) for a local network resource or a Uniform Resource Locator (*URL*) (Berners-Lee *et al.* 1994; Ragget *et al.* 1997) for a global, Web-based resource. A document's logical location is also kept, providing the information about who is currently holding a document: is it a specific user or is a document stored in archive.

Every document can be linked to one or more documents with the following types of links:

- Simple associative link document-to-document (manually authored by document authors or administrators)
- Package (documents bound in a group by a master document)

Together with the meta-information (definitions of document classes and attributes), the AWOCADO repository also stores information, such as instances of documents in the system (their locations and physical placement), and attributes values. The

population of the repository, or system meta-database, is performed via the administrative application. The repository stores information about the system setup such as:

- Individual users and groups of users (groups can overlap),
- Document classes (each class is defined by its name and a set of attributes applicable to its future documents),
- Common bulletin boards (places where documents are published, in other words named archives),
- Definition of the automatic workflow procedures depending on the document classes and groups,
- Access and handling rights on various levels.

A document generally passes through two main stages during its lifetime. After its creation, a document can move through the workflow, between users and user groups. In every step of the workflow a new version of the document is created in a physical file (if applicable). Documents that are moving through the workflow will appear in user personal baskets: *InBasket* (inbox for received documents), *OutBasket* (documents sent out by the user) or *WorkZone* (private user's zone for creating and editing documents). When the final version is generated, a document is directed to the named archive, (named Bulletin Board) or a document is finally retired in the unnamed, general archive. The document can be recalled to the flow later. The operation of recalling effectively means returning the document to the first phase.

The details about the AWOCADO implementation are provided elsewhere (Andric & Hall 2005a; 2005b). Examples of the system usage and the user interaction walkthrough are presented in Appendix B. The example of the Web page showing document attributes is given in figure B-2. User interface for querying by specifying the search values for each of the desired attributes, is given in figures B-6 and B-7.

5.2.4 AWOCADO Evaluation

In order to evaluate this version of the AWOCADO system, an experimental study was conducted. The AWOCADO prototype was deployed in a software development

organisation and used as a prototype Web-based document management system for managing software engineering documentation and artefacts. The central aim of the trial was to draw some conclusions that might be applicable to document management systems in general, (of which AWOCADO is a representative), and also to show us directions for future work.

The trial was conducted over a period of 3 months. Five system users were then selected from the group of the most active and regular AWOCADO users, the idea being that observing their behaviour would provide the best overall picture. The main objective was to understand how users search for documents in AWOCADO and to observe in which ways they interact with the document repository. Users were observed while trying to find information in the course of their everyday work. The study was conducted during informal user sessions. The experiment consisted of individual sessions with each user. We were observing the user behaviour during the sessions and noting user actions and comments. Sessions took approximately one hour each.

Some of the most interesting observations we made are as follows:

- When posting a new document, users would often fill in, (or accept defaults for), only the mandatory attributes. Later on, this led to difficulties in finding the document because attributes by which they were searching were missing. On the other hand, searching by mandatory attributes (for example *Creator*) usually presented too many results as the users could not narrow their search enough in the first place.
- Users would often submit duplicate documents with slightly different titles or attributes because they could not find whether the document already existed in the repository. This led to problems in version management. Instances of essentially the same documents would branch and be maintained by different authors.
- Users mostly searched by *Creator*, *Title* or *Keyword*. However, they would often get frustrated by using keyword search since it required guessing. Users typically

did not know which keyword was used when the document was submitted into the system.

- Users were missing many documents in the results of their search because they did not know the right value for the search attribute.
- In the case when users knew the exact term, they sometimes did not know with which attribute type that value was associated. For example an attribute *Team name* was sometimes stored as a *Keyword* type of attribute. They needed to know this in order to appropriately fill in the search form consisting of a number of attribute-value text boxes (see figure B-6 for the example of multiple attribute-value boxes interface in AWOCADO). Frequently, users were frustrated and asked for only one box where they could type in their query, similar to the well-known search engine user interface paradigm.
- Many times, users were looking for the same thing as in previous searches but could not recollect what attributes they used previously.
- Users did not know that some colleagues had already found/searched for the same document earlier. On many occasions, users were asking their colleagues were they searching for similar things and how they accomplished that. When colleagues were contactable they were providing some recommendation from their experience.
- We noticed that users utilised explicit document to document links hardly at all.
- Finally, we observed that searching is usually conducted in two or more cycles. Users start with an initial query and then, if the right result or results are not shown at once, they continue further. The users then look for clues and terms among the search result to help them change or refine the query. This observation was confirmed in a study described in (Hotchkiss 2005): "when the search results come up, I'm looking at them through a 'semantic map' that contains many words that flesh out my concept and might catch my attention".

The following in the list of the most common attributes used in searches:

- Document Title (part of the title used in addition to wildcard symbols, such as "Iteration%" for finding all titles that start with the string "Iteration");

- Document Type (organisational classification of documents, such as "Meeting Minutes", "Technical Specification" etc.);
- User/Creator (creators of documents);
- Team (usually the creator's organisational unit, such as "Middleware" or "Operations");
- Keyword (terms that were set manually, such as "Incident", "Hardware" or "Template");

We noticed that users mostly ignored other (project specific) attributes and focused on the mostly common ones. The reason for that was the sparse population of such attributes.

5.2.5 Conclusions of the AWOCADO Research

We built the AWOCADO prototype to gain some insights into issues users have with searching for information in document management systems using metadata. We have introduced extendable medatata types along with the simple ontology in the form of controlled vocabulary. Based on the informal observations, the AWOCADO system has demonstrated that those concepts seemingly do not dramatically enhance the information retrieval activity. The observations have shown that many of the issues, such as using the right term for searching and inability to reuse team member's searching experiences, are still present.

Here we will briefly summarise the main conclusions we can draw from this work and lay out the plans for future investigations.

Firstly, a need for recommending *has been recognised:*

We have concluded that in most of the cases users usually do not know in advance exactly what are they looking for and need some kind of recommendation. It is quite difficult to guess the right attributes, especially for other people's data. The user might never find out that a colleague posted a very similar document into the system. Use of recommender systems has been identified as a solution that could alleviate this well known research issue in the information retrieval and collaborative systems fields. This inspired our further research on the recommender system MAGENTA,

which is described in detail in section 5.2.

The idea of extending document management system with recommender system techniques, such as answering a question "which other documents this user finds interesting", was explored in the Presto document management system (Dourish 1999). The implications of inter-user and inter–group coordination in document management systems are discussed in the works of Ginsburg (2000). Ginsburg finds that efficient access and retrieval is hampered by the lack of coordination while searching and concludes that document management systems must be extendable to allow for collaboration mechanisms between system users.

Secondly, a need for a knowledge-based approach has been identified:

In order to help users to make use of the right attributes values when formulating queries, it would be helpful to somehow identify and organise the information about the existing attribute values in the system. The need to improve information finding using a knowledge-based approach comprising concepts and categorisations behind the documents, brought us to the field of knowledge management, the Semantic Web and Ontologies.

Some initial investigations are presented in the third part of this chapter, section 5.3.

A potential for improving document management systems (or content management systems) with knowledge management techniques was discussed as the document-based corporate memory idea in (Dieng *et al.* 1999). Mika and others (Mika *et al.* 2003) concluded that keyword based searches suffered from similar issues we highlighted. Their study was followed by the development of the On-To-knowledge project[45]. The Ontalk project (Kim *et al.* 2004) also recognises the problems user have in guessing the right metadata and proposes an ontology-based personal document management system to solve them.

5.3 Suggesting Guided Tours in Recommender Systems:

[45] Project Home page. Available from World Wide Web: <http://www.ontoknowledge.org/>

MAGENTA

5.3.1 Research Motivation

This section presents the author's and others initial research results in the fields of recommender systems and guided tours, that were achieved using the MEMOIR framework.

5.3.2 MAGENTA Overview

A sensible approach for mining information on the Web is to exploit knowledge acquired by other users who have previously travelled the Internet or intranet exploring similar or related material. This is the method undertaken by the MEMOIR project and recommender systems in general.

MAGENTA, which stands for **M**EMOIR **A**ssisted **G**uided tours **EN**gineered from **T**rails using **A**gents, is a recommender system that has been built as an extension of the MEMOIR framework (Andric *et al.* 1998; Griffiths & Andric 1998). The motivation behind the MAGENTA project was to construct a guided tour through a *trailbase* (database of trails) or combined trailbases that explore the same or similar topics of interest. Guided tours are known as a suitable and intuitive tool that helps users to overcome cognitive disorientation (De Young 1990) in a hypertext system.

MAGENTA has been integrated into the MEMOIR framework to take the advantage of the trailbase data. It is these trails, which define a record of work or the history of the user, that are important for MAGENTA. Trails within MEMOIR are purposely used to store a list of documents that individual users have decided are related to a specific topic of interest. MAGENTA explores the trailbase to find documents which are related to one another, and uses that information as the basis from which to begin dynamically constructing guided tours on topics of the same subject.

MAGENTA employs agents[46] to dynamically locate relevant Web pages. Agents are used to find Web pages associated with a given trail of documents whose contents

[46] Agents are defined in section 3.5

describe a subject of interest to the user. MAGENTA then presents those related pages to the user as a branching guided tour. The guided tour format was chosen because it provides an intuitive way to present information to users who wish to learn more about a selected topic by following different routes through the tour. Guided tours also have other practical uses within the corporate intranet, which is our main area of interest in this thesis, for example for displaying documents in a user-determined order, a sequence of presentation slides, tutorials or educational tours.

Initially, MAGENTA begins with a trail of documents supplied by the user. The contents of these documents should be related to a topic of interest for that user. MAGENTA then builds a tour consisting of those documents in the original trail with additional documents that are related to the subject of concern combined with the ready-authored tours from the database of tours, called *tourbase*. This is achieved by launching one or more agents that will explore the trailbase to find other documents that are related to those documents in the primary "topic of interest" trail. For every URL in the trailbase that is related to an URL in the "topic of interest" trail, a new branch is dynamically appended to the tour. The appended branch consists of all the documents in the trail where the related URL was discovered. All those documents that the agent has determined to be related to the original "topic of interest" trail, are then recommended to the user, who can examine their contents by using navigation through the tour. We began this process by implementing existing agents in the MEMOIR framework to predict the next steps in the tour.

Two agents we used were *FindCoVisitors* and *SuggestedReading* (Pikrakis *et al.* 1998). *FindCoVisitors* gives a number of visitors of that page and *SuggestedReading* returns a list of recommended documents. MAGENTA operates within an intranet environment as an applet in a Web browser. The example of the MEMOIR applet extended with the MAGENTA user interface is given in figure 5-3.

The visualisation in MAGENTA is achieved by two separate tab folders added to the MEMOIR applet: *Tours* (traversing tours manager) and *Edit* (tourbase maintenance). The list of *Next destinations* represents branches of a tour and the number of co-

visitors is given in brackets. Branching enables grouping of documents – each branch represents a set of closely related documents.

Branching enables users to follow different routes through the tour. Branching also provides additional information to the tour navigator: each branch represents a collection of documents (located down that branch) which are more closely related to one another compared to documents down another branch. Therefore users can navigate down a branch to learn in greater detail about topics associated with the top document of that branch.

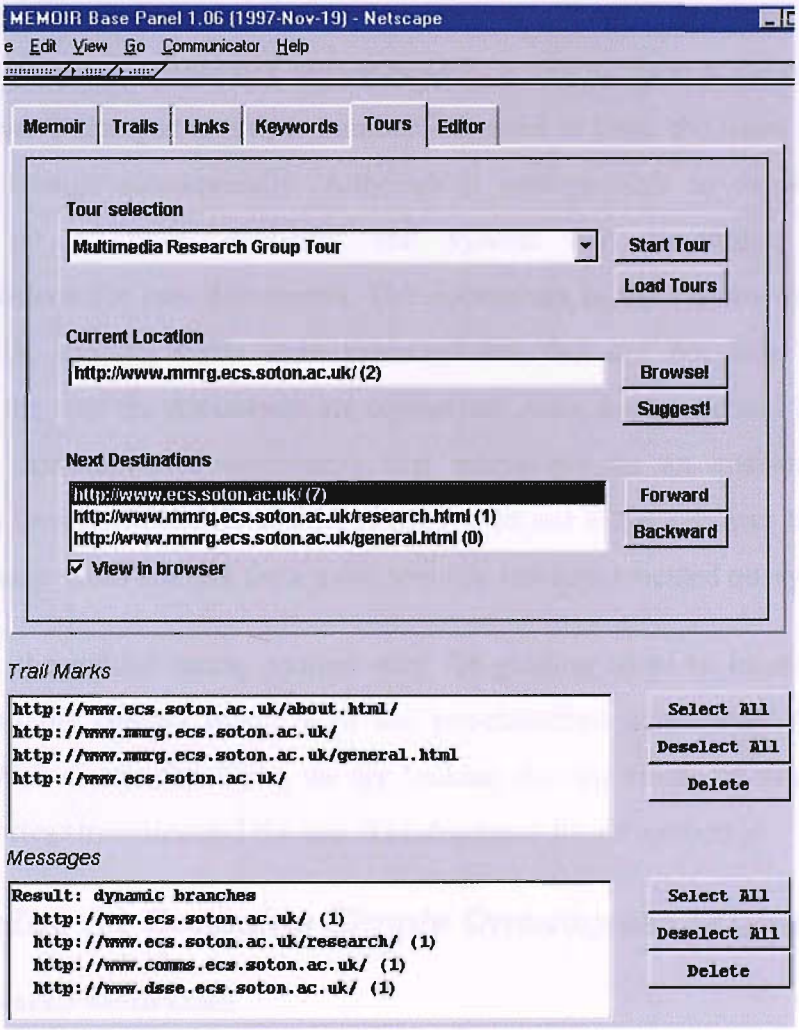


Figure 5-3: The MAGENTA system screen shot example

5.3.3 Conclusions of the MAGENTA Research

MAGENTA proposes dynamic construction of guided tours from trails using agents.

The reason why the guided tour format was chosen, is because this format provides a suitable environment for users to learn more about a topic of interest, hence users are free to easily move forwards and backwards through the related data. The MAGENTA prototype implementation has shown that dynamic guided tours employing agent technology could be easily added to the MEMOIR system for recommending relevant documents. It demonstrated that a guided tour concept can be extended not only to present a manually authored tours, but also to benefit from trails blazed by similar users, using those trails to expand possible branches of a guided tour. However, MAGENTA was limited to recommend as the next step in a guided tour, either a manually set document from a fixed guided tour prepared in advance, or a document that has been viewed by a similar user. If the documents in the system were changed and their number increased in time, the fixed guided tours would not change automatically. Although it was possible to explicitly find a document by a given keyword, the system had limitations in giving recommendations for new documents. The documents in the system were analysed by MEMOIR and keywords were extracted but that did not help the user to understand the way the documents are connected. Also, the user could be presented with many recommended documents that might not be an answer to his/her information needs. MEMOIR and MAGENTA do not allow the user the means to articulate those needs and ask for a more specific and sophisticated query.

Apart from the guided tours, another way for guiding users to locate interesting pages within the desired topic, is to use pre-classified taxonomies of Websites, known as Web directories. Next, we are looking into the issues of visualisation of Web directories structures and the use of zzstructures for navigation.

5.4 ZigZag for Browsing Simple Ontologies: ZZDirectory

5.4.1 Research Motivation

The Web Directories or Catalogues, such as *Yahoo*[47] or Open Directory Project

[47] Yahoo Website Directory, Available from World Wide Web: <http://www.yahoo.com>.

(*ODP*)[48], used to play a significant role in assisting Web surfers in finding the websites associated with a certain topic. Nowadays[49], they are not that popular as they used to be several years ago. Web directories can offer the assistance by exploiting the human trait that sometimes we do not know what are we looking for, but we can recognise it when we see it. Web directories classify a portion of the Web in a hierarchical topic/subtopic organisation and searchers can navigate this hierarchy. However, many Web resources (such as websites or newsgroups) cannot be clearly categorised to belong to only one group or subgroup. Connecting the topics by other, non-hierarchical, side links, then must be introduced. Those side links can be easily overlooked. Therefore searchers might miss an important piece of information about the relationship between topics, and consequently the search results which are associated with the missed topics.

This section presents a novel hypertext based user interface for browsing and navigating Web directories, called ZZDirectory. ZZDirectory builds on the existing XML based implementation of a ZigZag browser developed at Southampton University (Carr 2001a). The motivation of this work was a belief that the employed ZigZag paradigm has the potential to make Web directories popular again by providing a better user experience and improving cognitive orientation while surfing the Web directory as a network of connected topics.

5.4.2 ZZDirectory Overview

ZZDirectory system demonstrates how the Web directory can be browsed using the ZigZag interface. An example data set is taken from Netscape's Open Directory Project, a volunteers' initiative for human-edited Web categorisation used on Google.com[50] until recently. ODP uses Dublin Core of the bibliographic metadata and a custom schema for expressing topic hierarchies. The sample ODP's XML/RDF file, containing part of its Web topic categorisation, was used.

---

[48] ODP home page, Available from World Wide Web: <http://www.dmoz.org>.

[49] In 2006

[50] <http://www.google.com>

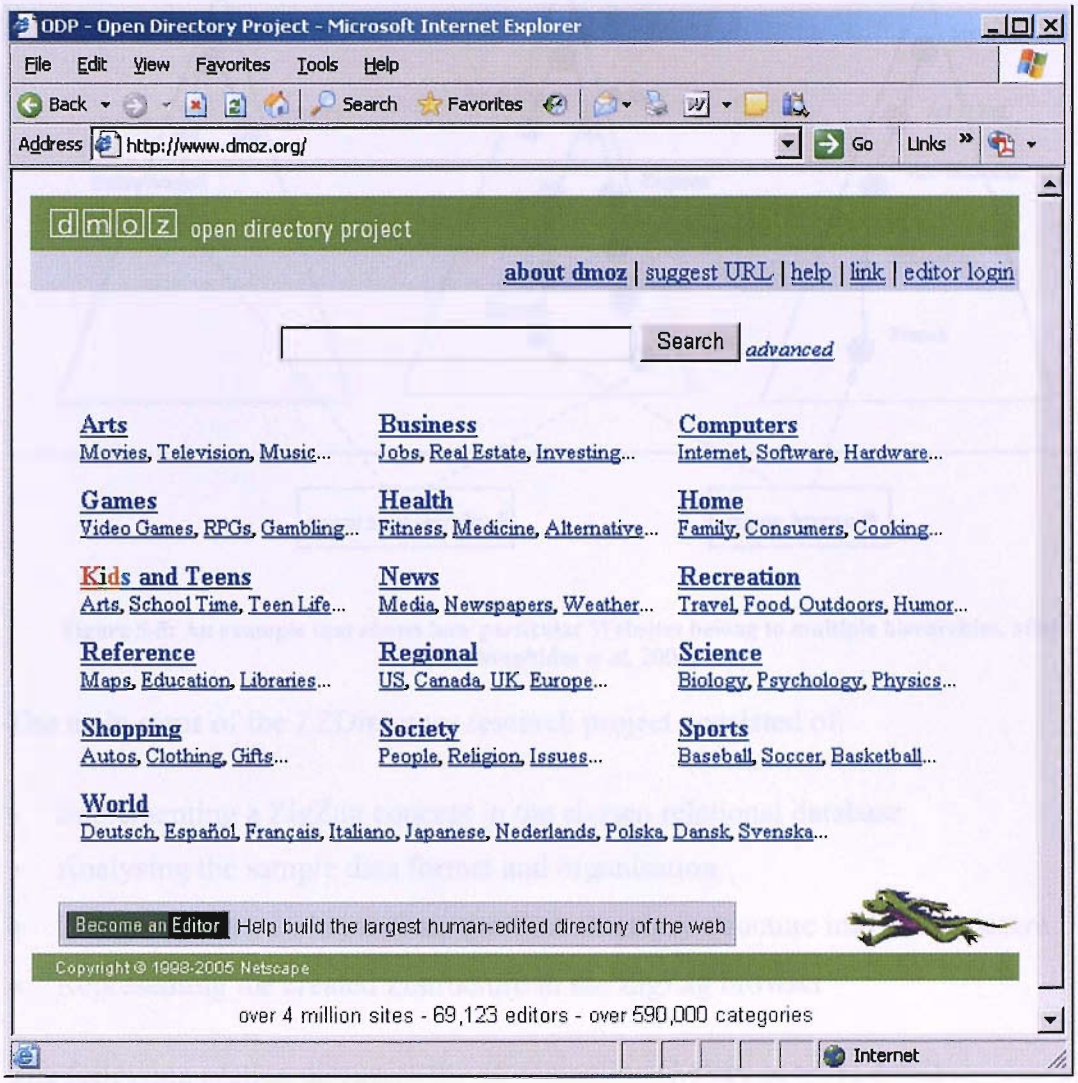


Figure 5-4: The Open Directory Project home page: the initial levels of hierarchy, after <http://www.dmoz.org>

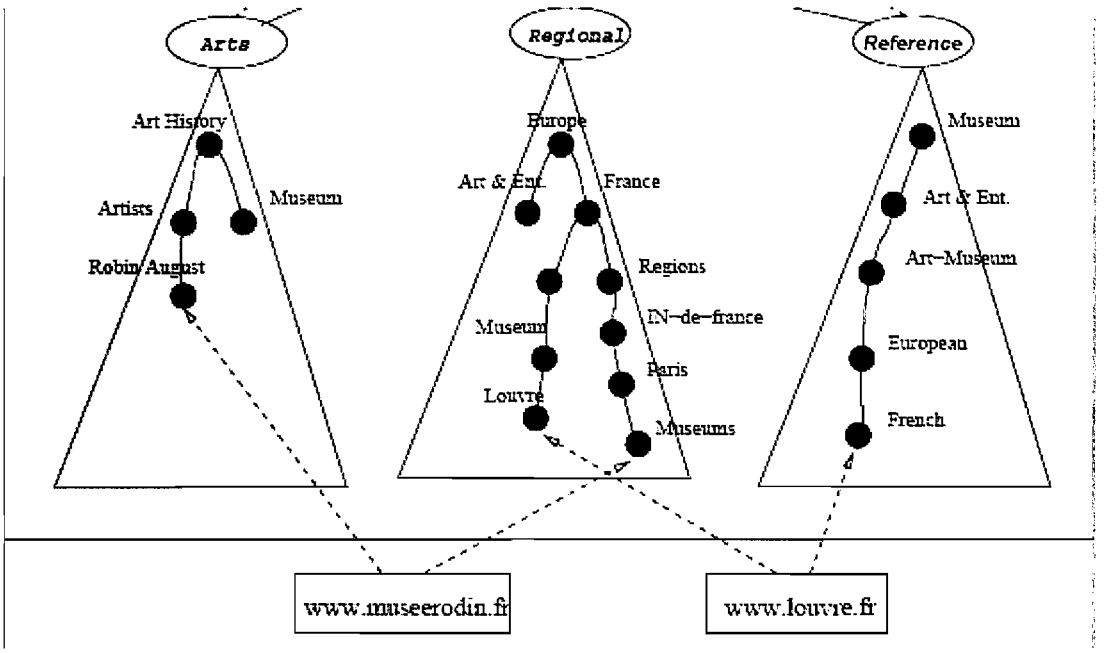


Figure 5-5: An example that shows how particular Websites belong to multiple hierarchies, after (Christophides *et al.* 2004)

The main steps of the ZZDirectory research project consisted of:

- Implementing a ZigZag concept in the chosen relational database
- Analysing the sample data format and organisation
- Designing the translation from the Web directory structure into a zzstructure
- Representing the created zzstructure in the ZigZag browser

The following section discusses the aforementioned steps in more detail.

5.4.3 ZZDirectory Implementation

Although a ZigZag structure would be most naturally represented by a network database model (CODASYL 1971), a contemporary standard, relational database, was selected.

After analysing the ODP data, we have noticed that the topics in ODP are organised in predominantly hierarchical way with the following types of relationships between topics:

- *Narrow*: represents subtopic in the main hierarchy

- *Symbolic*: represents an item from the other main hierarchy place that should symbolically be placed under the current parent
- *Related*: related topics
- *Newsgroups*: related newsgroups

An example of the ODP data in XML format is given in figure 5-6.

```
<RDF xmlns:r="http://www.w3.org/TR/RDF/"
     xmlns:d="http://purl.org/dc/elements/1.0/"
     xmlns="http://directory.mozilla.org/rdf">

<Topic r:id="Top">
  <tag catid="1"/>
  <d:Title>Top</d:Title>
  <narrow r:resource="Top/Arts"/>
  <narrow r:resource="Top/Business"/>
  <narrow r:resource="Top/Computers"/>
  <narrow r:resource="Top/Games"/>
  <narrow r:resource="Top/Health"/>
      ...
  <narrow r:resource="Top/Private"/>
  <narrow r:resource="Top/Bookmarks"/>
</Topic>

<Topic r:id="Top/Arts">
  <tag catid="2"/>
  <d:Title>Arts</d:Title>
  <narrow r:resource="Top/Arts/Books"/>
  <narrow r:resource="Top/Arts/Music"/>
  <narrow r:resource="Top/Arts/Television"/>
      ...
  <symbolic r:resource="Typography:Top/Computers/Fonts"/>
</Topic>
...

<Topic r:id="Top/Computers">
  <tag catid="4"/>
  <d:Title>Computers</d:Title>
  <narrow r:resource="Top/Computers/Hacking"/>
      ...
  <newsGroup r:resource="news:comp.misc"/>
</Topic>
...
</RDF>
```

Figure 5-6: The ODP file example

Each topic is uniquely identified which helped us to perform the translation into ZigZag.

We have chosen a main hierarchy identifier that uniquely identifies the topic, for example *Top/Arts/Book*, to become the content of a ZigZag cell in ZZDirectory. In case of the *symbolic* relationship, the additional information is added, placed in the so-called *cell titles* shown above the cell contents.

The dimensions present in the final ZigZag are, quite naturally translated as:

- Dimension *Top*: the main level such as 'Arts and Business';
- Dimension *Sublevel n*: one for each hierarchy level. There could be several Sublevel dimensions: Sublevel 1, Sublevel 2 etc.;
- Dimension *Symbolic*;
- Dimension *Related*;
- Dimension *Newgroups*.

The translation from the ODP hierarchy to a ZigZag is such that all dimensions originated from *Narrow* actually are of the same type *Sublevel* (semantically meaning *isA*). However they cannot be in the same dimension because a cell cannot have 3 neighbours: sibling before, sibling after and the first child of the sublevel below. That is why each sublevel dimension becomes a new ZigZag dimension whenever the tree level increases. In the example file the hierarchy is only one level deep; therefore there are only one *Top* and one *Sublevel* (meaning *Sublevel 1*) dimensions.

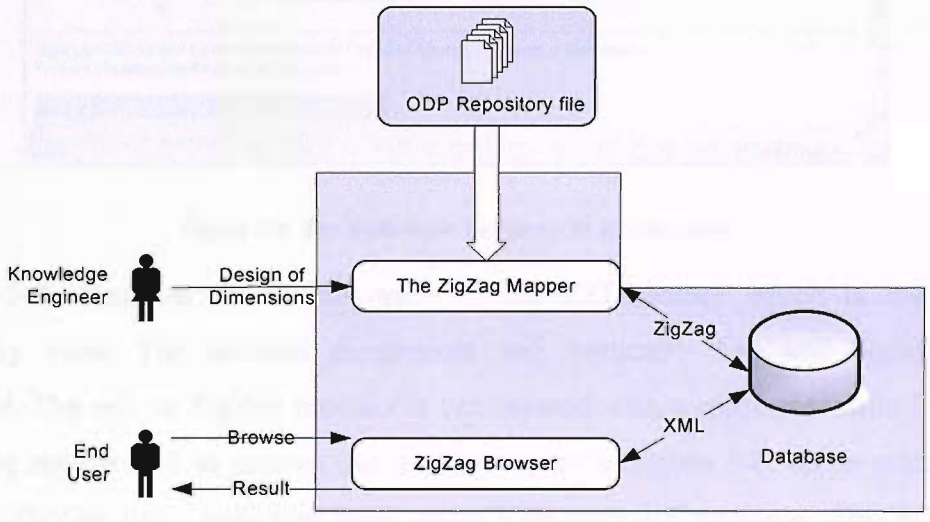


Figure 5-7: The ZZDirectory system block architecture

The architectural diagram of the ZZDirectory system is shown in figure 5-7.

A knowledge engineer designs dimensions and interacts with the ZigZag Mapper component which then reads from the ODP file, creates zzstructures and stores them in the database. The end-user can then browse the zzstructures from the database using a component called ZigZag Browser.

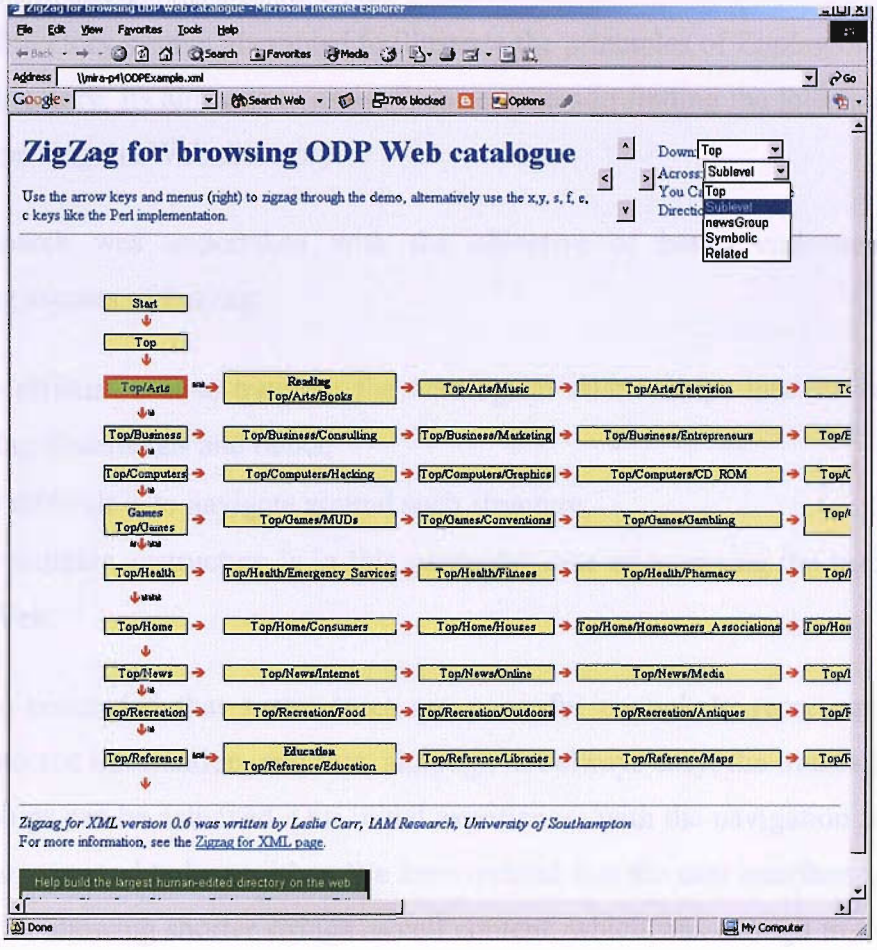


Figure 5-8: The main topic hierarchy in ZZDirectory

Figure 5-8 illustrates the default view in the ZZDirectory which is the main hierarchy view. The selected dimensions are: vertically *Top* and horizontally *Sublevel*. The cell in ZigZag browser is represented with a rectangle while ZigZag links are represented as arrows. As it can be seen in figure 5-8, some cells have several ZigZag links indicated with slanted arrows. For example, *Top/Arts* cell participates in several other dimensions, whose names can be seen by hovering the

mouse over the arrow. The cell *Top/Arts/Book* (next to the current cell) possesses the additional piece of information, show on top of the cell: *Reading*. The cell title *Reading* indicates symbolic relationship to a topic found in the other classification.

The detailed instructions about the prototype usage and a walk through the ZZDirectory demonstration are given in Appendix C.

5.4.4 Conclusions of the ZZDirectory Research

The ZZDirectory system attempted to illustrate the principles of ZigZag on browsing a Web directory. Its aim was to assist Web searchers in finding the interesting topics while browsing the Web directory.

The research was undertaken with the objective of better understanding the following aspects of ZigZag:

- How difficult is it to translate the ontological relationships into the concept of ZigZag dimensions and ranks;
- How difficult is to navigate around such structure;
- How suitable zzstructure is in this particular case of browsing the taxonomy of the Web.

We have concluded that zzstructures are powerful enough to represent complex interconnected information, and that, although not always easy, the translation of the relationships can be achieved. Our initial experiences with the navigation around the zzstructures proved to be positive. We have noticed that the user interface we used is suitable for showing shorter strings as cell content, which was enough in the case of ODP Web topic names.

We have also experimented by using one, two or three ZigZag browser pages embedded in the main HTML frame. Individual pages show different views of the same zzstructure, mimicking the original ZigZag browser design. It turned out that it is the best to have two or even three open views (limited by the screen size). The main one could be always kept to the default top hierarchy view, while different routes through the zzstructure could be explored in the remaining windows.

ZZDirectory's contribution in browsing the Web directories is as follows:

- Explicit showing of the available relationships among the topics brings better understanding of the related categories;
- Context-sensitive revealing of only the portion of the neighbouring topics prevents information overload, while having two or more ZZDirectory windows open at the same time promises to improve orientation in the topic space.

Similar work on Web catalogues was undertaken in (Spyratos *et al.* 2002). The authors analyse relationship types and the structure of the ODP catalogue and they also note the semantic inconsistencies found. Spyratos and others attempt to extract and reorganize index terms to enable personalised catalogue queries. The complexity of the ODP hierarchy is examined also in (Christophides *et al.* 2004). The authors observe that Web resources are described using one or more topics from each facet and that the terms get replicated. The labeling scheme proposed in this work illustrates how taxonomy like ODP must be transformed into a complex graph in order to improve querying by traversal. These conclusions strengthen our belief that ZigZag can improve both visualisation and traversal capabilities due to its inherent simplicity.

Moreover, we believe that employing this kind of solution can contribute to the revival of the existing Web directories. The most likely reason for them to gradually die off is the overhead users experience while navigating their hierarchical levels. Going down or up each level until the desired topic is brought into view, required a lot of cognitive effort from a busy surfer. Providing faster and easier navigation in the networked topic structure could prove to be a method for Web directories revival.

5.5 Summary

In this chapter, the first stage of our research was presented. The early investigation was conducted in three directions, covering the three related research fields introduced in previous background chapters.

5.4.1 Document Management Investigation Summary

Finding and retrieving the relevant documents present a challenge in document management systems. The aim of the research described in this section was to

investigate how flexible metadata management could improve Information Retrieval in Web-based document management systems.

The AWOCADO document management system was designed to provide basic document manipulation, such as creating, viewing, checking in and out, downloading and uploading documents. Also, metadata, or document attributes were introduced. A need for more flexible metadata in document management systems was recognised and our research aimed to fill in that gap. A process of searching for documents using metadata was observed in an informal study.

5.4.2 Recommender Systems Investigation Summary

This chapter also described the early research undertaken by the author and others. MAGENTA, a dynamically branching guided tour generator using agents and trails, has been presented, along with the MEMOIR framework within which MAGENTA was developed. MEMOIR stores past sequences of visited documents (URLs of Websites on the Internet), and those trails are used for recommending next destinations. MAGENTA combines that approach with the idea of static, pre-built guided tours in such a way that a list of recommended, next destinations, comes both from guided tours and MEMOIR trails.

5.4.3 Knowledge Management with ZigZag Investigation Summary

Finally, the explorative research into utilising ZigZag for some aspects of knowledge management was described. Our initial investigation was looking into possibilities of using ZigZag for representing and browsing complex information structures. The ZZDirectory prototype investigated use of ZigZag for browsing ontologies in the form of somewhat complex hierarchies representing Web Directories (Catalogues).

5.4.4 Initial Research Conclusion and a Way Forward

The overall conclusion of the AWOCADO experiment was that the addition of a flexible metadata management into a document management system in itself does not fully benefit the system end-users. The research has highlighted the metadata usage issues and pointed out the need to provide a knowledge-based framework together with the recommendation service, in order to considerably improve searching in

document management systems.

We have then looked into providing users with the guided tour recommendations inside a prevalently collaborative-based recommender system. Although MAGENTA proved that the guided tour concept can be extended, it has limitations in providing recommendation related to inability to suggest new, unseen documents.

The need to overcome the limitations in AWOCADO and MAGENTA, together with the positive initial experiences with ZZDirectory for knowledge representation, led to the second, final stage of this thesis research. The final phase of this work extends the AWOCADO paradigm in order to use a knowledge-based ZigZag supported recommender system for enriching document management systems. The continuation of the research in the area of document management systems is described in the following two chapters, Chapters 6 and 7.

Chapter 6

"à la": Associative Linking of Attributes

Chapter 6 describes the major implementation of the "à la" system, the final stage of the work in this thesis. It presents both the architecture of the "à la" prototype system, and its internal algorithms. Extracting the metadata from a document management system is described in detail. Constructing metadata ZigZag links, as well as using the obtained metadata network for searching, is subsequently elaborated. The chapter contains an example of the end-user interaction, an illustration of how the system can be used to assist searching and browsing document collections. Finally, the related work is discussed and compared with the relevant aspects of the work presented in this chapter.

6.1 System Overview

Our early research, AWOCADO, MAGENTA and ZZDirectory, has motivated the final investigations conducted in this thesis. The user study in AWOCADO revealed the difficulties users have while searching document management systems. We have decided to attempt to improve the user searching experiences in two areas:

- suggesting similar items, such as documents, users and past queries;
- suggesting associated metadata in order to assist query reformulation;

Investigation into ways of suggesting similar items is the main motivation for adding a recommender facility inspired by MEMOIR/MAGENTA into the AWOCADO system. Investigation into discovering the metadata vocabulary and metadata associations is the main motivation of the work described in this chapter. Finally, investigation into the best ways of visualising discovered metadata connections is the motivation for adding a browser component from ZZDirectory to the AWOCADO prototype.

The principal aim of the "à la" project is to improve document sharing in a collaborative environment. Its approach is to use a knowledge-based recommendation service built on top of the legacy document management system. Both content-based and collaborative filtering recommender systems techniques are used together with the hypertextual ZigZag visualisation.

The objective of the system is also to provide the user with better knowledge about metadata links, i.e.: how the attributes that describe the underlying documents relate to each other. Two kinds of users could benefit from "à la":

- An end-user querying a collection of documents in a document management system (or in a closed Internet domain such as intranet);
- A knowledge engineer who mines the data in the system in order to maintain document versions and metadata or model the organisation in order to build its ontology.

The overall aim of the "à la"" system is thus to harness the value of the unstructured information (document content) and the structured information (attributes) in order to promote both human and machine searching and processing within an organisation.

6.2 *"à la" Architecture*

In order to demonstrate the feasibility of the "à la" solution, and subsequently to evaluate its usefulness, an experimental prototype of the "à la" system was built. The following setting was used as the starting point:

- Various artifacts are kept as documents in the document management system. Some items in the repository have readable textual content, while others are in non-textual formats (drawings, video, audio etc.). The system used here is AWOCADO, and is considered representative of a typical document management system.
- The document management system stores user-generated metadata about the documents. These are usually document identifiers, authors, creation/editing dates etc. It is also assumed that the document management system manages users and stores some kind of audit logs on document manipulation such as information about who has viewed or edited a document.

The "à la" prototype system accomplishes its objectives using 9 main components, presented in figure 6-1.

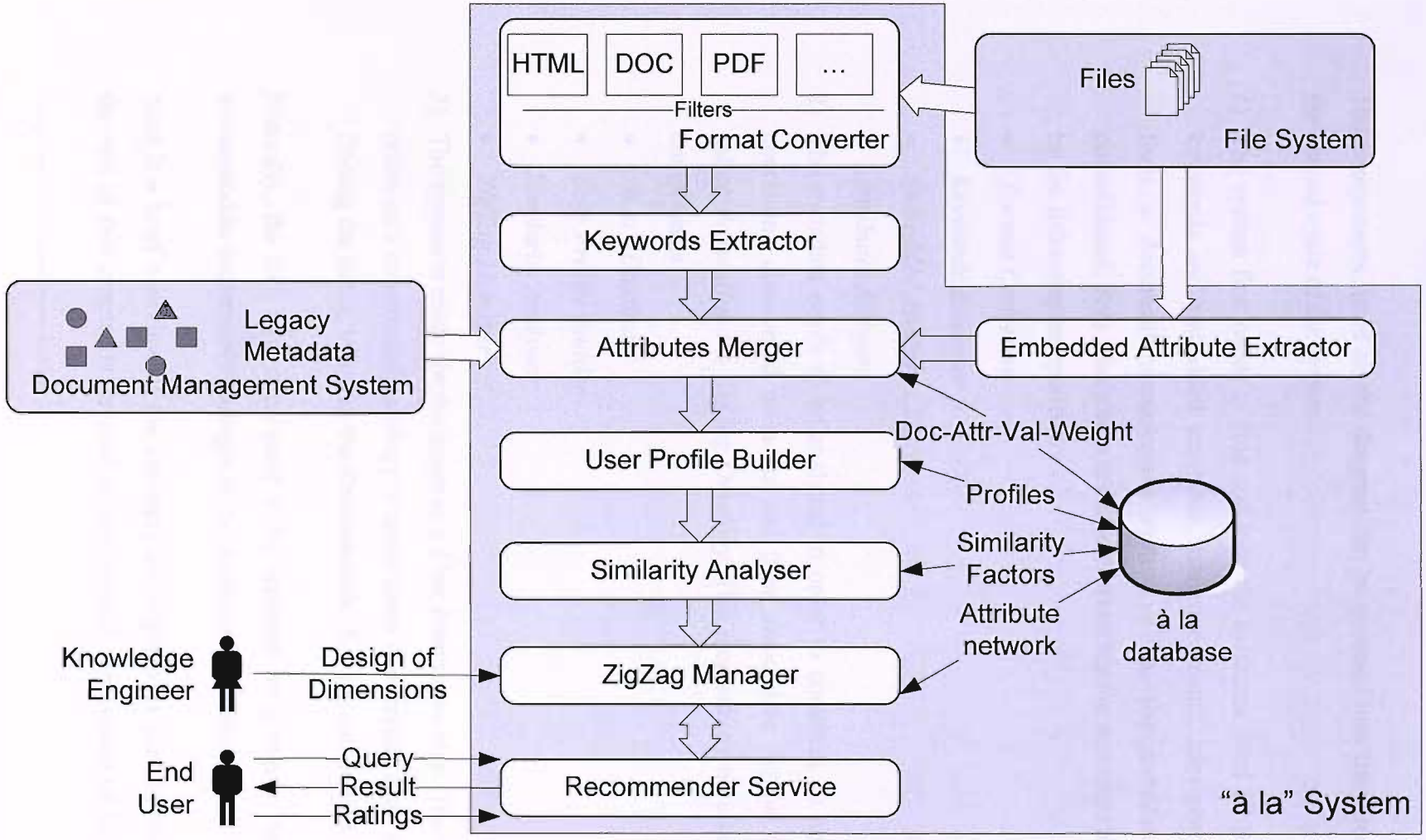


Figure 6-1: The "à la" system block architecture

The components listed on the diagram can be grouped into the three sets following the natural cycle of the system:

1) The system first needs to find and load the metadata from all possible sources: keywords and embedded attributes from documents, user-generated attributes from a document management system. Then, the metadata needs to be consolidated. This whole step is called *Harvesting the metadata* and is performed by the following components:
   - *Format Converter*
   - *Keywords Extractor*
   - *Embedded Attributes Extractor*
   - *Attributes Merger*
2) The metadata needs to be analysed in order to construct an ontology network containing discovered metadata and their associative ZigZag links, in a step called *Populating the ZigZag Ontology.* This step mainly involves the following components:
   - *"à la" Database*
   - *User Profile Builder*
   - *Similarity Analyser*
   - *ZigZag Manager*
3) The system is ready for the usage in a *User Interaction* step. The "à la" prototype relies on a constructed ontology to assist users in everyday tasks of browsing and finding the items, by using the *Recommender Service* component.

Naturally, the first two steps need to be repeated on a regular basis in order to accommodate incremental changes in the document collection.[51]

Next is a brief overview of the concepts and algorithms used in this system, while the rest of this chapter is devoted to the in-depth description of inner workings of

---

[51] Discussion about the frequency of these refreshes is given in section 6.4.4 The ZigZag Manager

each of the mentioned components

The central object in the "à la" system is a *document*. All items that represent artifacts having content in a digital form are considered documents in "à la". Each document possesses an identifier, called a *Title,* which is a piece of metadata capable of implicit linking with the document's body.

A document in "à la" is modelled as a weighted vector of attributes constructed by combining automatically extracted attributes from the document's content with the attributes manually entered by users. The "à la" system user is also represented with a weighted vector of metadata obtained from the documents user has authored, edited or viewed. Users' past searches (queries) are elevated to be first class objects and modelled as a matrix containing information about the used attribute types and the actual query values.

Another type of object, an association between attributes, is also promoted to be a first class citizen. The association or a ZigZag link between two attributes has a type. The ZigZag link type represents a type of the relationship, and it is assigned to a dimension in the ZigZag multidimensional space. The dimensions of this space are created by a knowledge engineer. The knowledge engineer is the one who initiates an association finding algorithm that creates instances of ZigZag links between attributes for the selected dimension.

Also, the ZigZag links are given a weight expressing the strength of the relationship which can evolve with time. The implementation of a zzstructure in this research somewhat modifies the original concept in order to store weights together with ZigZag links.

The following sections describe in detail the system bootstrapping process (harvesting the metadata and populating the ZigZag ontology) and the user interaction.

6.3 Harvesting the Metadata

6.3.1 The Format Converter

Attribute harvesting in the "à la" system starts with the Format Converter, a module that first retrieves content from the document management system. Document files are retrieved either from the local file server or from the Web, in case of locations expressed as URLs. The Format Converter converts the content to a format (HTML or a pure text) suitable for the next module in the workflow, the Keywords Extractor.

6.3.2 The Keywords Extractor

For each Web readable file, a keyword TF-IDF extractor developed in the QuiC project (El-Beltagy *et al.* 2001) is used.

The extractor performs the following document pre-processing steps:

- Document parsing using text or HTML parser
- Tokenization or segmentations into tokens
- Stop word removal using a list of words to be removed
- Stemming, lemmatization based on Porter's algorithm (Porter 1980).

Feature selection, in this case removal of infrequent terms, is performed latter during the building of a zzstructure (see section 6.6.4 "The ZigZag Manager"). Each processed file is initially represented with a vector of weighted terms that are stored in the "à la" database as document attribute of a type *Keyword.* Weights are normalised for a file. For example, after this step, the database of attributes could contain the attributes shown in table 6-1.

Table 6-1 Example of the keyword extraction output

Document	Attribute	Value	Weight
Database Spec-1924	Keyword	requirement	0.45
Database Spec-1924	Keyword	db	0.32
Database Spec-1924	Keyword	batch	0.15
Database Spec-1924	Keyword	view	0.08

6.3.3 The Embedded Attributes Extractor

The embedded attribute is considered to be a file property that is usually managed by the application responsible for creating a certain document format. Embedded attributes are found inside the document content, but they are not supported by all file formats. Embedded attributes can be classified into one of the following types:

- System defined, such as file size or creation date, typically available for all types of documents;
- Standard or predetermined attribute types that we typically find in the Microsoft Office documents (Microsoft 2006b), such as Title, Subject and Author (as presented in figure 6-2);
- Custom or user defined types, such as Client or Audience, again typically found in the MS Office documents;
- Embedded tags, such meta tags in HTML documents or tags in XML.

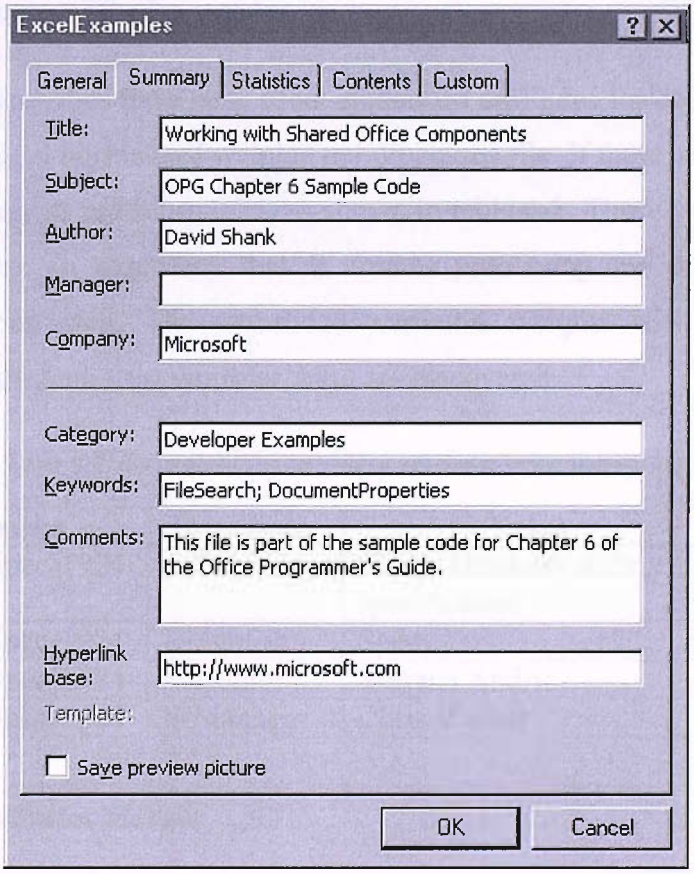


Figure 6-2: Embedded Attributes Example

The Embedded Attributes Extractor can be applied only on a limited number of file formats, which possess a capability of storing the attributes embedded inside the content. Examples are Microsoft Office documents or HTML documents.

Every different file format needs to have its own Embedded Attributes Extractor.

Once extracted this component saves attributes of the each processed document into the "à la" *Document-Attribute-Value-Weight* database.

Attributes harvested by the Embedded Attributes Extractor are not considered very trustworthy: it has been observed that users frequently do not fill those kinds of attributes and instead frequently use a copy of an existing document as a basis for the new document. Therefore we have assumed that embedded attributes are mostly missing or misleading. Meta tags in HTML documents have also been proved to be misleading (Himmelstein 2005). Therefore the importance of the embedded attributes, implied by their attribute weights, must be estimated to be lower than for example importance of the keywords.

The following heuristics have been used: embedded attributes for each document are initially given equal normalised weights per processed file. If there are for example 4 attributes, all weights are set to 0.25, as shown in table 6-2. Those weights might be changed later by an algorithm that is further processing the document in the attributes merging step. The embedded attributes weights will be lowered if attributes of a different, trust worthier, kind are discovered.

Table 6-2 Example of the embedded attributes extraction output

Document	Attribute	Value	Weight
Database Spec-1924	E-Title	CSDB Database technical specification	0.25
Database Spec-1924	E-Subject	Spec	0.25
Database Spec-1924	E-Author	Mirjana Andric	0.25
Database Spec-1924	E-Manager	Chris Walker	0.25

6.3.4 The Attributes Merger

The Attributes Merger first interfaces the document management system in order to gather the user-generated metadata. The module is also responsible for determining

the User-Document history of interactions, if such metadata is not explicitly stored. For example this module would find who the author of the document is, and which other users subsequently viewed or edited it. The user-generated attributes are then saved in the "à la" database with normalised weights, as in the example shown in table 6-3.

Table 6-3 Example of the user-generated attributes extraction output

Document	Attribute	Value	Weight
Database Spec-1924	User	Mirjana Andric	0.33
Database Spec-1924	Team	Database team	0.33
Database Spec-1924	Document Type	Design and Implementation	0.33

When all possible attributes are gathered, the following sets of attributes are merged for each of the documents:

- Embedded attributes for different document versions
- Keywords for different document versions
- User-generated attributes

Each type of attribute is given an initial weight, proportionally higher for a more trusted set, and those weights are finally normalised for the whole document. Heuristics have been adopted so that keywords and user-generated attributes are given the same importance, while embedded attributes are counted as four times less reliable. This is accomplished by adjusting the weights of the embedded attributes before merging. The merging is performed according to the following algorithm:

Step 1. Document descriptor is empty for the document encountered for the first time.
Step 2. Divide the weights for embedded attributes by the factor 4.
Step 3. For each attribute repeat:
  Step 3.1. Check if the attribute exists in the descriptor.
    If it already exists increment the existing weight for the value of the new weight.
    If it didn't exist, add a new entry to the descriptor.
Step 4. Divide the weights with the sum of all weights in order to normalise them.

Figure 6-3: The attribute merging algorithm

The document descriptor obtained by merging can change with time and consequently track how the content or the attributes change. The keywords that repeat in different versions are emphasised by increasing their weights, while keywords that appeared in earlier versions but do not appear in the subsequent versions keep losing their importance, as their weights are continually adjusted in the process of renormalisation.

6.4 Populating the ZigZag Ontology

6.4.1 The "à la" Database

The "à la" system database is modelled using the Entity Relationship model and implemented as a contemporary standard relational database.

As mentioned in the previous sections, the "à la" system database stores *Document-Attribute-Value-Weight* table. Also, it represents zzstructures in a set of tables (cells, dimensions and ZigZag links), using the conceptual data model shown in figure 6-4.

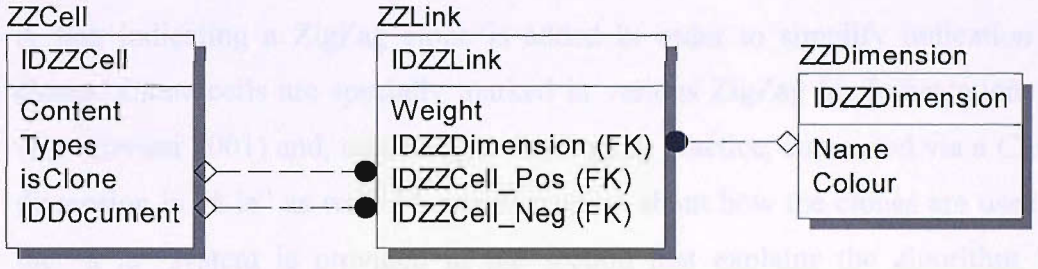


Figure 6-4: The ZigZag data model in a relational database

Cells are represented in a table called *ZZCell* with an identifier (primary key), the content of the cell (string) and the additional fields: *Types* (explained later), *isClone* (indication if a cell is a clone) and *IDDocument*, a reference to the document inside a document management system. Dimensions are stored in the table *ZZDimension* which contains the dimension's identifier, name and the colour by which its ZigZag links are to be visually represented. A table *ZZLink*, corresponds to a link table between the Dimension and Cell and, besides references to them, has only one own field: *Weight*.

The original concept of zzstructures is somewhat extended with the added fields, the

only reason being easier implementation using the relational database paradigm. In each case we shall explain how it can be translated to the pure zzstructure model.

- The cell is assigned a property called *Types*, representing a type of the cell content, for example *Keyword* or *User* (or both!). This additional feature of a cell is introduced only for the convenience, enabling simpler visualisation of this piece of meta-information. The *Types* property can be implemented as a proper additional dimension, having in mind that a cell can have multiple types. However, keeping this degenerated dimension within cell enables us to show the third, mini dimension (having short ranks of typically one to two cells), with the two regular dimensions on the 2D surface.
- The cell possesses an identifier to the underlying object, i.e. document (*IDDocument*), which allows quick linking to the document space. Not all cells would have this feature, only those identifying the object, where the attribute type is *Title*. Again, this feature could have been represented with an additional dimension.
- A flag indicating a ZigZag clone is added in order to simplify indication of clones. Clone cells are specially marked in various ZigZag implementations as well (Ervasti 2001) and, according to the ZigZag practice, connected via a *Clone* dimension in "à la" as well. More information about how the clones are used in the "à la" system is provided in the section that explains the algorithm for creating zzstructures, 6.4.4. The ZigZag Manager.
- The ZigZag link has been given a weight, besides reference to a dimension and participating cells (positive and negative). This feature enables the practical usage of ZigZag for an ontology network, in order to indicate strength of a certain ZigZag neighbouring association between cells. Again, there is no conceptual reason why this weight could not be simply assigned to another *Weight [DimensionX]* dimension. Also there is no conceptual need for a ZigZag link to be addressable, first class object – this is modelled here this way in the spirit of the relational model.
- The dimension has been assigned a name and a colour with which its ZigZag links should be presented. This is a quite natural implementation of a dimension,

enabling easier visual distinction between ranks belonging to different dimensions.

6.4.2 The User Profile Builder

The User Profile Builder creates users profiles and tracks users preferences. Its responsibility includes analysing user behaviour, i.e. past users queries and explicit interests in a document, indicated by explicit ratings (positive or negative).

Users are able to rate the documents in the search result and more about this feature will be described in section 6.5 User Interaction. However, a rating is not required, and if not present, a user profile is generated implicitly.

The user descriptor in "à la" is in a form similar to a document descriptor: the user is represented with a weighted vector of attributes that describe the user's interests. The user profile is stored in the "à la" database component.

The User Profile Builder mines the audit facilities of the document management system in order to find out who is creating, editing and viewing documents. Vectors of terms representing these documents are then superimposed on each other to create a dynamic user profile. Weights of the resulting, averaged vector are adjusted when a new vector is added to the profile. The changing of weights depends on the type of event (create, edit or view) and occurs when a user manipulates the observed document.

It is assumed that the creation of the document presents the strongest association between a user and the document terms. Since the user's interests do not usually become quickly obsolete in a closed domain environment, such as a company intranet, the decaying of the weights for terms in the profile was not implemented, leaving it as a task for future work.

6.4.3 The Similarity Analyser

The Similarity Analyser computes the similarity factor between two requested documents or user profiles. The cosine similarity (Salton 1989) as implemented in *Linking in Context* project (El-Beltagy *et al.* 2001) is used to produce a similarity

factor.

The similarity factor represents a number in a range of 0 to 1 indicating to what degree two observed documents have similar metadata and the corresponding metadata weights. A factor of one means that a pair of documents has exactly the same vector representations, the same metadata, and exactly the same weights each document. A factor of zero means that there is no single piece of metadata that is common to the pair of observed documents. The similarity factors are stored in the repository ("à la" database component) for later usage in building the similarity dimensions (see next section).

6.4.4 The ZigZag Manager

The ZigZag Manager is the central "à la" component. It takes document attributes, analyses them using the information which document they belong to, and establishes attribute ZigZag links. ZigZag links between attributes are calculated and loaded into zzstructures. The zzstructure construction contains three sub-steps:

- Creating Cells: Identifying the unique ZigZag cells
- Constructing Dimensions: Creating temporary many-to-many relationships for a combination of two attribute types. For example *User-Keywords* combination: there can be many keywords used by an instance of a *User* attribute and also many users could utilise the same keyword. The combination of two attributes types is what is to become a dimension in the ZigZag space, representing a relationship between a pair of attributes.
- Creating ZigZag Links: Computing the final zzstructure links for each dimension

The building of a ZigZag structure starts with populating its cells. The ZigZag rules assume that content inside a cell does not repeat in another cell, i.e. cells are unique. This is a feature that can be utilised to identify and place in the same cell a string that seems to be appearing under different attributes. For example, a *team* name can also feature as a *keyword* and we want to mine that kind of associations and store in a designated dimension.

In order to create the unique ZigZag cells the algorithm has to pass through all stored

attribute values from the *Document-Attribute-Value-Weight* table and recognize the unique values. At the same time, one special dimension, degenerated dimension *Types*, is populated with the information about the attribute type, such as *User* or *Title*. In most cases the ranks of this special dimension would be very short, consisting of only one attribute type. However, in some cases two or more attribute types would have the same value and therefore will be represented with a singular ZigZag cell.

When unique cells are identified, the next step is building the dimensions and connecting cells into appropriate ranks along dimensions. The process of initiating dimension building is a human driven and a domain ontology is used as a guideline for determining and naming dimensions.

The domain ontology needs not to be specially designed; it can be naturally deduced from the document management system in the following way. During the document management system setup, it is quite common that the knowledge engineer determines, or decides, which attribute types will exist. By doing that, the knowledge engineer designs (in a more or less formal way) an ontology containing the most important domain concepts. A concept of a document and each determined attribute type form this ontology.

This ontology always has a concept of a *Document*, and commonly a *Document Class*. Quite frequently the concepts of the *Creator* and *Creation Date* exist as well, both of which are connected to a document with the appropriate relationships, for example *CreatedBy* and *CreatedOn*.

Each of the attribute types participates in a simple 'star like' relationship with a document entity. That is a domain model that can be reused in the "à la" solution. The knowledge engineer starts from a central document entity and selects the most interesting attributes from the pool of the available metadata in the document management system. In most of the cases, this choice of attributes contains metadata that resembles the Dublin Core ontology set (Dublin Core 2004). Ultimately, the ontology used for the further link analysis is influenced by a set of already available attributes if a legacy system is analysed.

An example of such ontology is shown in figure 6-5. This particular example is designed in order to illustrate the principle of the "à la" research idea and is later used in the experimentation on the data collection for evaluation purposes (see Chapter 7).

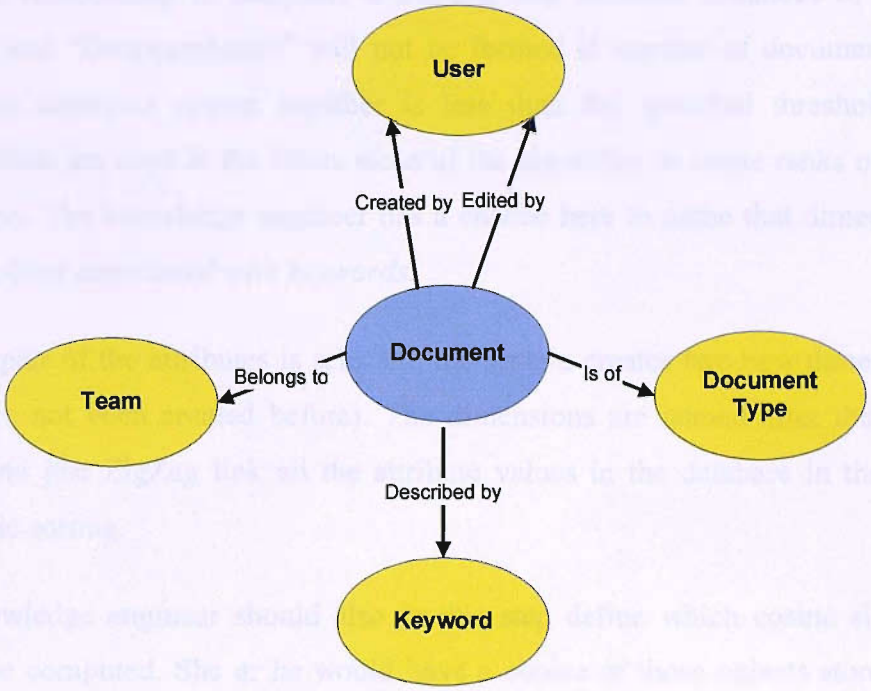


Figure 6-5: The initial attribute-document relationships in the zzstructure building case

The example presented here contains a document concept with a four simple attributes related to a document using 5 relationships. The *User* attribute represents a user's name or ID, as kept in the document management system. *Document Type* (class of a document) and *Team* (name of the team that owns the document) in the case of document management system in the "à la" experiment, take values from a controlled vocabulary, while *Keyword* can take any value.

Guided by the knowledge engineer, a task of "à la" is to analyse the existing data and create new ZigZag links between instances of different types of attributes. The knowledge engineer selects a pair of attributes and the system algorithm establishes many-to-many links between different values of analysed attributes based on a fact that pair of attribute values shares common documents. Statistical collocations are thus identified and links between attribute instances are created.

The potential number of links is huge, and some pruning needs to be employed. The algorithm can use a certain threshold of number of occurrences for common documents (k, where in the default case k=1). A ZigZag link between the pair of attribute values is not formed if a threshold is not reached. For example if a *User-Keyword* relationship is analysed, a ZigZag link between instances of "Mirjana Andric" and "Datawarehouse" will not be formed if number of documents where these two instances appear together is less than the specified threshold. These ZigZag links are used in the future steps of the algorithm to create ranks of the new dimension. The knowledge engineer has a chance here to name that dimension, for example *User associated with keywords*.

When a pair of the attributes is selected, the system creates two new dimensions (if they have not been created before). The dimensions are named after the attribute names and just ZigZag link all the attribute values in the database in the default, alphabetic sorting.

The knowledge engineer should also in this step define which cosine similarities should be computed. She or he would have a choice of those objects stored by the User Profile Builder as vectors: documents, users and queries, and name the dimensions accordingly. For example in this case *Similar users* and *Similar documents* were selected.

To summarise, the following dimensions are created and stored in the ZZDimension table by the "à la" algorithm:

- An attribute *types* dimension showing which attribute types are associated with each value;
- A dimension with only one rank connecting all values of the same attribute type;
- Dimensions for each selected attribute pair representing the relationship between the two attributes;
- Similarity dimensions;

- Clone dimension.

In the process, the knowledge engineer could derive the relationships as shown in the example in figure 6-6. Each arc in the picture corresponds to a ZigZag dimension.

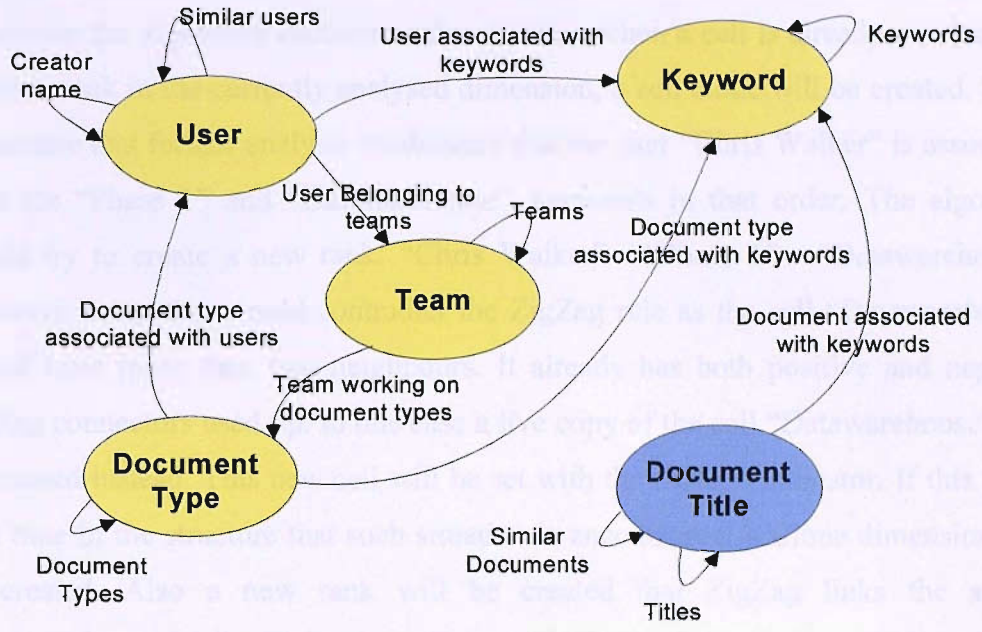


Figure 6-6: An example of the derived ontology in the zzstructure building case

Now that dimensions are determined and relationships between attributes established, the final step of the algorithm remains – to create valid ZigZag links and populate ZZLink table.

Because of the ZigZag "up to 2 neighbours" restriction, the initial many-to-many links need to be rearranged. This is achieved by breaking the existing graph where attribute values are nodes, into typically two or more 'ZigZag well-formed' graphs, using clones to represent cells which have already 'used-up' the number of allowed neighbour connections (positive and negative).

For example, let us assume that the *User Associated with Keywords* dimensional analysis conducted between *User* and *Keyword* attributes, established that the user "Mirjana Andric" is associated with "Datawarehouse" and "Specification" keywords. According to the algorithm, the three cells are created and ZigZag linked

into a rank belonging to then mentioned dimension: "Mirjana Andric" - "Datawarehouse" - "Specification". The rank is formed by connecting the head cell, user name, with the most frequent keyword, which is later connected with the less frequent keyword and so on.

The algorithm then continues with creating further ranks in the same dimension. Whenever the algorithm encounters the situation when a cell is already a participant in some rank in the currently analysed dimension, a cell clone will be created. So let us assume that further analysis established that the user "Chris Walker" is associated with the "Phase 1" and "Datawarehouse" keywords in that order. The algorithm would try to create a new rank: "Chris Walker" - "Phase 1" - "Datawarehouse". However doing that would contradict the ZigZag rule as the cell "Datawarehouse" would have more than two neighbours. It already has both positive and negative ZigZag connectors used up. In this case a live copy of the cell "Datawarehouse" will be created instead. This new cell will be set with the isClone indicator. If this is the first time in the structure that such situation is encountered, a Clone dimension will be created. Also a new rank will be created that ZigZag links the source "Datawarehouse" cell with its clone. If the rank already existed, the new clone would be appended to the end of the rank and tied up to the previous clone. The system would then take care in the future that any changes in the source cell or any clone cell are reflected on the whole rank.

The complete algorithm for creating zzstructures is laid out in figure 6-7.

Step 1. Create Types dimension

Step 1.1. Create unique cells for each value in the Document-Attribute-Value-Weight table.

Step 1.2. Create a cell for each attribute name.

Step 1.3. Pass the Document-Attribute-Value-Weight table and create a rank for each value by using a value as a head cell and by attaching an attribute name cell to it. (Ranks for this dimension are simulated in "Types" field of a cell as comma separated list of attribute names.)

Step 1.3.1. If the rank with that value already exists just attach the attribute name to the end of the existing rank.

Step 2. Create attribute-pairs dimensions (repeat per each knowledge engineer input):

Step 2.1. Receive input for the dimension details: name, names of the parent-child attributes, colour, threshold.

Step 2.2. If the dimension with the parent name doesn't already exist: create the parent dimension rank by connecting all distinct parent instances ordered alphabetically. Use combined (summed) weights for all documents containing an attribute in order to assign a weight to a negative link of each attribute.

Step 2.3. Repeat the previous step for the child attribute.

Step 2.4. Create the parent-child dimension.

Step 2.4.1. For each distinct parent process the many-to-many links:

Find all the children (where the parent-child appear in the same document) by analysing Document-Attribute-Value-Weight table

Order the children by the decreasing weights (use summed weights if encountered in more than one document).

Create rank for the parent-child dimension by connecting parent as a head cell to the ordered list of children. Use children's weights for the negative links.

Whenever it is encountered that the cell is already in some other rank: create its clone and add the clone to the rank. Add the clone to the end of its original cell's rank in the Clones dimension as well.

Step 3. Create similarity dimensions (repeat per each knowledge engineer input):

Step 3.1. Receive input for the dimension details: name, names of the objects for which to compute similarity, colour.

Step 3.2. If not already existing, create cells to represent the given objects (*Title* represents documents, presumably already in the zzstructure).

Step 3.3. Compute mutual similarity factors and place in the matrix.

Step 3.4. Create the similarity dimension and use the many-to-many links in the similarity matrix to create zzstructures in the same manner as in step 2.4. Use similarity factors for link weights.

Figure 6-7: The algorithm for creating zzstructures

Finally, the frequency of updating the metadata index (in the form of zzstructures) is an outstanding issue. Desktop search tools face a similar challenge. In some cases they run update, triggered by a manual or scheduled task, and in some cases a monitor process runs whenever a monitored files change (Buckley 2005). The best solution obviously depends on the expected volume of change and the processing power available. "à la", being a prototype system, opted for the simplest, manual re-indexing of changed versions of documents.

6.5 User Interaction

6.5.1 Recommendation Generation

The system can initiate the interaction with the user through its Recommender Service in the following three ways:

- When the user logs in, a daily recommendation, based on his/her profile, is generated. The system uses a deduced query, assembled from a list of terms in the user profile limited to the top five terms.
- When users are browsing a document from the repository, they can request to see similar documents. The user query is assembled from the current document profile.
- The users can submit a free text search query.

Having a query defined, the "à la" system consequently employs the algorithm as shown in figure 6-8.

To summarise, the algorithm is trying to find the best matching ZigZag cells while performing a simple ontological network traversal. All the documents found along the way are included in the result, which can be later personalised to suit the user's profile. Hence, a query is expanded by following the associative ZigZag links.

6.5.2 User Interaction Example

Figure 6-10 shows the "à la" user interface after the user has submitted an initial free text query and performed some actions that are going to be explained in this section.

In the example shown in this figure, the user is looking for more information about the usage and setting of "LDAP[52]" among the available artifacts.

The user starts by typing a desired string, "ldap" and by pressing the *search* option on the menu bar on the far left. The screen representing a search result is divided into two parts. The left side of the page lists the suggested metadata and documents while the right side presents the browsable ZigZag neighbourhood of the first query term.

Step 1. Parse the user query as in the TF-IDF algorithm and obtain all stemmed terms.

Step 2. The result set containing {Cell content, Cell types, Dimension name, Importance Level} is initially set to empty.

Step 3. For each stemmed term repeat:

- **Step 3.1.** Search for a match in zzstructure cells where content contains the term. Add found cells, their types and the default importance levels of 0.1 to the result set (importance is set to default, to be adjusted later and to influence the final item ordering). Whenever the same cell or its clone is encountered again, accumulate its importance.
- **Step 3.2.** For each dimension in which found cells participate, repeat:
  - **Step 3.2.1.** Find immediate positive and negative neighbours on all ranks for all clones. Add cells, dimensions and the link weights as importance levels to the result set. If the cell that already exists in the result is found again, just accumulate the importance level by adding a link weight to it.
  - **Step 3.2.2.** Find head cells, i.e. rank beginnings. Add them to the result set using the default importance.

Step 4. Divide the result into two groups, one representing documents, and the other representing other metadata. (Cells containing the non-null IDDocument reference are representing documents).

Step 5. Order the entries in the document group by the importance.

Step 6. Optional step: Remove from the result set those documents that have similarity with the current user profile, which is less than a set threshold.

Step 7. Remove all but top *n* documents (n is set to be 30 for the free text search query or 3 for other types of queries).

Step 8. Cluster the metadata group by dimensions and retain in each group only top n (n is set to be 3) items, ordered by importance.

Figure 6-8: The algorithm for querying "à la"

[52] Lightweight Directory Access Protocol – open network protocol for accessing information stored in a directory services server (Bambulis *et al.* 1993)

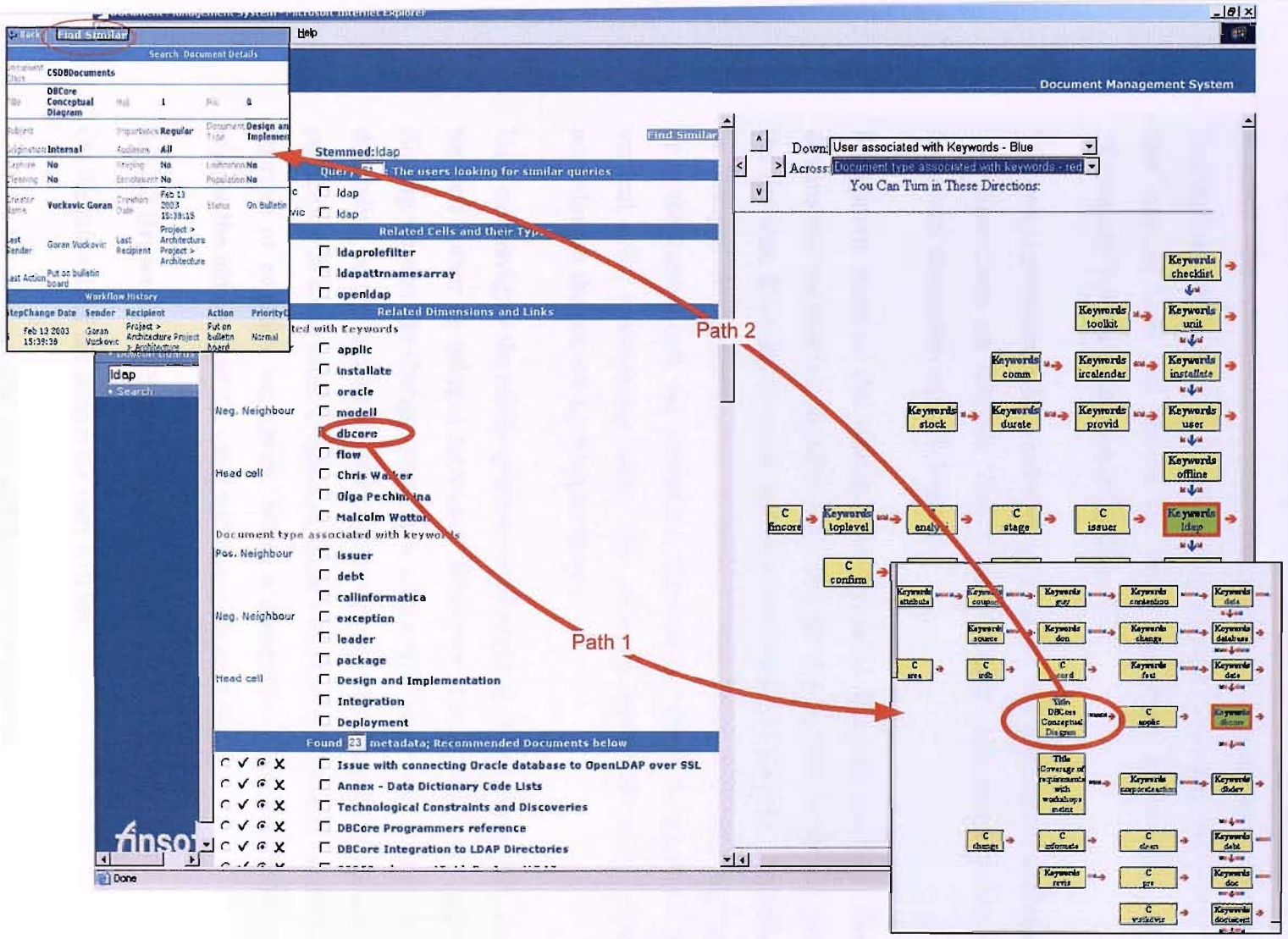


Figure 6-9: The "à la" system user interaction example

6.5.2.1 Browsing the Ontology Network

The browser receives the appropriate ontology network segment in the form of dynamic XML data obtained from the database. In the example on the figure 6-9, the ZigZag browser in initially positioned on the cell containing the query, "ldap". The first line in each cell shows the short dimension *Types* (such as "Title" or "Keywords") or the indication of a clone, "C".

The user is presented with a network of cells and two dimensions to choose at a time. The dimensions are labelled "Down" and "Across" and represent the vertical and horizontal dimension in the 2D space.

Navigation starts at the current cell, which is specially marked. The names of dimensions are ordered alphabetically and the starting one is initially determined as the first one from the list in which the current cell participates. The dimensions can be changed during the session. If a selected dimension is having ZigZag links *from* or *to* the current cell, the connected cells will be shown as arrays of horizontal or vertical cells, representing ranks. The cells are connected using coloured arrows according to the pre-set dimension colours.

User can navigate the ranks up/down or left/right by using the buttons situated in the top left corner or using a keyboard. Whenever user changes the current cell, the ZigZag view might change: some new cells might be revealed, some old hidden, all depending on the current position and the two selected dimensions. If any of the cells participate in some other dimension, there will be a hint, indicating the existence of a positive or negative neighbour. When a dimension is changed, the new one will replace the old one and the view will change accordingly.

6.5.2.2 Browsing the Search Result

On the left side of the screen the user will see:

- a list of suggested users and their similar queries;
- a list of suggested associated metadata, grouped by dimensions and showing

up to 3 neighbours in the positive, negative directions and headcells;

- titles of the recommended documents.

The left panel showing metadata suggestions represents flattened and shortened (up to 3 cells) alternative view on the cell neighbourhood in the ZigZag browser on the right. The difference here is that all visible and invisible dimensions are listed. Neighbours in the most important ranks, including those in which clones are participants, are listed as well.

Each piece of metadata is itself a link into the ZigZag browser on the right side, which navigates the ontology network to the requested cell. In the example on figure 6-9 user was initially searching for the term "ldap". Results show that there are several documents mentioning "DBCore", which appears as one of the suggested cells as well. The user has received a suggestion that the in the dimension *User associated with keywords*, a keyword "dbcore" seems to be associated with the keyword "ldap". This means that some user was authoring a number of documents that each had these two keywords.

The user then decides to navigate to the term "dbcore" in the ZigZag ontology network and follows the Path 1. Let's assume that user clicked on the term "dbcore" and thus navigated to the different place in the ZigZag browser in the right panel. A portion of the ZigZag network is superimposed on the screenshot in the right bottom area, for the purpose of illustration. We can see on the second ZigZag network that some cells represent documents – those of a type *Title*. By changing the dimension "Across" to become *Document associated with keywords*, the user discovers that "dbcore" is a keyword of the document entitled "DBCore Conceptual Diagram", by finding it connected on the *Document associated with keywords* dimension.

The example further shows that by clicking on such cell, the document window will be opened as shown in the example Path 2. The "Search Document Details" page superimposed on the top left area shows the document attributes and allows for downloading of its content.

The user can open that document and then, for example after finding that this is an

important LDAP integration module, proceed by asking for similar documents, or by returning to the initial query result and repeating a cycle.

If the cell in ZigZag on the right hand side was not a document, clicking on it is treated as a new query, the result recalculated and presented in the left panel. This enables user to explore the search result and to easily browse the document and metadata space together.

6.6 Comparing "à la" to the Related Work

6.6.1 Semantic Metadata Layer

A set of metadata with some sort of semantic structure (e.g. an ontology) which is used to describe documents in a hypertext system or a document collection, forms a layer of descriptive data referred to in this thesis as Semantic Metadata Layer. In other words the semantic metadata layer facilitates expressing explicit semantic relationships between the underlying documents.

There are numerous examples of attempts to use a semantic metadata layer on top of a hypertext system or a document collection. The "à la" system draws inspiration from many of them and applies or extends mentioned approaches to its metadata management.

One of the early hypertext projects TEXTNET (Trigg & Weiser 1986) explored the idea of imposing layers of semantics on top of the hypertext system. The aim of TEXTNET was to provide navigation and retrieval based on the meaning of related concepts. The similar approach was undertaken later in SNITCH (Mayfield & Nicholas 1993), where a semantic net was placed on top of a text corpus in order to support content-based hyperlinking.

Two-layer architecture for hypertext documents proposed by Bruza (1990) comprises a semantic index space of so called hyperindices. Hyperindices form the top layer while the hypertext content forms the bottom layer. The hyperindex consists of a set of indexes linked together. The user can navigate either the index or the underlying content space (*hyperbase*) by "beaming up and down". Bruza named

the process of navigation though the hyperindices and retrieval of information from the hyperbase "Query by Navigation".

Cunliffe and others (1997; 2000) used a similar approach. They noticed that associating metadata with hypertext nodes could complement the representation of semantic relationships expressed by links. The adopted approach proposes to accomplish navigation by using semantically indexed and computed retrieval links. A study in (Tudhope & Cunliffe 1999) discusses lexical and semantic approach to finding the possible relationships between index terms. "à la" as well attempts to mine the relationships between index metadata. Enhancing thesaurus relationships to improve retrieval was discussed by the same group of authors in (Tudhope *et al.* 2001). Visualising relationships of terms in a thesaurus using hypertext representation was introduced in (Bosman *et al.* 1998). The aim of the visualisation is supporting the user search. Bosman and others follow the approach similar to Bruza's query by navigation in order to carry out the search.

The metadata index space constructed by mentioned methods can be very large and can feature complex interconnections. The system called Pathfinder (Chen 1999) uses LSA for building such a space from a collection of documents. This approach resembles the building of metadata layer method in "à la". Pathfinder copes with the complexity of the obtained metadata network in the following way: It uses additional techniques for reducing the number of relationships, in attempt to preserve only the most important relationships. The technique in principle works well; however there were some visualisation problems as reported in (Hughes 2003). Hughes uses the PathFinder's document analysis (document and keyword relationships) to build a contextual hypermedia link service CA-DLS (Hughes 2003). It might be beneficial replacing the "à la" relationship extraction module with the PathFinder's. This would enable us to investigate of how zzstructures could be built from the relationships discovered by this method.

The issue of the real time management of the extracted keywords was addressed in a system called WordSieve (Bauer & Leake 2001). WordSieve builds on a technique similar to TF-IDF. It manages 3 layers of keywords using dynamic lists. The first

layer contains a list of the most frequent terms. The second keeps a ranked list of keywords that are appearing in sequences of documents. Keywords are ranked higher in that list if they appear in the first layer. The third layer maintains a list of words that are disappearing from the second list. This method allows for tracking the current topic of the document user is looking at, and for detecting the context changes.

On the topic of embedded attributes extraction - Metadata Miner Catalogue[53] supports embedded metadata extraction from a range of formats and also converts to Dublin Core RDF, similarly to the open source Microsoft Office RDF Extractor[54].

The VKB (Visual Knowledge Builder) system from Texas A&M University proposes using the hypertext system for knowledge representation similarly to "à la" intention (Shipman *et al.* 2001; 2002). The VKB suggestion agents provide task assistance to users for managing their information workspaces by suggesting items such as attribute types, values and relations. Objectives of such a system are somewhat different than what "à la" is attempting – to support information seeking in a document management system.

6.6.2 Overlaying Metadata on the Web

Ideas of combining formal semantics specifications (ontologies) with hypertext and applying them on the Web using the emerging Semantic Web techniques, fuelled much interest in the recent years.

The OntoSeek project supports content analysis of Web-based yellow pages and on-line catalogues (Guarino *et al.* 1999). It makes use of the semi-automatic construction approach in which users verify links found by analysis.

Uniting ontologies with hypertext into ontological hypertexts was attempted in the ESKIMO project and the OntoPortal initiative[55] (Kampa *et al.* 2001; Carr *et al.* 2001c; Miles-Board *et al.* 2001). Its aim was to spread a meta layer of semantic links

---

[53] <http://peccatte.karefil.com/software/Catalogue/MetadataMiner.htm>

[54] Microsoft Office RDF Extractor, Available from World Wide Web: <http://www.ldodds.com/projects/>.

[55] Ontoportal Home, Available from World Wide Web: <http://www.ontoportal.org.uk/>.

over the unlinked Web resources which are found to be related. Ontology was used to provide the structure and understanding of relationships, while the hypertext system was providing the linking mechanism. This is similar to "à la" in a sense that the links between the hypertext nodes (ZigZag links between attributes in "à la") represent candidate ontological relationship instances. In Ontoportal, the meta layer of ontology concepts and their relationships is projected over the existing Web pages in a similar manner as "à la" network overlays a document collection.

The On-To-Knowledge project (Kietz *et al.* 2000; Fensel *et al.* 2000) attempts to semi-automatically build an ontology from a corporate interanet. The terms (concepts of the ontology) are extracted from textual resources using a variety of techniques, in a multi-step process involving a human ontology engineer. Concepts are extracted in the first step, similar to terms in "à la". Further on, concepts are included into a taxonomy that extends a core ontology, while "à la" keeps the original collocation term relationships. To enable constructed ontology to be focused on the domain, On-To-Knowledge removes general concepts. In contrast, the "à la" approach relies on the TF-IDF weights to lower the significance of the non-specific terms. On-To-Knowledge mines conceptual relationships using frequent correlations of concepts within the sentence whilst "à la" looks at the terms mentioned together in the whole content of a document.

The project "Linking in Context" from University of Southampton explores an idea of reusing manually authored hyperlinks for similar Web pages (El-Beltagy *et al.* 2001). "à la" uses the keyword extracting, document modelling and similarity components developed in this project. It also extends the idea of reusability of manually authored links on the metadata-document "links", i.e. identification bonds. If metadata was explicitly assigned to the document by the users, it is assumed in "à la" that the same piece of metadata can be considered associated with similar documents.

The system called WebTop (Wolber *et al.* 2002) provides personal information space management. It finds related pages on the Web, local files and directory information. WebTop employs multiple techniques to establish document relationships, including

some document analysis and passing information to the Web search engine Google. WebTop is similar to the "à la" system in some ways although it does not provide support for the group of users and also does not build and utilise network of metadata like the "à la" system.

The "à la" approach is consistent with the COHSE (Conceptual Open Hypermedia Service) principles (Goble *et al.* 2001; Carr *et al.* 2001b). COHSE project is one of the Semantic Web developments that introduce an ontologically-controlled hypermedia system. COHSE considers that "concepts are linked and hence their associative documents are linked". In COHSE, an ontology service supports defining the links and a metadata service semantically annotates regions of a document with a concept from the ontology, which is implemented as a thesaurus. The resource service provides list of documents related to the annotated concepts or matched language terms. "COHSE brings together both navigation and querying, directed browsing and serendipitous discovery" (Carr 2000c). "à la" also strives for serendipity, however COHSE uses a ready-made ontology, while "à la" tries to construct a network of metadata on the fly.

The Magpie project (Dzbor *et al.* 2003; Domingue *et al.* 2004a; 2004b) takes the COSHE approach further: Web documents are on-the-fly supplemented with entities definitions from an ontology service. Magpie automatically associates a semantic layer to a Web page based on a ready-made ontology-based lexicon. The Magpie project shares a similar objective with the "à la" project: supporting the understanding of documents found by search. The distinction lies in facts that "à la" does not use any ontology in advance and does not include an active component of triggering services as the Magpie does. Also, the "à la" system works by searching the documents and recommending metadata for refining a query, in addition to understanding the search result context. The latest Magpie research looks into collaborative Web browsing (Domingue *et al.* 2004c).

The Ontalk, Ontology-based personal document management system study discusses usage of a semi-automatic metadata generator and ontology-based browser (Kim *et al.* 2004). The Ontalk can either import an external ontology or extract embedded

document properties such as *Title*, *File name* or *Size*, using a similar approach as "à la" in that respect. However, Ontalk is focused on a singular user and presents the search results as a list.

Metadata Catalog (*MCAT*) provides a metadata management layer on top of distributed file collections in the Storage Resource Broker (*SRB*) project[56] (Singh *et al.* 2003). MCAT allows for definition of attribute types from the Dublin Core set or an arbitrary set. Collection items are manually assigned the *attribute-value pair* descriptions which can be later searched for.

Finally, topic maps (Pepper 2000) use similar approach as "à la". They involve text-mining techniques to automatically extract and classify information from documents, which can be later refined by knowledge managers, i.e. ontology authors.

6.6.3 Query Reformulation and Searching by Spread Activation

Search Enhancer system (Stenmark 1997; 2003) investigates how the single word search queries on the corporate intranet can be augmented. Search Enhancer uses a simple semantic net to represent knowledge about keywords and their relationships within the domain of an intranet. Only two relationships are established: generalise and specialise, and one attribute called *Synonyms*, and they are used for query refinement. Stenmark concludes that this approach allows for more precise search results and improving the users' understanding of the intranet collection of documents. The same benefit, familiarisation of the user with the application domain and with the metadata vocabulary, motivated the "à la" research.

The FACET project investigates integration of faceted thesaurus for semantic term expansion in retrieval (Binding & Tudhope 2005). Thesaurus relationships are based on a core set of standard semantic relationships, such as broader or narrower, and used together with the semantic closeness to accomplish query modifications. Querying include automatic traversal of associative relationships. FACET also uses graph/hierarchical visualisation of the thesaurus relationships in the form of

---

[56] SRB resources Available from World Wide Web: <http://www.npaci.edu/online/v6.9/srb_user_guide.html>.

browsable hyperlink structure. Like in "à la", term suggestion facility is provided.

The traditional search engine techniques are combined with the ontology based spread activation method in the works of Rocha and others (Rocha *et al.* 2004). Similarly to the "à la" concept, the system presented in this research uses terms connected to each other. Each connection has a weight attached to it, a weight that indicates the importance of a relationship.

Guha and others (2003) use keyword querying combined with the underlying ontology and navigating the ontology instances network in order to complement search results. This approach is similar to the "à la" system, however, there are the differences. The "à la" method discovers the attributes relationships by using data mining, not having a ready-made ontology instances network from the other sources. Dimensions of zzstructure are used to represent relationships and govern the spread activation, which is in case of Guha's work a simple breath-first algorithm.

6.6.4 Searching in Organisational Environments

The collaborative aspects of IR tools were discussed by (Stenmark 2000). Stenmark's study compares a user's behaviour while using a search engine and a recommender system on a corporate network. Stenmark notes that in order to be efficient in finding information via the search engine, the user must enter a very precise search string, while the true interests are often vague or difficult to express. The author notes that recommender systems are far better equipped for collaboration among system users as compared to search engines because they support awareness of the knowledge that resides within people. Our research indicates the same conclusion and we have evaluated our system in comparison to the search engine family of technologies (see Chapter 7).

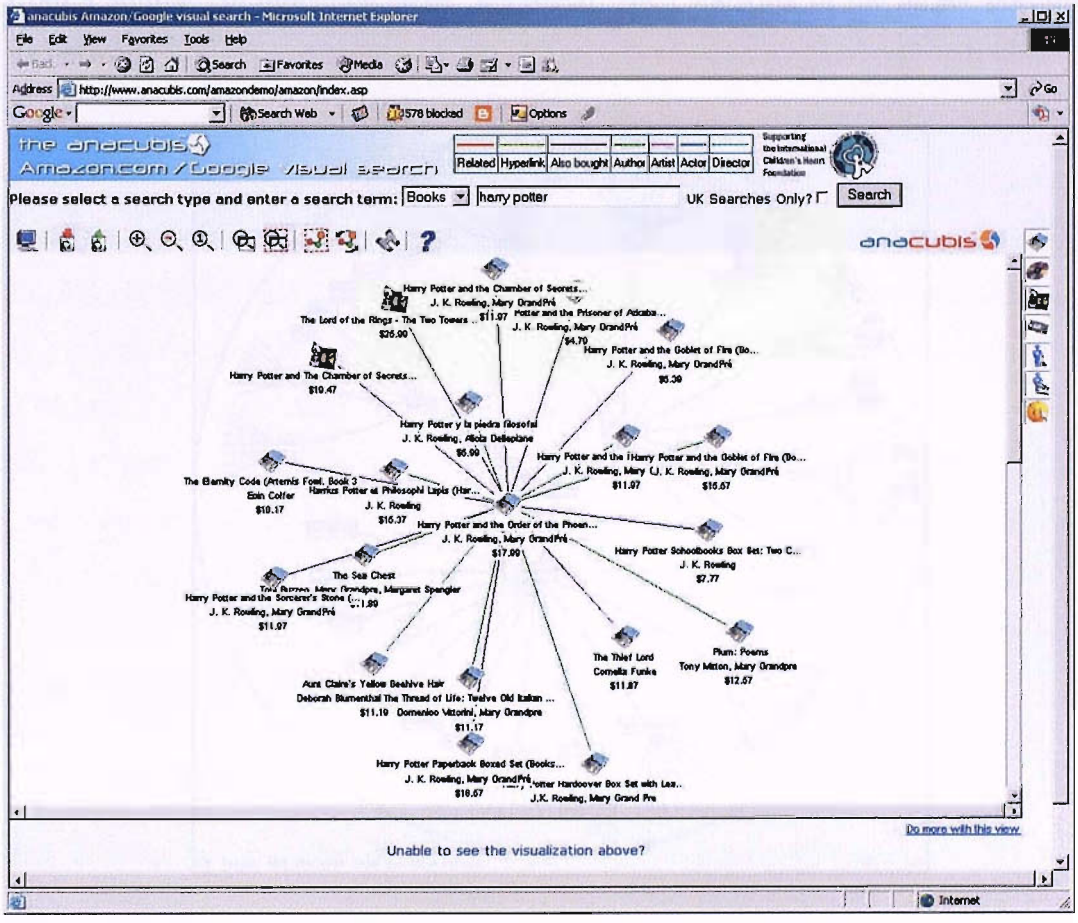
The concept of combining the metadata and the content-based retrieval methods are described in the DLESE (Digital Library) system (Deniman *et al.* 2003). The pilot study proved this approach useful in the strict librarian environment, where skilled personnel create metadata records. However in a typical corporate environment, the user-defined metadata cannot be relied on to such an extent and "à la" attempts to

combine them with the metadata mined from document content.

6.6.5 Visualisation of Complex Information Spaces

On the topic of visualising complex ontological structures, the existing research is mostly focused on representing hierarchical structures.

The Anacubis (now i2 Choice Point) demo[57] shows Amazon/Google visual search for books. It visualises a portion of the information space, centralised on the found item, such as book. The demo represents different relationships, such as *Related*, *Author* and *Also Bought*, in different colours, as shown in figure 6-10. However, the items and relationships cannot be easily navigated and only the immediate neighbourhood, i.e. items directly linked, is displayed. As a comparison, ZigZag visualisation in "à la" provides better navigation features and more flexible relationship display.



[57] Anacubis demo page, Available from World Wide Web: <http://www.anacubis.com/amazondemo/amazon>.

Figure 6-10: The Anacubis demo: user interface example, after (Choice Point 2006)

Visualisations based on hyperbolic geometry are used to display large hierarchies. The hierarchical levels are shown on the circular display surface in such a way that the current node takes the central position. The node in focus has a largest size while other level components diminish in size as they move outwards. There is also an exponential growth in the number of components, as shown on the example in figure 6-11 taken from (Lamping *et al.* 1995).

Although experimental evaluations suggest that no statistically significant performance difference was gained in comparison with a conventional hierarchical browser, it has been noticed that user preferred a hyperbolical browser. It can be argued that this kind of browser is not suitable for showing relationships other than hierarchical. On the other hand, zzstructures browser, as used in "à la", might benefit from a similar "diminishing sizes" approach when showing of the larger portion of its network.

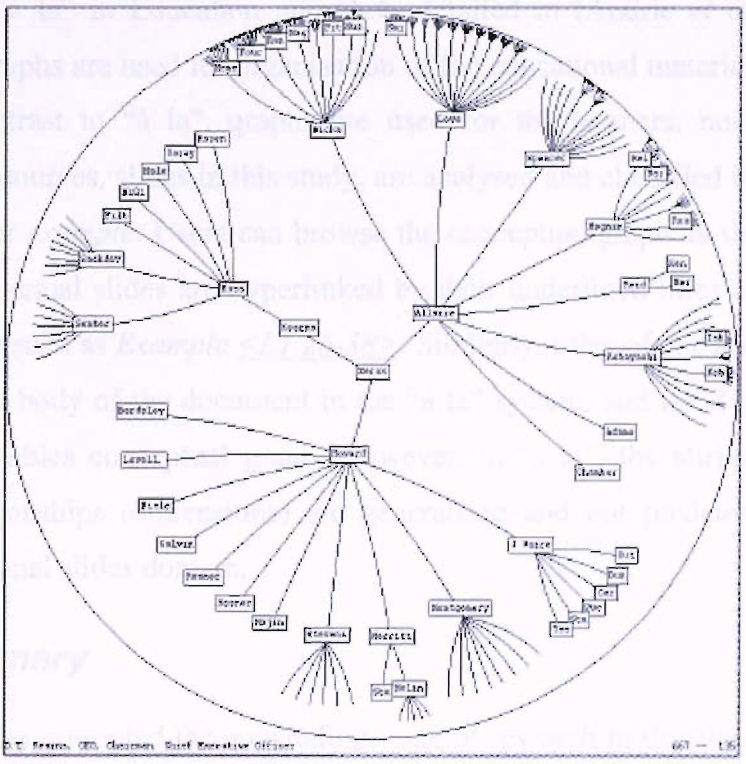


Figure 6-11: An example of the hyperbolical geometry browser for an organisational chart, after (Lamping *et al.* 1995)

Visualising complex network of relationships between metadata (and documents) is

achieved in Pathfinder (Chen 1999) using VRML (Virtual Reality Modelling Language). The visualisation presents the weights of connections in such a way that nodes with strong connections appear to be closer.

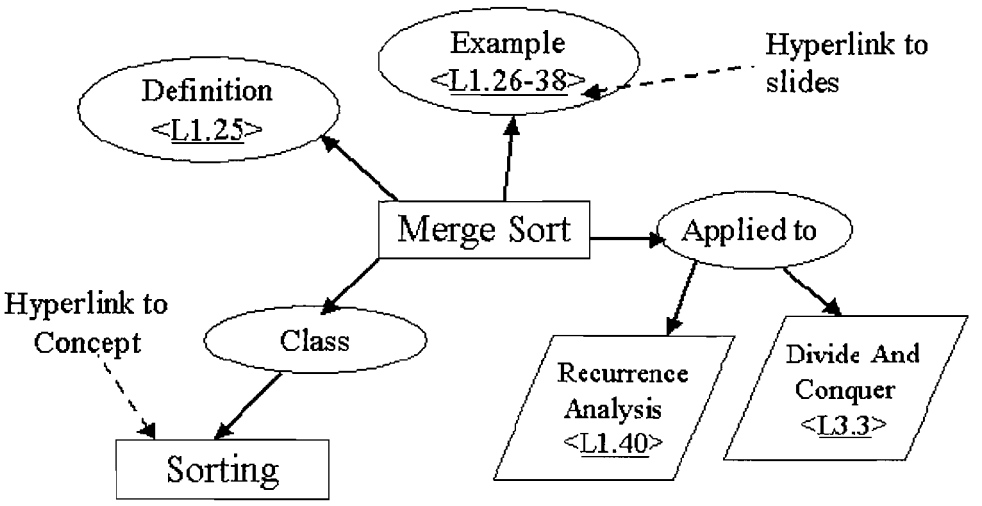


Figure 6-12: Conceptual graph example for the "Merge Sort" concept, after (Mittal *et al.* 2003)

Similarly to "à la" in Education approach, detailed in (Andric *et al.* 2005a) and section 7.2, graphs are used for organisation of the educational material (Mittal *et al.* 2003). In contrast to "à la", graphs are used for the learners, not authors. The educational resources, slides in this study, are analysed and classified into types such as *definition* or *example*. Users can browse the conceptual graph as shown in figure 6-12, and the actual slides are hyperlinked by their underlined titles attached to the attribute name such as *Example <L1.26-38>*. Similarly, titles of documents represent pointers to the body of the document in the "à la" system, and the ZigZag graph in principle resembles conceptual graph. However, in "à la", the attributes and their possible relationships (dimensions) are generalised and not predetermined to the actual educational slides domain.

6.7 Summary

This chapter has presented the concluding stage of research in this thesis. The "à la" system builds on experiences that have been collected during working on the AWOCADO project and other initial research studies. This chapter described the aims and architecture of the "à la" system. The "à la" project investigated how a

recommender system can support the task of searching when boosted by hypertextually connected metadata. The objective of the "à la" system was to supplement a typical document management system by building a rich network of semantically connected metadata on top of the document space. Three main steps, harvesting metadata, creating attribute ZigZag links and utilising the constructed structure, were elaborated. Particular attention was given to the design and population of the domain ontology. Preparing recommendations during the searching process together with the usage example scenario were described. Finally, related research, relevant to this work, was reviewed and compared with our approach.

The following chapter describes how we have approached the evaluation of the "à la" system in two selected application domains: software engineering and education.

Chapter 7

"à la" Evaluation

This chapter is focused on evaluation of the "à la" system, that makes use of HCI systems evaluation methodologies. The "à la" system is evaluated in two domains: education and software engineering. The experimental scenario is described; the results are then presented and discussed.

7.1 Introduction

In order to formally evaluate the "à la" system, a methodology using statistical data analysis in Human Computer Interaction (HCI) systems was employed (Dix *et al.* 1998)[58]. An experimental evaluation of HCI systems typically involves the following steps:

- A theory about some aspect of the system behaviour is set.
- An experiment is designed in such a way that some measurements can be taken in order to assess the system behaviour or user satisfaction.
- Metrics, i.e. numerical system behaviour indicators, are designed. They are derived from measurements.
- A small group of users (a sample) is selected so that it can well represent a wider group of users (a population).
- A series of experimental cases is conducted and the raw data (desired measurements) are collected.
- Metrics are calculated and some form of statistical analysis of the results performed.
- Based on the statistical indicators, a proposed theory is tested and either approved or disapproved.

We decided to follow the listed steps in our approach because we intend to use statistics to indicate the likelihood of the correctness of our theory. We assume that it can be inferred that a larger group of users would have similar experiences with the

[58] Book's Home page: Available from World Wide Web <http://www.hcibook.com>

system as a smaller group that participated in the experiment (Fowler *et al.* 1998). The evaluation methodology of HCI systems is described in more details in Appendix D.

The "à la" system was evaluated in two different domains, as described in the following sections. In each of the experiments and for each of the calculated metrics we obtained results as two series of data: one series for the "à la" system and another for the reference system. Two series of data are represented by their means. To test if the differences between the means of two groups of data are statistically significant, the *null hypothesis* is used. A hypothesis is a prediction of the experiment. It has to be stated in such a way that it can be tested. The null hypothesis is expressed in a negative way and the aim of the experiment is to disapprove the null hypothesis in order to prove that the prediction is correct. The null hypothesis in the case of evaluation in this thesis states that the two groups of metrics come from the same population, meaning that there is no difference between two groups of data. It actually claims that whatever the difference appears to be, it is the result of pure chance, not the statistically significant systematic difference between the observed systems. If the null hypothesis is disapproved, and the two systems proved to be different in respect of the chosen metrics, then we can look into assessing the difference.

7.2 Application of the "à la" Method to the Domain of Education

7.2.1 Experimental Setting

The first study looks into applying the "à la" principles in the area of education for supporting authors of educational material.

Web-based education has become a very important branch of educational technology (Devedzic 2004). Teachers and authors of educational material can use numerous possibilities for Web-based course offerings and teleteaching, including authoring tools for developing Web-based courseware.

In a typical scenario of creating learning material in such a context, the author would

look for resources on the Web. Then he/she would reuse and reorganise parts of the material found, creating new learning material, either using whole documents or creating specialised chunks (Brailsford *et al.* 2002) . Generally, the new material will take the form of a sequence or a network of interconnected learning objects. With current technology, the author typically uses a search engine and a keyword-based approach to locate the learning material on the Web.

The *"à la" in Education* prototype (Andric *et al.* 2005a; Andric *et al.* 2007) provides a layer between the search engines results and the user. It takes the current search result as a document collection that is analysed using the "à la" method. In this case there is no document management system, just on-the-fly document collection. Here we are focusing on evaluating how the "à la" method performs in the first cycle of searching (when the user provides keywords and receives a search result) in comparison to a baseline searching system, in this case the Internet search engine Google[59].

The aim of the evaluation was to establish how the "à la" system compares to the reference system for the selected metrics. It was accomplished in a user trial in which users performed tasks using one or other system. During the trial the subjective opinions of users were collected by the means of a questionnaire. The indicators that were of interest were: learnability, friendliness of the user interface, effectiveness and user satisfaction. The null hypothesis in each case stated that there is no difference whether those metrics were taken for the "à la" system or a reference system. The evaluation is actually aiming to show that the difference exists and to investigate which system is easier to learn to use, which has the friendlier interface, which is more effective (is better in information retrieval) and finally which system more fulfils overall user satisfaction.

The evaluation was based on a laboratory study. A set of twenty teachers was selected for the evaluation. The teachers were selected randomly among the staff of a Computer Science University. Twenty was the largest number of users that was

[59] <http://www.google.com>

practical to organise. The assumption made was that the teachers were reasonably and equally skilled in Internet search techniques and that they were using them regularly, as was a practice at the institution environment.

7.2.2 Empirical Evaluation

The experiment was conducted in the following manner:

The users were gathered together in the IT laboratory of the institution and then randomly divided into two equal groups. A brief "à la" system demonstration was given to both groups. The first group was given the task to select material for a short course in their own area, using only a search engine and bookmarking techniques. After a brief demonstration, the second group was instructed to perform the same task but using the "à la" tool. The groups were then switched. The duration of the sessions was limited to one hour. After that, they were presented with the following questionnaire to complete for the "à la" system and search engine:

> Provide a grade from 1 (the worst) to 10 (the best) for each of the following questions:
>
> - How easy was it to learn to use the system?
> - How friendly was the user interface?
> - How effective was the system in supporting your task?
> - What was the overall satisfaction with the system?

Table 7-1 was made available for users to fill in. A multi-point rating scale (Preece *et al.* 1994) was used and the meanings of its end-points were given in the questionnaire.

Table 7-1: The user questionnaire: list of measurements taken in the "à la" in education experiment

Criteria	Description	Score (range 1-10) 1 - poor 10 - excellent
Learnability	The measure to which user feels that the system is easy to become familiar with	
Friendliness of the user interface	How friendly was the system's user interface and interaction feel	
Effectiveness	The degree to which user feels that he/she can complete the task while using the system	
User satisfaction	The degree of the comfort and acceptability of the system to the user	

7.2.3 Results and Discussion

The results we obtained were in the form of 20 pairs of measurements for each of the four characteristics we measured. In this case metrics were directly equal to measurements and no further processing of the results was needed. We selected the most suitable statistical test in order to analyse them. For more details about the selection of the statistical test refer to Appendix D.

The flow chart in (Foster 2001), reproduced in figure D-1, was used as a guide in the selection process. In that diagram, a number of questions need to be answered and an answer *yes* or *no* guides through the flow chart to the other questions. When all the questions are answered a selected flow-end contains a list of applicable tests for the selected situation.

This is how the questions for the experiment in this thesis were answered:

- "data frequency counts" is not of interest (*no* branch is taken in the flow chart)
- we are looking for a difference between scores (*yes* branch taken)
- we are not comparing a mean from a single standard value (*no* branch taken because we are looking at comparing two series of measurements obtained from the "a la" and reference systems)
- sets of scores are coming from the same respondents (*yes* branch taken)

- there are 2 sets of scores ("*2*" choice selected over "*3 or more*")

Based on these choices the Wilcoxan signed rank non-parametric test (Fowler *et al.* 1998; Dix *et al.* 1998) was used to compare the obtained results, in order to show the differences between the paired observations.

Table 7-2 Evaluation results showing comparison to the classical approach using ranking 1 to 10

Metric used	Avg. rank (search engine)	Avg. rank ("à la" method)	No of <> pairs	Probability of identical distributions
Method learnability	7.70	6.75	17	<= 0.06487
Friendliness of the user interface	8.00	6.70	14	<= 0.01074
Effectiveness	7.30	8.40	15	<= 0.03534
Overall user satisfaction	7.90	8.25	13	<= 0.41430

The results shown in table 7-2, indicate that null hypothesis is not proved, i.e. "à la" and search engine methods indeed demonstrate different behaviour as input data distributions are different. The statistical test provides the degree of confidence in the obtained conclusions as well (the last column).

By looking at the averages the difference between the systems can be assessed. Results indicate that the initial learnability and the user friendliness of the interface are lower for the "à la" system compared to the reference solution. However, this observation is expected as the way of using the standard search engine solution is widely known. On the positive side, the results demonstrate better effectiveness and overall user satisfaction with the "à la" system.

We recognised the need to explore the use of other metrics, in order to confirm and expand the observations obtained in this trial. This is especially related to the effectiveness which should be objectively measured (Devedzic 2003). A more complete and formal evaluation was conducted in the domain of software engineering during the second experimentation.

7.3 Application of the "à la" Method to the Software Engineering Domain

7.3.1 Experimental Setting

The second application of the "à la" method implements our approach in the area of software engineering. It considers a document management system where software development related documents and artefacts are stored.

A decision about the hypothesis that is to be proved by the experiment was first decided on. The objective was to establish the quality of the "à la" system in comparison to the reference system. Criteria and metrics had first to be selected to measure the "quality".

We started from the assumption that the search aspect of our system is the most suitable for comparison, as it is quantifiable: search 'hits' and 'misses' can be counted. Also, our experimental preparation was guided by the assumption that we already had access to a large corpus of accumulated corporate documents in a document management system called CBP[60] (Collaborative Business Portal), a commercial product that originated from the AWOCADO research (described in Chapter 5). The user group that we planned to include in our experiment had already been using that system for months. The interviewer was observing users during this period and collecting notes on typical search interactions, the user needs and comments, which also helped in designing the experiment. It was natural then to plan a user trial in which a group of users would use the "à la" system on top of the CBP document corpus for performing search tasks. Full-text search for corporate documents could be considered as a valid reference method, having in mind that full-text search is somewhat similar to the one of Web search engines. However, as corporate documents usually lack hyperlinks, navigating from one document from the search result to another would not be taken into account.

We adopted the methodology of user experiment where users were providing explicit ratings directly into the user interface of the software evaluated. Row measurements

[60] <http://www.finsoft.com/solutions/turnkey_workflow.htm>

were then automatically collected in the database and later on processed in order to obtain the desired metrics. The ratings and metrics selection is detailed in section 7.3.2. During the sessions while users were evaluating the system, the interviewer was present. The interviewer was not interfering with the evaluation but was writing down the users' anecdotal comments.

The indicators of interest to be measured and compared with the full text search were precision (percentage of the correct documents returned) and serendipity (percentage of unexpected but valuable items found). We hoped to demonstrate that the "à la" system would prove superior in respect to those two metrics. Therefore the null hypothesis in cases of precision and serendipity claims that there is no difference whether those metrics were obtained from the "à la" system or a reference system evaluation.

The experimental setting for evaluating the "à la" prototype system was set in a real-world environment: a small team in a software vendor company developing a medium size software project. It was a live user experiment, a field study on a natural data set[61]. The CBP document management system was used as a repository of various artifacts such as release packages, test cases, design documents, specifications and so on. All the textual data was in English. The environment in which the experiment was conducted comprised:

- 700 documents, all versions contained in about 1000 files
- 50 users divided into three groups: software vendor management, analysis and development in the UK, overseas development group and a client group in Germany
- Around 5000 attributes - user-generated metadata

The population of "à la system" users in this experiment is defined as teams working on software engineering projects in different roles and based in a corporate environment. This presumes individuals of working age with a computer related

[61] These terms are used while evaluating HCI systems for describing a type of experiment, for more details see Appendix D.

background and/or experience, all speaking a chosen team communication language and working in the same organisation.

Ten users were selected as participants in the experiment. In order to avoid language problems, all participants are from the UK group. They represent a typical software team across several roles. The subjects were selected at random from within the team and type of roles. For the role we were following the general classification given in RUP (Rational Unified Process) (Kruchten 2001). While assembling the population sample, we tried to cover all roles equally as much as possible, although roles inside a software team are frequently not fixed, with people assuming different roles when needed.

A log of past queries for the selected users was inspected, and we randomly selected ten existing single-word terms from the log for each user separately. The one-word queries were selected because they are used most frequently in searches in general (Silverstein 1998; 1999; Wen *et al.* 2000) and also because indexing by single words is easier and faster than phrase-based indexing (Evans & Zhai 1996). The single-word term, i.e. 'a search expression used' was chosen as one independent variable[62] and based on that, several dependent variables were measured, as detailed in the next section.

7.3.2 Formal Evaluation

Prior to setting up the evaluation experiment we analysed the users' usual tasks that the system would have to serve. We noticed by observation that users were mostly interested in finding a small set of documents (items) related to their current interest. They were expressing their query, usually using only a single-word term, and submitting it to a full text search engine (in this case Microsoft Indexing Service) which was integrated into the document management system in order to automatically find and show document titles for each file found.

Then, the users were browsing through the ranked list and opening the potentially

---

[62] See definitions in Appendix D.

interesting documents. If they were not satisfied with the result, the users were modifying their query and repeating the whole process. We identified that in a software project, typical user' tasks that our recommender service should cover fall into two categories, as identified in (Herlocker 2004): *Find Good Items*, where a list of ranked recommendations is returned, and *Help Others*, where users are contributing their opinion to the community.

In contrast to a typical recommender system, our users wanted to see again some items they already knew about. Users were looking for a 'golden copy' from a central location which could have been changed since their last access, or they simply needed the information again. The importance of "refind" tasks was highlighted in the study by Capra and Perez-Quinones (2005).

We also noted that users are not interested in the degree of the relevance. Therefore, it was concluded that only an explicit binary rating for relevance is needed: "Relevant/Irrelevant". One more type of rating was introduced: the "Bonus/Not Bonus" rating. This rating serves to identify Serendipity or Novelty items, items which might be both attractive and somewhat surprising to the user. Users were required to mark "Bonus" items in our experiment. All other items would automatically assume "Not Bonus" ratings.

As shown in figure 6-9, the "à la" user interface contains a series of check boxes placed on the left hand side of each of recommended items. The check boxes are present for both metadata and documents. The boxes are initially un-ticked with the implicit meaning of item not being serendipity, i.e. bonus item. The users would need to tick the Bonus box in order to mark an item as a bonus item. For the items in the list which are documents (not metadata), there exist also a radio button on the GUI interface, with two settings: ✓and ✗. Selecting the appropriate radio button enables users to mark the document as relevant or irrelevant to the query. A document can be irrelevant for the current query, but present an unexpected and valuable finding and thus marked with the bonus rating. The "à la" user interface was slightly extended for the purpose of this experiment. The list of found documents was repeated twice, only the second list was created by consulting the full

text search method. This extension enabled users to compare the search results and to rate both methods at once.

The main goal of the experiment was to measure the quality of the recommendations compared to the standard way of searching (full text search paradigm), in other words to collect the desired metrics and test the null hypothesis. At the same time, the evaluation attempted to collect user impressions about the ZigZag visualisation.

Each user was given a list of the queries, some brief training and the same task:

> Submit a given query to the system. Rate returned items as either relevant or irrelevant for both the "à la" system and the full text search. Also, please mark *Bonus items*, those that are interesting and surprising to you, regardless of their relevancy to a current query. Try out navigating metadata in the ZigZag browser. Feel free to express your observations and opinions during the session.

The interviewer sat with each of the users separately and wrote down their comments during the session.

Users rated the documents/artifacts using the Relevant/Irrelevant choice and also set the Bonus rating, both for the "à la" and the full text search (FTS). For the metadata recommended, only the Bonus rating was applicable. The aim was to measure how users found suggested attributes as candidates for query modification.

Table 7-3 shows the measurements that were taken for each of the 100 cases (10 users x 10 queries).

Table 7-3: List of measurements taken in the "à la" software engineering experiment

D_{ala}, D_{fts}	The total number of returned documents using "à la" and the FTS respectively
R_{ala}, R_{fts}	The number of documents rated relevant using "à la" and the FTS respectively
B_{ala}, B_{fts}	The number of bonus documents using "à la" and the FTS respectively
M_{md}	The number of bonus returned metadata
A_{md}	The total number of returned metadata items

We only showed the top 30 returned documents although in many cases the total number of documents was higher. The choice of 30 was selected based on observation that, for example, Google users most of the time look for results in only the first 2-3 pages, each page containing 10 results. Initially, in the pre-experiment trial, we showed all results to two users, and they expressed a desire to see less than 'a few dozens' items. Therefore both D_{ala} and D_{fts} are less than or equal to 30. One possible future work task would be to change this number during future evaluations and monitor how this influences results.

Table 7-3 shows the metrics that were formed based on the defined measurements.

Table 7-4: List of metrics used in the "à la" experiment

$P_{ala} = \frac{R_{ala}}{D_{ala}}$	Precision for the "à la" system
$P_{fts} = \frac{R_{fts}}{D_{fts}}$	Precision for the FTS system
$S_{ala} = \frac{B_{ala}}{D_{ala}}$	Serendipity indicator for documents for the "à la" system
$S_{fts} = \frac{B_{fts}}{D_{fts}}$	Serendipity indicator for documents for the FTS system
$S_{md} = \frac{M_{md}}{A_{md}}$	Serendipity indicator for metadata (in the "à la" system only)

The first quality indicator used, in accordance with the typical information retrieval metric, is *precision*. Precision is here defined as a ratio of relevant items and the full number of items assumed to be relevant, i.e. measures the correct classification. It is usual that IR systems are assessed by measuring *recall* as well, however we considered that impractical for a large number of documents. It would be possible to

estimate recall using an expert's knowledge of which documents should have been returned but were not, in conjunction with the sampling techniques (Salton 1968). Estimating recall will be attempted as part of a future evaluation.

7.3.3 Results and Discussion

When the row measurements were obtained via the user interface for all users, metrics were calculated. In the end there were 5 series of 100 numbers: a pair of series for precision, a pair for document serendipity and only one series for metadata serendipity. The statistical tests were then selected in order to process data. The flow chart in (Foster 2001), reproduced in figure D-1, was used as a guide in the selection process. The diagram was used in a similar manner as explained in section 7.2.2 and the answering process guided the selection of two sample t-test for the paired data and one sample t-test for the remaining metadata serendipity series.

Table 7-5: Experimental results: calculated metrics in the "à la" experiment

User	Example query	Avg no of returned docs		Avg no of relevant docs		Precision (documents)		Serendipity (documents)		Serendipity (metadata)
		D_{ala}	D_{fts}	R_{ala}	R_{fts}	P_{ala} (%)	P_{fts} (%)	S_{ala} (%)	S_{fts} (%)	S_{md} (%)
Project Manager	Scope	20	30	5	5	29.76	17.33	5.00	7.33	16.90
Infrastructure Team Leader	Release	23	27	8	7	37.48	28.67	15.01	6.00	18.08
Business Analyst	RUP	23	30	11	5	50.15	17.67	10.05	7.00	17.90
Support	Startup	22	28	8	7	35.05	27.16	3.73	3.5	16.48
Research Analyst	Rules	16	26	3	3	25.77	11.47	11.33	5.53	13.61
ExtractTransformLoad Specialist	Interfaces	24	30	5	6	22.12	21.67	7.18	7.33	10.18
DataWarehouse Specialist	Dimension	20	30	4	5	19.24	16.66	8.32	2.33	12.06
Program Manager	Iteration	23	28	10	7	45.61	26.10	10.11	4.33	27.35
Account Manager	Issues	21	30	8	7	36.47	25.67	2.13	3.33	12.63
Testing Specialist	Portal	27	30	12	9	48.83	30.33	7.58	3.66	13.19
OVERALL						**35.05**	**22.27**	**8.04**	**5.03**	**15.83**

Table 7-6: T-test results for comparing series of metrics

	Precision (documents)	Serendipity (documents)	Serendipity (metadata)
T value	4.44	3.10	
Probability (x>t)	< 0.0001	<0.00102	
Standard deviation	for P_{ala}: 23.87, for P_{fts}: 16.17	for S_{ala}: 8.53, for S_{fts}: 4.61	
Mean Absolute Error	for P_{ala}: 0.65, for P_{fts}: 0.78	for S_{ala}: 0.92, for S_{fts}: 0.95	for S_{md}: 0.84

The metrics defined in table 7-4 were gathered during the experimental run. The averaged experimental results, represented by the mean values of the established metrics of the top 30 items returned, are given in table 7-5. The results are presented as percentages for each user separately and overall.

We compared precisions for both systems by applying a two sample t-test in order to compare the two paired samples representing raw precisions P_{ala} and P_{fts} for each experimental case. Then we again applied a two sample t-test, this time for the raw serendipities. The results of both tests (table 7-6) show that there is a significant difference between the two observed systems, therefore our null hypothesis that precision and serendipity do not differ in the observed systems, is disapproved, with a relatively high probability. Metrics which were compared are represented by their mean values and the relationship between means (averages) indicates the size of that difference.

The results presented in table 7-5 indicate better precision and serendipity levels for the "à la" system, precision at 35.05% compared to 22.27%, and serendipity indicators for documents at 8.04% vs. 5.03%. "à la" was consistently performing better than full text search method (FTS). The serendipity metrics for the metadata (15.83%) could not be compared to anything, as there is no equivalent in the full text search paradigm, nor does there exist a hypothetical mean value needed for the one sample t-test.

If we look at the precision, which is about 35% on average for our method, compared with around 22% for the reference system, we notice that both averages

are below the usual rates of 50%-70% reported for pure IR systems (Lancaster 1968) and Web searches (Cassola 1998; Sherman 2002). This result is however not unusual for the organisation-based highly focused document collection where a large percentage of queries tend to have a small set of correct search results (Fagin *et al.* 2003).

As precision and recall are considered to be inversely related (Herlocker 2004), it is to be expected that the FTS would have better recall. Indeed, as noticed in the experiment, in cases where queries were of a specific nature for the domain, the FTS was superior. The reason is that specific infrequent terms would never occur in the metadata, thus influencing the "à la" coverage. However, the "à la" system provided much better results in the case of the more general domain terms, i.e. terms which are quite general but fall inside a domain. An example in a software engineering domain would be the term "Configuration" and such terms would hold more ZigZag links towards the other metadata. We concluded that rich ontological interconnections enable the discovery of such results which could not be found otherwise by using traditional text-based techniques. During the session interviews users indicated that a combination of the two systems would yield the best results.

Another matter that influenced the lower precision for the FTS was that this method reported various versions of the same document. Users mostly did not consider them relevant and were marking only the latest versions with a positive rating. Users also commented that the system helped them to identify the duplicate or almost duplicate documents.

There is not much research on serendipity levels (Herlocker 2004), but intuitively we can say that the serendipity levels, reported as about 8% for documents and approximately 16% for metadata, were high. We could expect this to become lower with further usage of the system, as the novelty declines.

The time taken to perform the tasks was not measured in this experiment. This was considered not to be a relevant indicator, because users were fairly familiar with most of the items and were running known queries. Therefore, it was quite easy and fast for them to determine the relevancy of each item.

The typical error rate (mean absolute error) of 0.73 is reported for collaborative recommender systems evaluations while rating movie datasets with the ranking of 5 (Herlocker 2004). Compared to that, the error rate in "à la" for binary ranking (i.e. good/bad), was lower, calculated to be 0.65. The reference system was found to have an error rate of 0.78. A comparison with the collaborative recommender systems cannot be directly drawn, because of the different data set and the usage of content analysis in "à la", but the lower error indicates a somewhat better classification accuracy. We conclude that this is again due to the fact that the data set was not very large and it contained domain-focused items. However we believe that such a situation is quite usually the case in software development projects, especially when only one project is considered.

The secondary objective of this experiment was to gather the user comments on the ZigZag visualisation. No formal evaluation was performed though and only anecdotal evidence is presented here.

User experiences with the ZigZag visualisation were generally positive; although they expressed a preference for looking at the list of recommendations, as shown on the left half of the screen (figure 6-9 in Chapter 6). They also noted that if the metadata recommendations were not shown in the list, they would browse the ZigZag ontology network to see more connected items along a selected dimension. However, if the data users were browsing in ZigZag was not familiar to them, for example not a list of teams or users but a long list of unknown stemmed keywords, they experienced a sense of being lost.

Also, if their navigation of the metadata was interrupted, with a telephone call for example, users reported that after returning to the ZigZag browser they had difficulty remembering how to back track to the previously selected cell in order to follow some other dimension from there. In that case users would frequently use the hyperlink in the left list of metadata as a shortcut to 'jump' to a familiar place in ZigZag from where they would restart their browsing. Those observations were consistent with reported observations users had with the ZZDirectory system, previously described in Chapter 5 and in (Andric *et al.* 2004a).

7.4 Summary

Experimentation and related evaluation issues were presented in this chapter. The main objective of this evaluation was to test the hypothesis that the "à la" approach provides better support to users for the tasks of information retrieval than a reference solution.

The first study measured users' subjective opinions about the learnability, friendliness, effectiveness and the overall satisfaction of the "à la" system compared to a search engine. The results indicate that the "à la" system performed better than the reference system in respect to the last two metrics. However, we attribute the lower perceived learnability and friendliness to the fact the users did not have daily experiences with the system and that those metrics would improve with time.

The second study adopted a more formal approach. The experimentation setting comprised a real life collection of documents and artifacts created in a software engineering project. The study involved a selection of the user population engaged in a task of searching using our approach and in parallel, using a reference, full text search (FTS) system. Precision (the fraction of relevant search results) and serendipity (the fraction of novelty or positively surprising items in a search result) were the metrics chosen. The t-test statistical method was selected for results analysis. We concluded that our system shows better performances than the FTS for the chosen indicators. For example, our overall precision was 35.05% compared to 22.27% for the FTS. The calculated mean absolute error rate was 0.65 which compares well with some evaluations the precision aspect of recommender systems described in the literature.

This chapter concludes the description of the work undertaken in this research. The final chapter of this thesis presents some directions for future research and concluding remarks.

Chapter 8

Future Work and Conclusion

The objective of this chapter is to conclude this thesis and to outline the potential areas of future work.

An overall summary of this dissertation is presented and its main achievements identified. The three key contributions of "a la" research are revisited: a novel approach to metadata ZigZag linking; using a metadata network to support search; and the prototype system evaluation. The initial thesis is then reiterated and discussed has it been proven.

In the future work sections, a method of ontological network analysis is proposed, aiming to make the metadata network denser. Possibilities of system learning by adjusting weights are discussed. Finally, personalisation issues are examined and ways to add multi-profile and multi-ontology network, are also discussed.

8.1 Summary and Hypothesis Revisited

As pointed out by a search engine pioneer Tim Bray[63], searching for words is not really what people want to do; they want to search for ideas, for concepts, for solutions, for answers (Bray 2003). Also, users frequently need to refind the same information. The future search tools must go further than the habitual keyword searching and make use of "user's recognition and recall abilities" (Capra & Perez-Quinones 2005).

As there are still a lot of issues that need to be resolved before the true potential of the Semantic Web can be fully exploited (Ossenbruggen *et al.* 2002), this thesis provides an initial step forward into that direction. It attempts to show how a typical document management system search can be enriched to support more intelligent searching for items, even at this point of time[64] when the Semantic Web does not yet contain much information. Also, work in this research shows possibilities of providing bases for the future Semantic Web metadata vocabulary building by mining terms and their associations from text.

---

[63] Tim Bray is also best known as a co-author of the XML specification.

[64] In year 2006

The reviews presented in Chapters 2, 3 and 4 have set a scene for the areas covered in this thesis. Document management including information retrieval, collaborative information management in recommender systems and the knowledge management field have been surveyed. Consequently, our initial research in the mentioned areas was described in Chapter 5, through the descriptions of systems AWOCADO, MAGENTA and ZZDirectory.

The idea of using associative metadata ZigZag links for document recommending through a concept of "Query by Association", have been introduced in this thesis in the core Chapters 6 and 7. The research was conducted, and the prototype system "à la" was developed and evaluated in two different areas: the area of finding educational material for course authors and in the area of assisting software engineering team members to find and refind related documents. The evaluation results have demonstrated that utilising metadata ZigZag linked into an ontological structure for supporting information retrieval offers an advantage over traditional searching techniques. These findings support the claim of the initial hypothesis of the research.

The claim of this work, as stated in the introductory chapter, was:

The hypothesis of this dissertation is that utilising metadata linked in a ZigZag fashion into an ontological structure for supporting information retrieval, offers an advantage over the traditional searching techniques.

8.2 Thesis Main Achievements

The section will summarise the main achievements of the thesis.

8.2.1 Novel Approach to Metadata ZigZag Linking

The novel approach to metadata ZigZag linking proposed in this thesis comprises:

- *Extracting metadata from textual resources, embedded attributes in documents and structured data.*

  The keywords were extracted from texts using standard text analysis and text mining techniques. Embedded attributes from MS Office type of documents were

collected. Structured data, in the form of the relevant database fields (attributes) from the document management database, were gathered as well. All three types of attributes were merged together in a document header for each document and their importance for a document established by setting the appropriate weights. The importance was determined using a combination of statistical and heuristic rules.

- *Discovering metadata associations and weaving their* ZigZag *links in an ontological hypertext network.*

  Metadata ZigZag links were discovered by using guided (semi-automatic) collocation analysis between values of selected attribute types. Two types of attributes were selected for analysis at a time and ZigZag links between the pairs of their instances established. Weights of the ZigZag links between values were calculated separately. Metadata and their ZigZag links were appropriately converted and stored in a multidimensional ZigZag information space, in the form of zzstructures implemented in a relational database.

- *Browsing metadata and their* ZigZag *links.*

  A novel way for browsing the metadata ontology network was also presented. The ZigZag browser prototype, built on top of the existing ZigZag research prototype, shows how to browse a very complex ontological network of metadata whilst at the same time keeping users oriented. The *"lost in hyperspace"* effect was addressed by progressively revealing new items and new ZigZag links as users navigate the high dimensional information space, using only two dimensions at a time.

8.2.2 Novel Approach to Using a Metadata Network for Retrieving Documents

A network of metadata and ZigZag links was used in a novel way not only to enable suggestions for query expansion, but also to find documents and metadata by traversing the network. A document identifier is considered to be a representation of a document and it is a piece of metadata itself. Therefore documents themselves are participants in the metadata network. When the system is used for term searches a user's query is analysed and terms are located in the ontological network. The ZigZag links are then traversed from a starting term and the query is effectively

expanded with the neighbouring and headcell terms. The network of metadata is subsequently traversed further and certain visited nodes collected in the recommendation list. Finally, a recommendation list is divided into two result sets: the recommended metadata and the recommended documents.

8.2.3 System Evaluation

An initial system evaluation was conducted in order to determine how the searching aspect of the system behaves compared with a traditional method of information retrieval. Firstly, the system was evaluated in the domain of education, where it was used to assist resource location for authoring the Web courseware. Secondly, numerical evaluation was conducted in the domain of a software engineering. The software development team was using a system to deposit and consequently search software documentation and artifacts. A full text search, used as a reference approach, was compared to the "à la" approach by measuring precision and serendipity of the obtained search results. In the first experiment the "à la" system's provided average mark of 8.25 (out of 10) for the overall user satisfaction, compared to 7.90 for the reference system. In the second experimentation the overall precision and serendipity were at 35.05% and 8.04% respectively, compared to 22.27% and 5.03% for the reference system. The mean absolute error rate was 0.65 which compares well with some evaluations of recommender systems precision aspect described in literature (error rate of 0.73).

The key evaluation observation was that the "à la" system behaves significantly better than the classical search methods, especially in cases where general terms within the domain are looked up. The "à la" method discovers more associated metadata and documents and is capable of serendipitous discoveries as well. The evaluation confirmed the expectations that in a numerous number of cases the presented approach proved to be superior.

8.3 Future Work: System Enhancements

The potential future work for the "a la" research can be divided into two groups, one which contains smaller improvements (system enhancements) and the other for larger issues (major new directions).

8.3.1 Keyword Extraction Improvement

Currently the "à la" system uses TF-IDF techniques (Salton 1989) for the text analysis and keyword extraction. The usage of LSA would give a better start to the zzstructures building algorithm, as some keyword relationships, such as synonymy, will be easier to discover.

8.3.2 Multi-word Keywords and Queries

The "à la" system could be extended by introducing multi-word index terms. The text analysis would start with finding n-gram keywords, as is normally utilised in LSA. Therefore, attributes originated from keywords would be in a form of multi-word expressions.

Also, future evaluations should include multi-word queries (for which support already exists in the "à la" system). This would then involve comparing the results obtained by using single-term or multi-term user queries with the aim of establishing whether the usage of multi-term queries influences the results.

8.3.3 Evaluation Enhancements

It is difficult to exactly determine IR measure recall in a large collection of documents. Future evaluations could utilise estimated recall for returned documents. An experienced evaluator would need to inspect all the documents for each query in order to precisely determine how many items were supposed to be returned as part of the search result. This is not practical and recent approach to this evaluation issue is to use experts, people who are quite familiar with the full contents of a document collection (Herlocker 2004). Such experts are expected to know (as precisely as possible) which documents should be returned as matching for a particular query. They would then estimate what proportion of the items was not returned in the search result, compared with what should have been returned. Conducting the experiment in this way would enable a recall measure to be gathered and compared to the other measures.

Future work could also include experimenting with a different number of top items returned by a query and observing how this influences precision and recall. The goal

would be to find the right balance between presenting a manageable list of recommendations (search results) to the user, without paying the price of missing the relevant items.

8.3.4 Improving Portability and Scalability

The portability of the "à la" system could be improved, to use the standard API for interfacing with document management systems, called ODMA[65]. Also, scalability issues need to be addressed.

Firstly, the keyword extraction implementation could be improved by porting it from Java to a faster C language.

Secondly, we could investigate what is the impact on scalability and performances if zzstructures are built and stored in the RDF format. The obvious benefit lies in the usage of existing RDF manipulation tools and consequent portability of data itself.

Thirdly, better visualisation of a large zzstructure should be addressed by implementing more advanced ZigZag concepts, such as rasters, a pre-selected set of cells singled out from the zzstructure according to some rule (Lukka 2002), or by introducing bookmarking of the interesting cells and dimensions.

8.4 Future Work: Major New Directions

8.4.1 Using Ontology Network Analysis

The major direction for further work lies in better preparation of the ontology network for the task of searching. The focus would be in discovering new ZigZag links or associations between the existing terms, in order to make a network denser. This could be obtained by using the carefully selected transitive rules, i.e. if node A is connected to a node B and the node B is connected to a node C, in some cases, (for certain relationships) it makes sense to deduce that nodes A and C can be connected.

The "à la" method for population of metadata could be further advanced by utilizing

---

[65] Open Document Management, collection of resources, Available from World Wide Web: <http://www.infonuovo.com/odma/>.

Ontology Network Analysis (*ONA*), as in ONTOCOPI (Alani *et al.* 2002; Middleton *et al.* 2002), with the aim of inferring further attribute ZigZag links. The obtained network can be further analysed for occurrences of authorities (nodes on the receiving end of many connections) and hubs (nodes pointing to many other nodes, especially authorities). It would be interesting to observe which ontological terms represent hubs or authorities and what their meaning might be in the context of the specified domain.

Furthermore, "à la" could benefit from a synergy with an external ontology. The "à la" metadata could be enriched by mapping them onto the external ontology's categories in a similar way as proposed in (Kamolvilassatian *et al.* 2001). Thus, more semantic links can be generated.

8.4.2. Learning and Personalisation Issues

Future work should investigate issues of making the system learn. This could be accomplished by feeding the user ratings into the ZigZag ontology network and adjusting ZigZag link weights accordingly. If a certain document was rated positively by a user, then it could influence all the ZigZag links on the path between that document and search term(s) used to find it. The weights on all the paths which connect search terms to a document and are traversed via a search algorithm, would be adjusted positively by a certain factor. This would enable users to implicitly blaze their own associative metadata trails. These trails could then be recommended to the users' collaborative group - an idea inspired by Bush's vision of trail usage (Bush 1945). Repeated experiments should be conducted in order to determine how learning of user preferences would improve the recommendations.

The issue of a dynamic and evolving ontology network is closely connected with the issue of personalisation. In one approach, all the ratings from different users could be fed into the same ontology network. In this way the changing weights would reflect an averaged community opinion about which terms are relevant to which documents, and also which terms are associated strongly with each other.

However, not all people have the same interests and furthermore they might be

looking for information in different contexts. Moreover, users can act in different roles and need different profiles, such as a work or a hobby profile. Therefore the challenges of adaptive ontology networks need to be addressed. In this case, a ZigZag's generality can help in two ways:

Firstly, multi-profile support can be easily added. A new user's profile can be included as just an additional dimension or set of dimensions. At the moment a user's profile in "à la" is kept as a weighted vector of terms which can be considered to be the user's profile dimension. Other user's profiles can be kept in a similar way, only belonging to their own dimensions. Users could select their profiles and thus implicitly set an active profile dimension. In this case a multi-profile environment is seamlessly achieved.

Secondly, a *multi-ontology network*, in other words an ontology network that is adapted to each user, can also be achieved. If a result of the user's actions causes some ZigZag link weights to be adjusted, there exists a way to store this change. At the same time, the existing set of weights would be kept, which is possible by adding a new dimension. A newly created dimension is meant to enable saving new sets of weights. Let's say for example that the search term was "LDAP" and that the search has found a document "LDAP roles management", via a rank belonging to a dimension *Document using Keywords*. It is easy to add a new dimension *Document using Keywords [UserA]* that would connect the same nodes but with the adjusted weights for *UserA*. In this way dimensions in "à la" could be either the default ones (i.e. system built) or there would be a number of user personalised dimensions per user. A search algorithm would always preferentially use a current user's dimensions over those of the system. In this case a personalised, *multi-ontology network* is seamlessly achieved with ZigZag.

8.5 Conclusion

The aim of the final chapter was to revisit the work presented in this thesis and to confirm that the initial objectives are met and that contributions are achieved.

The way in which the research presented in this thesis could be continued was

described in this chapter. Certain smaller scale changes have been discussed first: e.g. improving the keyword extraction by using the LSA method and improving system portability. Issues for a new experimental design have been suggested: introducing multi-word queries, using estimated recall measure and varying a number of top ranked documents. A way forward for increasing a number of ZigZag links between network nodes using ontological network analysis has been pointed out. Ways to deal with personalisation and learning issues of the "à la" ontology network have been described as well.

In the course of this work a software prototype was designed to prototype the research ideas and fully realise their benefits. The "à la" system was designed with the aim to investigate possibilities of using associative ZigZag linking of attributes for the search support in a document management system. In the course of research we have concluded that "à la" allows also for the organisational knowledge to be identified (by content analysis), extended (by discovering associations) and provided to users (in a ZigZag browser). The advantages of the presented "à la" system are believed to be twofold.

- Firstly, with a reasonably little effort, using knowledge-based recommender system techniques on metadata ontology instances built from both structured and unstructured data, can yield immediate effect in augmenting the quality of search. The information retrieval improvement claim has been supported by the evaluation results, summarised in section 8.2.3. Moreover, the "à la" approach supports guiding the information delivery process by making use of switching between searching and browsing in ZigZag managed information space.

- Secondly, "à la" can potentially help understanding the existing domain concepts and their relationships, as a first step towards building a more formal organisational ontology in the bottom-up manner. By navigating the visualised metadata network and following up suggestions for recommended related metadata and dimensions, users can gradually become more aware of the organisational vocabulary: frequent attributes and their associations.

Mining metadata in the "à la" system as a preparatory step for building ontologies, represents the observation about how the "à la" approach could be utilised in building of the future Semantic Web applications and promoting its principles in the knowledge-intensive corporate environment.

The overall "à la" research conclusions support the claims of the father of the Semantic Web itself, Sir Tim Berners-Lee, that merging the web of human-readable documents with a network of machine-understandable metadata promises immense potential (Berners-Lee 1998b).

This research was a journey along which the research landscape was constantly changing: the last five year period have seen a very fast development of the knowledge management area in general. This research takes us one step closer to the final destination of intelligent information handling by computers. However, there is still a lot to be done in the vibrant field of the Semantic Web. While languages for exchanging the metadata are already available, it will take more time for the industry to agree on the shared vocabulary for the metadata elements and values, as argued in (Steinacker *et al.* 2001). Metadata management relevant to this field, and especially intelligent assistance for metadata navigation, will present a research challenge in years to come.

References

Agrawal, R., Imielinski, T., and Swami, A. (1993) Mining Association Rules between Sets of Items in Large Databases. *In Proceedings of the ACM SIGMOD International Conference on Management of Data.* Washington, DC, United States

Aho, V., Hopcroft, E., and Ullman, E. (1974) *The Design and Analysis of Computer Algorithms.* Reading: Addison-Wesley

Aitchison, J., and Gilchrist, A. (1987) *Thesaurus construction: a practical manual.* London: ASLIB

Alani, H., Jones, C., and Tudhope, D. (2000) Associative and Spatial Relationships in Thesaurus-Based Retrieval. In *European Conference on Digital* Libraries. Lisbon, Portugal. Available from World Wide Web: <http://citeseer.ist.psu.edu/alani00associative.html>. [Accessed Mar 2006]

Alani, H., O'Hara, K., and Shadbolt, N. (2002) ONTOCOPI: Methods and Tools for Identifying Communities of Practice. In *Proceedings of Intelligent Information Processing Conference*. Montreal, Canada

Alaniz, A., Pimentel, M., and Camacho-Guerrero, J. (2002) An infrastructure for open latent semantic linking. In *Proceedings of the ACM Hypertext 2002 Conference*. Maryland, United States. Available from World Wide Web: <http://portal.acm.org/citation.cfm?id=513338.513369>. [Accessed Mar 2006]

Andric, M., Griffiths, J., Reich, S., Davis, H., and Hall, W. (1998) Web Assistant Navigator: Dynamically Generating Branching Guided Tours using Agents and Trails, In *Proceedings of ETRAN '98 - XLII Conference for Electronics, Telecommunications, Computers, Automation, and Nuclear Engineering*, Vrnjacka Banja, Yugoslavia

Andric, M., Hall, W., and Carr, L. (2003) Implicit Document Recommending Using Associative Linking of Attributes. Poster and demonstration. In *Poster Proceedings of the ACM Hypertext 2003 Conference*. Nottingham, UK

Andric, M., Hall, W., and Carr, L. (2004a) ZZDirectory: Web Directory Browsing using ZigZag Paradigm. In *Proceedings of the PREP 2004 Conference*. Hatfield, UK

Andric, M., Hall, W., and Carr, L. (2004b) Assisting Artifact Retrieval in Software Engineering Projects. In *Proceedings of the ACM Symposium on Document Engineering (DocEng)*. Milwaukee, Wisconsin, United States

Andric, M., Devedzic, V., Hall, W., and Carr, L. (2005a) "à la" in Education: Keywords Linking Method for Selecting Web Resources. In *Proceedings of Intl. Conference on Artificial Intelligence in Education (AIED'05)*. Amsterdam, Netherlands

Andric, M., and Hall, W. (2005b) *AWOCADO: Using Metadata for Information Retrieval in Intranet-based Document Management Systems*. Technical report ECSTR-IAM05-010. Southampton: University of Southampton

Andric, M., and Hall, W. (2005c) Using Metadata for Information Retrieval in Document Management Systems. In *Proceedings of IEEE EUROCON 2005*. Belgrade, Serbia and Montenegro

Andric, M., Devedzic, V., Hall, W., and Carr, L. (2007) Keywords Linking Method for Selecting Educational Web Resources à la ZigZag. To be published in *International Journal of* Knowledge *and Learning*

Anick, P. (1994) Adapting a Full-text Information Retrieval System to Computer the Troubleshooting Domain. In *Proceedings of the ACM SIGIR 94*. Dublin, Ireland

Arampatzis, A., Tsoris, T., Koster, C., and van der Weide, T. (1998) Phrase-based Information Retrieval. *Information Processing & Management Journal.* Vol. 3, No. 6

Ashman, H. (2000) Relations Modelling Sets of Hypermedia Links and Navigation. *The Computer Journal.* Vol. 43. No. 5

Baeza-Yates, R., and Ribeiro-Neto, B. (1999) *Modern Information Retrieval.* New York: Addison-Wesley

Balabanovic, M. (1997) An Adaptive Web Page Recommendation Service. In *Proceedings of the First International Conference on Autonomous Agents (Agents '97).* Marina del Rey, California, United States

Balabanovic, M., and Shoham,Y. (1997) Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM.* Vol. 40. No. 3

Barret, R., Maglio, P., and Kellem, D. (1997) How to Personalize the Web. In *Proceedings of CHI '97.* Atlanta, Georgia, United States. Available from World Wide Web: <http://citeseer.ist.psu.edu/19781.html>. [Accessed Mar 2006]

Baudisch, P. (1999) Joining collaborative and content-based filtering, In *Online Proceedings of the CHI '99 Workshop Interacting with Recommender Systems.* Pittsburgh, Pennsylvania, United States. Available from World Wide Web: <http://citeseer.ist.psu.edu/baudisch99joining.html>. [Accessed Mar 2006]

Bauer, T., and Leake, D. (2001) WordSieve: A Method for Real-Time Context Extraction. *Lecture Notes in Computer Science.* Vol. 2116

Bechhofer, S., Goble, C., Carr, L., Kampa, S. Hall, W., and De Roure, D. (2003) COHSE: Conceptual Open Hypermedia Service. Annotation for the Semantic Web. In: S. Handschuh and S. Staab. (eds). *Frontiers in Artifical Intelligence and Applications.* Vol 96.

Belkin, N. (2000) Helping People Find What They Don't Know. *Communications of the ACM.* Vol. 43, No. 8

Berendt, B., Hotho, A., and Stumme, G. (2002) Towards Semantic Web Mining. *Lecture Notes in Computer Science.* Vol. 2342

Berners-Lee, T., Fischetti, M., and Dertouzos, M. (1999) *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor.* San Francisco: Harper

Berners-Lee, T., Hendler, J., and Lassila, O. (2001) The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American.* Vol 284. No. 5. Available from World Wide Web: <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>. [Accessed Mar 2006]

Berners-Lee, T., and Miller, E. (2002) The Semantic Web lifts off. *ERCIM News: Special theme: Semantic Web.* No. 51. Available from World Wide Web: <http://www.ercim.org/publication/Ercim_News/enw51/berners-lee.html>. [Accessed Mar 2006]

Binding, C., and Tudhope, D. (2005) KOS at your service: Programmatic Access to Knowledge Organisation Systems. *Journal of Digital information JoDI.* Vol. 4. Issue 4. Article No. 265. Available from World Wide Web: <http://jodi.ecs.soton.ac.uk/Articles/v04/i04/Binding/>. [Accessed Mar 2006]

Birbeck, M., Kay, M., Livingstone, S. Mohr, S., Pinnock, J., Loesgen, B., Martin, D., Ozu, N., Seabourne, M., and Baliles, D. (2001) *Professional XML.* Chicago: Wrox Press Ltd

Bloom, A., and Langley, P. (1997) Selection of relevant features and examples in machine learning. *Artificial Intelligence.* Vol. 97. No.1-2

Bosman, F., Bruza, P., van der Weide, T., and Weuseten, L. (1998) Documentation, cataloguing, and query by navigation: a practical and sound approach. In *Proceedings of the Second European Conference on Digital Libraries (ECDL'98).* Heraklion, Crete, Greece

Brailsford, T., Stewart, C., Zakaria, M., and Moore, A. (2002) Autonavigation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment. In *Proceedings of the Eleventh International World Wide Web Conference (WWW02).* Honolulu, Hawaii, United States

Brewster, C., and O'Hara K. (2004) Knowledge Representation with Ontologies: The Present and Future. *IEEE Intelligent Systems.* Vol. 19. No. 1

Brown, C. (1988) *Human-computer interface design guidelines.* Norwood: Ablex Publishing

Bruza P. (1990) Hyperindices: A Novel Aid for Searching in Hypermedia. In *Proceedings of the ECHT '90 European Conference on Hypertext, Databases, Indices and Normative Knowledge.* Paris, France

Buckley, R. (2005) New seekers. *M-iD: Managing Information and Documents.* Vol. November 2005. Available from World Wide Web: <http://www.the-word-is-not-enough.com/mid/gallery/new-seekers.php>. [Accessed Mar 2006]

Burke, R. (1999) Integrating Knowledge-Based and Collaborative-Filtering Recommender Systems. In *Proceedings of the AAAI Workshop on AI in Electronic Commerce.* Orlando, Florida, United States

Bush, V. (1945) As we may think. *Atlantic Monthly.* Vol. July 1945

Bush, V. (1967) Memex Revisited. In *Science is not enough.* Reprinted in J.M. Nyce and P. Kahn, (eds). *From Memex to Hypertext: Vannevar Bush and the Mind's Machine.* San Diego: Academic Press

Butler Group (2004) Enterprise Content Management: Strategies for Planning, Selection, and Deployment. In *Enterprise content Management, A Butler Group Symposium.* London, UK. Available from World Wide Web: <http://www.butlergroup.com/events/ecm/presentations.asp>. [Accessed Mar 2006]

Butler, M. (2004) Martin Butler Keynote. In: *Enterprise content Management. A Butler Group Symposium*. London, UK

Capra, R., and Perez-Quinones, M. (2005) Using Web Search Engines to Find and Refind Information. *IEEE Computer*. Vol. 38. No. 10

Carr, L., DeRoure, D., Hall W., and Hill, G. (1995) The Distributed Link Service: A Tool for Publishers, Authors and Readers. In *Proceedings of the Web Revolution: Fourth International World Wide Web Conference (WWW4).* Boston, Massachusetts, United States

Carr, L., DeRoure, D., Hall W., and Hill, G. (1996a) Open Linking Services. In *Proceedings of the Fifth International World Web Conference. Paris, France*

Carr, L., DeRoure, D., and Hill, G. (1996b) *Ongoing Development of an Open Links Service for the World-Wide Web*. Technical Report ECSTR M. Southampton: University of Southampton. Available from World Wide Web: <http://www.ecs.soton.ac.uk/~lac/tr1/tr1.html>. [Accessed Mar 2006]

Carr, L., Hall, W., and Miles-Board T. (2000a) *Writing and Reading Hypermedia on the Web*. Technical Report No. ECSTR-IAM00-1. Southampton: University of Southampton. Available from World Wide Web: <http://www.bib.ecs.soton.ac.uk/cgi-bin/data/3368/html/WRWH.html>. [Accessed Mar 2006]

Carr, L., Hall, W., and Miles-Board T. (2000b) Is the WWW Killing Hypermedia. In *Poster Proceedings of the Ninth International World Wide Web Conference (WWW9)*. Amsterdam, Netherlands. Available from World Wide Web: <http://www9.org/final-posters/poster55.html>. [Accessed Mar 2006]

Carr, L., Hall, W., Bechhofer, S., and Goble, C. (2001b) Conceptual linking: Ontology-based open hypermedia. In *Proceedings of the Tenth International World Wide Web Conference (WWW01)*. Hong Kong, China

Carr, L., Kampa, S., and Miles-Board, T. (2001c) *MetaPortal Final report: Building Ontological Hypermedia with the Ontoportal Framework*. Technical Report 6976. Southampton: University of Southampton

Cassola, E. (1998) *ProFusion PersonalAssistant: An Agent for Personalize Information Filtering on the WWW*. Master's thesis. Lawrence: University of Kansas

Chen, C. (1999) *Information Visualiasation and Virtual Environments*. London: Springer-Verlag

Chen, P. (1976) The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*. Vol. 1. No. 1

Cherry, S. (2002) Weaving a Web of Ideas. *IEEE Spectrum*. Vol. 39. No. 9

Chirita, P., Gavriloaie, R., Ghita, S., Nejdl, W., and Paiu, R. (2005) Activity based metadata for semantic desktop search. In *Proceedings of the 2nd European Semantic Web Conference*. Heraklion, Crete, Greece

Choo, W., Detlor, B., and Turnbull, D. (2000) *Web Work: Information Seeking and Knowledge Work on the World Wide Web*. Dordrecht: Kluwer Academic Publishers

Christophides, V., Karvounarakis, G., Plexousakis, D., Scholl, M., and Tourtounis, S. (2004) Optimizing Taxonomic Semantic Web Queries Using Labelling Schemes. *Web Semantics: Science, Services and Agents on the World Wide Web*. Vol. 1. No. 2. Available from World Wide Web: <http://citeseer.ist.psu.edu/707820.html>. [Accessed Mar 2006]

Clancey, J. (1997) *Situated Cognition, On Human Knowledge and Computer Representations*. Cambridge: Cambridge University Press

Claypool, M., Le, P., Waseda, M., and Brown, D. (2001) Implicit Interest Indicators. In *Proceedings of ACM Intelligent User Interfaces Conference (IUI)*. Miami, Florida, United States

Cleverdon, C., and Kean, M. (1968) *Factors Determining the Performance of Indexing Systems*. Cranfield: ASLIB

CODASYL (1971) Database Task Group Report. *The Conference on Data Systems Languages (CODASYL).* Box 12. Folder 17

Cohen, P., and Kjeldsen, R. (1987) Information Retrieval by Constrained Spreading Activation on Semantic Networks. *Information Processing and Management*. Vol. 23. No. 4

Cooley, R., Mobasher, B., and Srivastava, J. (1997) Web Mining: Information and Pattern Discovery of the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97).* Newport Beach, California, United States

Crestani, F. (1997) Application of Spreading Activation techniques in Information Retrieval. *Artificial Intelligence Review*. Vol. 11. No. 6

Croft, B. (1995) What Do People Want from Information Retrieval?. *D-Lib Magazine*. Vol. November 1995

Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2002) Probabilistic query expansion using query logs. In *Proceedings of the eleventh international conference on World Wide Web*. Honolulu, Hawaii, United States. Available from World Wide Web: <http://portal.acm.org/citation.cfm?id=511489>. [Accessed Mar 2006]

Cunliffe, D., Taylor, C., and Tudhope, D. (1997) Query-based Navigation in semantically Indexed Hypermedia. In *Proceedings of the ACM Hypertext 1997 Conference*. Southampton, UK. Available from World Wide Web: <http://www.journals.ecs.soton.ac.uk/~lac/ht97/pdfs/cunliffe.pdf>. [Accessed Mar 2001]

Cunliffe, D. (2000) Trailblazing: trends in hypermedia. *The New Review of Hypermedia and Multimedia*. Vol. 6.

Cyert, M., and March, G. (1963) *A Behavioral Theory of the Firm*. Oxford: Blackwell Publishers Ltd

Damjanovic, B., and Andric, M. (1997) A New Approach in Electronic Document Management. In *Proceedings of CAD Forum*. Novi Sad, Yugoslavia

Damjanovic, B., and Andric, M. (1998) Document Management System as a framework for Reengineering. *In Proceedings of 42nd conference of ETRAN*. Vrnjacka Banja, Yugoslavia

Davenport, T. H. (1997) *Information Ecology*. New York: Oxford University Press

Davis, H., Hall, W., Heath, I., Hill, G., and Wilkins, R. (1992) *MICROCOSM: An Open Hypermedia Environment for Informatics Integration*. Technical Report CSTR 92-15. Southampton: University of Southampton

Decker, S., *et al.* (2000) The Semantic Web: The roles of XML and RDF. *IEEE Internet Computing*. Vol. 4. No. 5

Dekkers M. (2000) Metadata Watch Report #3, London: PriceWaterhouseCoopers

Delphi (1998) Document Power. A special white paper report. London: Delphi consulting group

Deniman, D., Sumner, T., Davis, L., Bhushan S., and Fox, J. (2003) Merging Metadata and Content-Based Retrieval. *Journal of Digital information JoDI.* Vol. 4. Issue 3. Article No. 231

De Roure, D., Carr, R., Hall W., and Hill, G. (1996) A distributed hypermedia link service. In *Services in Distributed and Networked Environments. In Proceedings of Third International workshop on Services in Distributed and Networked Environments (SDNE 96).* Macau, China

De Roure, D., Hall, W., Reich, S., Pikrakis, A., Hill G., and Stairmand, M. (1998a) An Open Architecture for Supporting Collaboration on the Web. In *WET ICE '98 - IEEE Seventh International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.* Stanford University, California. United States

De Roure, D., Hall, W., Reich, S., Pikrakis, A., Hill G., and Stairmand, M. (1998b) An open framework for collaborative distribute information management. In *Proceedings of the Seventh International World Wide Web Conference (WWW7).* Brisbane, Australia

De Roure, D., Hall, W., Reich, S., Hill, G., Pikrakis, A., and Stairmand, M. (2001) MEMOIR – an open distributed framework for enhanced navigation of distributed information. *Information Processing and Management.* Vol. 37

Devedzic, V. (2003) Think ahead: evaluation and standardisation issues for e-learning applications. *International Journal of Continuing Engineering Education and Lifelong Learning.* Vol. 13, No. 5-6

Devedzic, V. (2004) Education and The Semantic Web. *International Journal of Artificial Intelligence in Education (IJAIED).* Vol. 14

De Young, L. (1990) Linking considered harmful. In *Proceedings of the ACM Hypertext 1990 Conference*. Washington, DC, United States

Dhar, V., and Stein, R. (1997) *Intelligent Decision Support Methods – The Science of Knowledge Work.* Upper Saddle River: Prentice Hall

Dieng, R., Corby, O., Giboin, A., and Ribiere, M. (1999) Methods and Tools for Corporate Knowledge Management. *International Journal of Human-Computer Studies (IJHCS)*. Vol. 51.

Dix, A., Finlay, L., Abowd, G., and Beale, R. (1998) *Human -Computer Interaction*, Hemel Hempstead: Prentice Hall

Domingue, J., Dzbor, M., and Motta E. (2004a) Magpie: Supporting Browsing and Navigation on the Semantic Web. In *Proceedings of the Semantic Web. Intl. Conference on Intelligent User Interfaces*. Madeira, Portugal

Domingue, J., Dzbor, M., and Motta E. (2004b) Semantic Layering with Magpie. In: S. Staab, and R. Studer. (eds). *Handbook on Ontologies in Information Systems.* London: Springer-Verlag

Domingue, J., Dzbor, M., and Motta E. (2004c) Collaborative Semantic Web Browsing with Magpie. In *Proceedings of the 1st European Semantic Web Symposium (ESWS)*. Heraklion, Crete, Greece

Dourish, P., Edwards W., LaMarca A., and Salisbury, M. (1999) Presto: an experimental architecture for fluid interactive document spaces. *ACM Transactions on Computer-Human Interaction.* Vol. 6. No. 2.

Dourish, P., Edwards, W., K., Howell, J., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry D., and Thornton, J. (2000) A programming model for active documents. *In Proceedings of the 13th annual ACM symposium on User interface software and technology*. San Diego, California, United States

Doyle, B. (2005) Metadata--Think outside the docs!. *EContent Magazine*. Article No. 7947. Available from World Wide Web: <http://www.econtentmag.com/Articles/ArticleReader.aspx?ArticleID=7947>. [Accessed Mar 2006]

Dumais, S., *et al.* (2003) Stuff I've seen: A System for Personal Information Retrieval and Reuse. In *Proceedings of the ACM SIGIR 2003*. Toronto, Canada.

Dzbor, M., Domingue, B., and Motta, E. (2003) Magpie - towards a semantic Web browser. In *Proceedings of the 2nd Intl. Semantic Web Conference*. Florida, United States.

El-Beltagy, S., Hall, W., De Roure, D., and Carr, L. (2001) Linking in Context. *In Proceedings of the ACM Hypertext 2001 Conference*. Arhus, Denmark

Engelbart, D. (1962) Augmenting human intellect: A conceptual framework, Report AFOSR-3233. Menlo Park: Stanford Research Institute. Available from World Wide Web: <http://sloan.stanford.edu/mousesite/EngelbartPapers/B5_F18_ConceptFrameworkInd.html>. [Accessed Mar 2006]

Engelbart, D. (1963) A Conceptual framework for the augmentation of man's intellect. *Vistas in Information Handling*. Vol. 1

Engelbart, D. (1995) Towards augmenting the human intellect and boosting our collective IQ. *Communications of the ACM*. Vol. 38. No. 8

Evans, D., and Zhai, C. (1996) Noun-Phrase Analysis in Unrestricted Text for Information Retrieval. *In Proceedings of the 34th Annual Meetings of the Association for Computational Linguistics*. Santa Cruz, California, United States

Fagin, R., Kumar, R., McCurley, S., Novak, J., Sivakumar, D., Tomlin, J., and Williamson, P. (2003) Searching the Workplace Web. In *Proceedings of the Twelfth International World Wide Web Conference (WWW03)*. Budapest, Hungary

Fensel, D., Horrocks, I., van Harmelen, F., Decker, S. Erdmann, M., and Klein, M. (2000) OIL in a Nutshell. In: R. Dieng *et al.* (eds). Knowledge Acquisition, Modeling, and Management. In *Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*. Juan-les-Pins, France

Fensel, D., van Harmelen, F., *et al.* (2000) On-to-knowledge: Ontology-based tools for knowledge management. In *Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference*. Madrid, Spain

Foltz, P., and Dumais, S. (1992) Personalized Information Delivery: An Analysis of Information Filtering Methods. *Communications of the ACM*. Vol. 35. No. 12

Fountain, A., Hall, W., Heath, I., and Davis H. (1990) MICROCOSM: An Open Model for Hypermedia With Dynamic Linking. In: A. Rizk, N. Streitz and J. Andre (eds). Hypertext: Concepts, Systems and Applications, *The Proceedings of The European Conference on Hypertext*. Versailles, France

Fowler, J., Cohen, L., and Jarvis, P. (1998) *Practical Statistics for Field Biology*. Chichester: John Wiley & Sons

Furnas, G., Deerwester, S., Dumains, S., Landauer, T., Harshman, R., Streeter, L., and Lochbaum, K. (1998) Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure. In *Proceedings of the Eleventh International Conference on Research & Development in Information Retrieval.* Grenoble, France

Garcia, E. (2005) The Keyword Density of Non-Sense. *E-marketing news*. Vol. March 2005. Available from World Wide Web: <http://www.e-marketing-news.co.uk/Mar05/garcia.html>. [Accessed Mar 2006].

Gibbins, N., Harris, S., Dix, A., and schraefel, m. (2004) *Applying mSpace interfaces to the Semantic Web - working paper*. Technical Report 0027. Southampton: University of Southampton

Gilheany, S. (2001) Brief Overview of Document Management. *Archive Builder's Analyses newsletter for document management*. Vol. 5. No. 5

Ginsburg, M. (200) Intranet Document Management Systems as Knowledge Ecologies. In Proceedings of the 33rd Hawaii International Conference on System Science (HICSS 2000). Maui, Hawaii, United States

Glance, N., S., Arregui, D., and Dardenne, M. (1998) Knowledge pump: Supporting the flow and use of knowledge. In: U. Borghoff and R. Pareschi (eds). *Information technology for management*. Berlin: Springer-Verlag

Glance, N., S., and Grasso, a. (2000) Collaborative document monitoring via a recommender system. In *Proceedings of International Workshop on Agent-Based Recommender Systems, Fourth International Conference on Autonomous Agents*. Barcelona, Spain

Goble, C., Bechhofer, S. Carr, L. De Roure, D., and Hall, W. (2001) Conceptual Open Hypermedia = The Semantic Web?. In *SemWeb2001 The Second International Workshop on the Semantic Web.* Hong Kong

Goldberg, D., Nichols, D., Oki, M., and Terry, D. (1992) Using collaborative filtering to weave an information tapestry. *Communications of the ACM*. Vol. 35. No. 12

Golshani, F., and Dimitrova, N. (1994) Retrieval and delivery of information in multimedia database systems. *Information and Software Technology*. Vol. 36. No. 4

Goraj, S. (editor) (1999) *Java 2 Complete*. San Francisco: Sybex

Greif, I. (editor) (1998) *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo: Morgan Kaufmann

Griffiths, J., and Andric, M. (1998) MAGENTA – **M**EMOIR **A**ssisted **G**uided tours **EN**gineered from **T**rails using **A**gents. Software Demonstration at *ACM Hypertext 1998 Conference*. Pittsburgh, PA, United States

Gruber, T. (1993) A translation approach to portable ontology specifications. *Knowledge Acquisition*. Vol. 5. No. 2

Guarino N., Masolo C., and Vetere G. (1999) OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems*. Vol. 14. No. 3

Guha, R., McCool, R., and Miller, E. (2003) Semantic Search. In *Proceedings of the Twelfth International World Wide Web Conference (WWW03)*. Budapest, Hungary

Hang C., Wen J.-R., Nie J.-Y., and Ma W.-Y. (2002) Probabilistic Query Expansion Using Query Logs. In *Proceedings of the Eleventh International World Wide Web Conference (WWW02)*. Honolulu, Hawaii, United States

Harris-Jones, C. (2002) Specifying your requirements and selecting your supplier for Content Presentation and Delivery. *Content Management Briefing*. London: Info

Heflin, J., and Hendler, J. (2001) A Portrait of The Semantic Web in Action. *IEEE Intelligent Systems*. Vol. 16. No. 2

Hendley, T., and Broadhurst, R. (2000) *Document Management Guide and Directory*. A Comprehensive Guide to Document Management and a Directory of Products and Services. Hertfordshire: Cimtech Limited, University of Hertfordshire

Hendley T. (2005) Managing information and documents, The definitive guide. *M-iD Magazine*. Vol. April 2005. Available from World Wide Web: <http://www.doconsite.co.uk>. [Accessed Mar 2006]

Herlocker, J., Kostan J., Borchers, A., and Riedl, J. (1999) An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*. New York, NY, United States

Herlocker, J. (2004) Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*. Vol. 22. No. 1

Hill, G., Hutchings, G., James, R., Loades, S., Hale J., and Hatzopulous, M. (1997) Exploiting Serendipity Amongst Users To Provide Support for Hypertext Navigation. In *Proceedings of the ACM Hypertext 1997 Conference*. Southampton, UK

Himmelstein, M. (2005) Local Search: The Internet Is the Yellow Pages. *IEEE Computer*.Vol. 38. No. 2

Horrocks, I. *et al.* (2000) *The Ontology Inference Layer OIL*. Technical report, Amsterdam: Free University of Amsterdam. Available from World Wide Web: <http://citeseer.ist.psu.edu/horrocks00ontology.html>. [Accessed Mar 2006]

Hotchkiss, G. (2005) I Love to Search but Words Get in the Way. *Search Engine Guide*. Vol. June 2005

Hughes, G. (2003) *An Open Hypermedia Link Service Architecture Supporting Multiple Context Models*. Phd. Thesis. Southampton: University of Southampton

Jennings, T. (2004) *Holistic Content Management*. Butler Group White Paper. London: Butler Group

Joachims, T., Freitag D., and Mitchell, T. (1997) WebWatcher: A Tour Guide for the World Wide Web. In *Fifteenth international Joint Conference on Artificial Intelligence (IJCAI-97)*. Nagoya, Japan

Jones, S., Gatford, M., and Hancock-Bealieu, M. (1994) Support Strategies for Interactive Thesaurus, In: H. Albrechtsen and A. Oernager (eds). *Knowledge Organization and Quality Management*. Copenhagen: Ergon Verlag

Kamolvilassatian, N., Vazguez, M., Taankand, S., and Banerjee, S. (2001) *Multi-Domain Recommender System: Utilizing Ontology Information and Two-Level User Model Representation in Personalization*. Technical Report. Austin: University of Texas

Kampa, S., Miles-Board, T., Carr, L., and Hall, W. (2001) *Linking with meaning: Ontological hypertext for scholars* Technical Report 0-854327-37-1. Southampton: University of Southampton. Available from World Wide Web: <http://www.bib.ecs.soton.ac.uk/data/5163/PDF/lwm.pdf>. [Accessed Mar 2006]

Karypis, G. (2001) Evaluation of Item-based Top-N Recommendation Algorithm. In *Proceedings of the tenth international conference on Information and knowledge management*. Atlanta, Georgia, United States. Available from World Wide Web: <http://portal.acm.org/citation.cfm?id=502627>. [Accessed Mar 2006]

Keen, M. (1971) Evaluation Parameters. In G. Salton (eds.), *The Smart Retrieval System-Experiment in Automatic Document Processing*. Englewood Cliffs: Prentice Hall

Kietz J.-U., Maedche A., and Volz R. (2000) A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. In: N. Aussenac-Gilles, B. Bibow, and S. Szulman (eds). *EKAW'2000 Workshop Ontologies and Texts*, Juan-les-Pins, France

Kim, H.-L., Kim, H.-G., and Park, K.-M. (2004) Ontalk: Ontology-Based Personal Document Management System. In *Proceedings of the Thirtheenth International World Wide Web Conference (WWW04)*. New York, NY, United States

King, M. et al. (1995) EAGLES Evaluation of Natural Language Processing Systems. FINAL REPORT EAGLES DOCUMENT EAG-EWG-PR.2. Version of September 1995. Geneva: EAGLES Evaluation Working Group. Available from World Wide Web: <http://www.issco.unige.ch/ewg95/>. [Accessed Mar 2006]

Kleinberg, K. (1999) Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*. Vol. 46. No. 5

Kolari, P., and Joshi, A. (2004) Web Mining: Research and Practice. *Web Engineering*. Vol. 6. No. 4

Kruchten, P. (2001) *The Rational Unified Process – An Introduction*. Boston: Addison-Wesley

Lamping, J., Rao, R., and Pirolli, P. (1995) A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95, ACM Conference on Human Factors in Computing Systems*. New York, NY, United States

Lancaster, F., W. (1968) *Evaluation of the Medlars Demand Search Service*, Washington: U.S. Dept. of Health, Education, and Welfare, Public Health Service

Laundauer T., and Dumais. S. (1997) A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*. Vol. 104

Lewis, P., Hall, W., Carr, L., and De Roure, D. (1999) The Significance of Linking, *ACM Computing Surveys*. Vol. 31. No. 4. Available from World Wide Web: <http://www.cs.brown.edu/memex/ACM_HypertextTestbed/papers/20.html>. [Accessed Mar 2006]

Linden, G., Smith, B., and York, J. (2003) Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*. Vol. 7. No. 1.

Linton F. (1998) OWL: A Recommender System for Organization-Wide Learning, MITRE Technical Report MTR 98B0000025V00S00R00. MITRE

Linton, F., Joy, D., Schaefer, H., and Charron, A. (2000) OWL: A Recommender System for Organization-Wide Learning, *Journal of International Forum of Educational Technology & Society and IEEE Learning Technology Task Force*. Vol. 3. No.1. Available from World Wide Web: <http://ifets.ieee.org/periodical/vol_1_2000/linton.html>. [Accessed Mar 2006]

Lowe, D., and Hall, W. (1999) *Hypermedia and the Web, an Engineering Approach.* Chichester: John Wiley & Sons

Maltz, D., and Ehrlich K. (1995) Pointing the way: active collaborative filtering. In *Conference on Human Factors in Computing Systems (CHI95)*. Denver, Colorado, United States

Maurer, H. (1998) Web-Based Knowledge Management. *IEEE Computer*. Vol. 31. No. 3

Mayfield, J., and Nicholas, C. (1993) SNITCH: Augmenting Hypertext Documents with a Semantic Net. *International Journal of Intelligent and Cooperative Information Systems.* Vol. 2. No. 3. Available from World Wide Web: <http://www.cs.umbc.edu/~mayfield/pubs/ijicis93.ps>. [Accessed Mar 2006]

McGrath, A. (2003) *An Insight into Enterprise Content Management*. Whitepaper. London: Unilog. Available from World Wide Web: <http://www.unilog.co.uk/WhitePapers/WhitePapers.aspx#>. [Accessed Apr 2005]

McGuffin, M., and schraefel, m. (2004) A comparison of hyperstructures: zzstructures, mSpaces, and polyarchies. In *Proceedings of the ACM Hypertext 2004 Conference*. Santa Cruz, California, United States

McLaughlin, L. (2004) What's Next in Web Search?. *IEEE Distributed Systems Online*. Vol. 5. No. 11. Available from World Wide Web: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1377089>. [Accessed Mar 2006]

Melville, P., Mooney, R., and Nagarajan, R. (2001) Content-boosted collaborative filtering. In *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*. New Orleans, LA, United States

Middleton, S., De Roure, D., and Shadbolt, N. (2001) Capturing knowledge of user preferences: ontologies in recommender systems. In *Proceedings of The first International Conference on Knowledge Capture (K-CAP-2001)*. Victoria, B. C., Canada

Middleton, S., Alani, H., Shadbolt, N., and De Roure, D. (2002) Exploiting synergy between ontologies and recommender systems, International Workshop on the Semantic Web. In *Proceedings of the Eleventh International World Wide Web Conference (WWW02)*. Honolulu, Hawaii, United States

Middleton, S., De Roure, D., and Shadbolt, N. (2004) Ontological User Profiling in Recommender Systems. *ACM Transactions on Information Systems*. Vol. 22. No.1

Mika, P., Iosif, V., Sure, Y., and Akkermans, H. (2003) Ontology-based Content Management in a Virtual Organization. In: S. Staab and R. Studer (eds.) *Handbook on Ontologies in Information Systems.* London: Springer-Verlag

Miles-Board, T., Kampa, S., Carr, L., and Hall, W. (2001) Hypertext in the Semantic Web. In *Proceedings of the ACM Hypertext 2001 Conference*. Arhus, Denmark. Available from World Wide Web: <http://eprints.ecs.soton.ac.uk/6975/ >. [Accessed Mar 2006]

Mittal, A., Dixit, S., and Maheshwari L. (2003) Enhanced Understanding and Retrieval of E-learning Documents through Relational and Conceptual Graphs. In *Workshop on Technologies for Electronic Documents for Supporting Learning held in conjunction with the Artificial Intelligence in Education (AIED), 11th International Conference on Artificial Intelligence in Education.* Sydney, Australia

Mladenic, D. (1996) Personal WebWatcher: Design and Implementation, Technical Report IJS-DP-7472, School of Computer Science. Pittsburgh: Carnegie-Mellon University. Available from World Wide Web: <http://citeseer.ist.psu.edu/mladenic96personal.html>. [Accessed Mar 2006]

Mladenic, D. (1999) Text-Learning and Related Intelligent Agents: A survey. *IEEE Intelligent Systems*. Vol. 14. No. 4

Mobasher, B. *et al.* (2001) Effective Personalization Based on Association Rule Discovery from Web Usage Data. In *Proceedings of the 3rd ACM Workshop Web Information and Data Management (WIDM 2001).* Atlanta, Georgia, United States

Mobasher, B., Jin, X., and Zhou, Y. (2003) Semantically Enhanced Collaborative Filtering on the Web. In *Web Mining: From Web to Semantic Web, First European Web Mining Forum*, EMWF 2003. Cavtat-Dubrovnik, Croatia

Moore, A., and Brailsford T. (2004) Unified Hyperstructures for Bioinformatics: Escaping the Application Prison. *Journal of Digital Information JoDI.* Vol. 5. Issue 1. Article No. 254

Moore, A., Goulding, J., Brailsford, T., and Ashman, H. (2004) Practical Applitudes: Case Studies of Applications of the ZigZag Hypermedia System. In *Proceedings of the ACM Hypertext 2004 Conference*. Santa Cruz, California, United States

Nelson, T. (1965) A File Structure for the Complex, the Changing and the Indeterminate. In *Proceedings of the 1965 20th ACM National Conference*. Cleveland, Ohio, United States

Nelson, T. (1967) Getting it out of our system. In *Information Retrieval: A Critical View*. G. Shector, (editor), Washington: Thompson Book Co.

Nelson, T. (1981) *Literary Machines*. early editions (1981 to 1986). Published by the author

Nelson, T. (1998) What's On My Mind. Invited talk at *The first Wearable Computer Conference*. Fairfax, United States. Available from World Wide Web: <http://www.xanadu.com.au/ted/zigzag/xybrap.html>. [Accessed Mar 2006]

Nelson, T. (2001) Deeper Cosmology, Deeper Documents (Technical Briefing). In *Proceedings of the ACM Hypertext 2001 Conference*. Arhus, Denmark. Available from World Wide Web: <http://portal.acm.org/citation.cfm?id=504281>. [Accessed Mar 2006]

Nelson, T. (2003a) Structure, Tradition and Possibility. In *Proceedings of the ACM Hypertext 2003 Conference*. Nottingham, UK

Nelson, T. (2004) A Cosmology for a Different Computer Universe: Data Model, Mechanisms, Virtual Machine and Visualization Infrastructure. *Journal of Digital Information JoDI.* Vol. 5. Issue 1. Article No. 298

Nielsen, J. (1993) *Hypertext & Hypermedia*. Boston: AP Professional

NISO (2004) Understanding Metadata. *Information Standards Quarterly (ISQ)* Vol. 18. Available from World Wide Web: <http://www.niso.org/standards/std_resources.html>. [Accessed Mar 2006]

Olstad, B. and Seres, S. (2005) A Roadmap to Search as a Strategic Enabler. *Supplement to KM world*. Vol. April 2005

Ossenbruggen, J., Hardman, L., and Rutledge, L. (2002) Hypermedia and the Semantic Web: A Research Agenda. *Journal of Digital information* JoDI. Vol. 3. Issue 1. Article No. 55

Oxnard, L., and Evans A. (2003) *Methodologies for the Automatic Location of Academic and Educational Texts on the Internet*. WORKING PAPER 03/01. Leeds: University of Leeds

Papadimitriou, C., Raghavan, P., Tamaki, H., and Vempala, S. (1998) Latent semantic indexing: a probabilistic analysis. In *Proceedings of 17th ACM Symposium Principles of Database Systems*. Seattle, Washington, United States

Paulson, L. (2005) Using Topic Maps to Improve Searches. *IEEE Computer*. Vol. 38. No. 5

Pepper, S. (2000) The TAO of Topic Maps, finding the way in the age of infoglut. *XML Europe 2000*. Paris, France. Available from World Wide Web: <http://www.gca.org/papers/xmleurope2000/papers/s11-01.html>. [Accessed Mar 2006]

Perugini, S., and Goncalves, M. (2002) *Recommendation and personalization: a survey*. Technical Report cs.IR/0205059. Computing Research Repository. Blackburg: Virginia Tech. Available from World Wide Web: <http://xxx.lanl.gov/abs/cs.IR/0205059>. [Accessed Mar 2006]

Perugini, S., Goncalves, M. and Fox. E. A. (2002) A Connection-Centric Survey of Recommender Systems Research. *CoRR: Information Retrieval*

Pikrakis, A., Bitsikas, T., Sfakianakis, S., Hatzopoulos, M., De Roure, D., Hall, W., Reich, S., Hill G., and Stairmand, M. (1998) MEMOIR – Software Agents for Finding Similar Users by Trails. In PAAM98, *The Third International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents*. London, UK

Pinkerton, B. (1994) Finding What People Want: Experiences with the WebCrawler. In *Proceedings of the Second International World Wide Web Conference (WWW2)*. Chicago, Illinois, United States. Available from World Wide Web: <http://www.thinkpink.com/bp/WebCrawler/WWW94.html>. [Accessed Mar 2006]

Poget, M. (1999) Using Semantic Nets to Model Troubleshooting's Knowledge. *Troubleshooting Professional Magazine: Troubleshooting CGI. Vol. 3. Issue 7*

Pokorny, J. (2004) Web Searching and Information Retrieval. *IEEE Computing in Science and Engineering*. Vol. 6. No. 4

Popkin, G., and Cushman, A. (eds) (1993) *Integrated Document Management - Controlling a Rising Tide: Office Information Systems Strategic Analysis Report*. Gartner Group RAS Services

Porter, M. (1980) An algorithm for suffix stripping, Program. *Automated Library and Information Systems*. Vol. 14. No. 3

Preece, J., Rogers, Y., Sharp, H., Benyion, D., Holland S., and Carey, T. (1994) *Human-Computer Interaction*. Reading: Addison-Wesley

Preece, A., and Decker, S. (2002) Intelligent Web Services. *IEEE Intelligent Systems*. Vol. 17. No. 1

Puchinger, J., and Raidl, G. (2005) Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation*. Vol. 3562

Quillian, M. (1968) Semantic Memories. In M. M., Minsky, (editor), *Semantic Information Processing*. Cambridge: MIT Press

Resnick, P., Iacovou, N., Suchak, M., Bergstron, P., and Riedl, J. (1994) GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of CSCW 94 Conference on Computer Supported Cooperative Work*. New York, NY, United States. Available from World Wide Web: <http://www.si.umich.edu/~presnick/papers/cscw94/GroupLens.htm>. [Accessed Mar 2006]

Resnick, P., and Varian, H. (1997) Recommender Systems. In *Communications of the ACM*. Vol. 40. No. 3

Rocha, C., Schwabe, D., and Poggi de Aragao, M. (2004) A hybrid approach for searching in the semantic web. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW04)*. New York, NY, United States

Robertson, R., Cameron, K., Czerwinski, M., and Robbins, D. (2002a) Polyarchy Visualization: Visualizing Multiple Intersecting Hierarchies. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'02.*, Minneapolis, MN, United States

Robertson, R., Cameron, K., Czerwinski, M., and Robbins, D. (2002b) Animated Visualization of Multiple Intersecting Hierarchies. *Information Visualization*. Vol. 1

Rocchio, J. (1971) *Relevance Feedback in Information Retrieval*. In G. Salton (editor). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewood Cliffs: Prentice Hall

Rose, D., and Levinson, D. (2004) Understanding user goals in web search. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW04)*. New York, NY, United States

Russell, S., and Norvig, P. (1995) *Artificial Intelligence – A Modern Approach.* London: Prentice Hall

Salton, G. (1968) *Automatic Information Organization and Retrieval.* New York: McGraw-Hill

Salton, G., Wong, A. and Yang, C.S. (1975) A Vector Space Model for Automatic Indexing. *Communications of the ACM.* Vol. 18. No. 11

Salton, G., and McGill, J. (1983) *Introduction to Modern Information Retrieval.* New York: McGraw-Hill

Salton, G., and Buckley, C. (1988) Term weighting approaches in automatic text retrieval. *Information Processing and Management.* Vol. 24. No. 5

Salton, G. (1989) Automatic *Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading: Addison-Wesley

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000) Application of Dimensionality Reduction in Recommender System: A Case Study, In *Proceedings of ACM SIGKDD 2000.* Boston, MA, United States

Schafer, J., Konstan, J., and Riedl, J. (1999) Recommender systems in e-commerce, In *Proceedings of the first ACM Conference on Electronic Commerce.* Denver, Colorado, United States

schraefel, m., Carr, L., De Roure, D., and Hall, W. (2004) You've Got Hypertext. *Journal of Digital Information JoDI.* Vol. 5. Issue 1. Article No. 253. Available from World Wide Web: <http://citeseer.ist.psu.edu/671609.html>. [Accessed Mar 2006]

schraefel, m., Smith, D., Russel, A., Owens, A., Harris, C., and Wilson, M. (2005a) The mSpace Classical Music Explorer: Improving Access to Classical Music for Real People. *In Proceedings of V MUSICNETWORK OPEN WORKSHOP: Integration of Music in Multimedia Applications*. Vienna, Austria

schraefel, m., Smith, D., Owens, A., Russell, A., and Harris, C. (2005b) The evolving mSpace platform: leveraging the Semantic Web on the Trail of the Memex. In *Proceedings of the ACM Hypertext 2005 Conference*. Salzburg, Austria

Schrein, A., Popescul, A., and Ungar, H. (2002) Methods and Metrics for Cold-Start Recommendations. In *Proceedings of the ACM SIGIR 2002*. Tampere, Finland

Shahabi, C., and Chen Y. (2003) Web Information Personalization: Challenges and Approaches. In *Proceedings of The 3nd International Workshop on Databases in Networked Information Systems (DNIS 2003)*. Aizu-Wakamatsu, Japan. Available from World Wide Web: <http://infolab.usc.edu/DocsDemos/DNIS2003.pdf>. [Accessed Mar 2006]

Shapiro, S. (1987) Semantic networks. In S. C. Shapiro *(editor) Encyclopedia of Artificial Intelligence.* New York: John Wiley & Sons

Shipman, M., Hsieh, H., Maloor, P., and Moore M. (2001) The Visual Knowledge Builder: A Second Generation Spatial Hypertext. In *Proceedings of the ACM Hypertext 2001 Conference*. Arhus, Denmark

Shipman, F., Moore, M., Maloor, P., Hsieh, H., and Akkapeddi, R. (2002) Semantics Happen: Knowledge Building in Spatial Hypertext. In *Proceedings of the ACM Hypertext 2002 Conference*. Maryland, United States

Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. (1998) *Analysis of a very large altavista query log*. Technical Report SRC 1998-014. DEC Research Center

Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. (1999) Analysis of a very large Web Search Engine query log. *SIGIR Forum*. Vol. 33. No. 3

Singh G., *et al.* (2003) A Metadata Catalog Service for Data Intensive Applications. In *Proceedings of the ACM SC 2003*. Phoenix, Arizona, United States. Available from World Wide Web: <http://www.globus.org/alliance/publications/papers/mcs_sc2003.pdf>. [Accessed Mar 2006]

Sowa, J. (1984) *Conceptual Structures: Information processing in Mind and Machine*, Reading: Addison-Wesley

Spertus, E., and Stein, L. (1998a) A Hyperlink-Based Recommender System Written in Squeal. *CIKM'98 Workshop on Web Information and Data Management (WIDM'98)*. Bathesda, Maryland, United States. Available from World Wide Web: <http://www.mills.edu/ACAD_INFO/MCS/SPERTUS/widm98.pdf>. [Accessed Jan 2004]

Spertus, E., and Stein, L. (1998b) Mining the Web's Hyperlinks for Recommendations. *AAAI-98 Workshop on Recommender Systems*. Madison, Wisconsin, United States. Available from World Wide Web: <http://www.mills.edu/ACAD_INFO/MCS/SPERTUS/recommender-workshop.pdf>. [Accessed Jan 2004]

Spyratos, N., Tzitzikas, Y., and Christophides, V. (2002) On Personalizing the Catalogs of Web Portals. In *Proceedings of the 15th International FLAIRS'02 Conference*. Florida, United States

Srivastava, J., Cooley, R., Deshpande, M., and Tan, P. (2000) Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations, ACM SIGKDD*. Vol. 1. No. 2

Steels, L. (1993) Corporate Knowledge management. In *Proceedings of ISMICK'94*. Compiegne, France

Stefik, M. (1995) *Introduction to Knowledge Systems*. San Francisco: Morgan Kaufmann

Steinacker, A., Ghavam, A., and Steinmetz, R. (2001) Metadata Standards for Web-based Resources. *IEEE Multimedia.* Vol. 10. No.1

Stenmark, D. (1997) *Searching Intranet Using Knowledge Representation – Experiences with the SearchEnhancer*. Internal report. Viktoria Institute, University of Sweden. Available from World Wide Web: <http://w3.informatik.gu.se/~dixi/publ/snet.htm>. [Accessed Mar 2006]

Stenmark, D. (2000) Collaborative Aspects of Information Retrieval Tools: Summarising three action case studies. In *Proceedings of IRIS 23*. Uddevala, Sweden. Available from World Wide Web: <http://www.viktoria.se/results/result_files/142.pdf>. [Accessed Mar 2006].

Stenmark, D. (2003) Query expansion using an intranet-based semantic net. In *Proceedings of IRIS-26*. Haikko Manor, Sweden

Stenmark, D. (2005) Query expansion on a corporate intranet: Using LSI to increase relative precision in explorative search. In *Proceedings of the 38th Hawaii International Conference on System Science (HICSS 2005).* Island of Hawaii, Hawaii, United States

Stojanovic, N., Struder R., and Stojanovic, L. (2003) An Approach for the Ranking of Query Results in the Semantic Web. In *Proceedings of 2nd International Semantic Web Conference (ISWC '03)*. Sanibel Island, Florida, United States

Swearingen, K., and Sinha, R. (2001) Beyond Algorithms: An HCI Perspective on Recommender Systems. In *Procedings of the SIGIR workshop on Recommender Systems*. New Orleans, LA, United States

Terveen, L., Hill, W., Amento, B., McDonald D., and Creter, J. (1997) PHOAKS: A System for Sharing Recommendations. *Communications of the ACM.* Vol. 40. No. 3

Terveen, L., and Hill, W. (2001) Beyond Recommender Systems: Helping People Help Each Other. In J. Carrol (editor) *HCI In the Millennium.* Reading: Addison-Wesley

Tobin, T. (2003) *Ten Principles of Knowledge Management Success.* White Paper. BizReport IT Research Library. Knova software. Available from World Wide Web: <http://research.bizreport.com/detail/RES/1063624180_419.html&src=TRM_TOPN>. [Accessed Mar 2006]

Towle, B., and Quinn, C. (2000) Knowledge Based Recommender Systems Using Explicit User Models. In *AAAI Workshop.* Austin, Texas, United States. Available from World Wide Web: <http://www.igec.umbc.edu/kbem/final/towle.pdf>. [Accessed Mar 2006]

Trigg, R., and Weiser, M. (1986) TEXTNET: A Network-based Approach to Text Handling. *ACM Transactions on Office Information Systems (TOIS).* Vol. 4. No. 1

Tudhope, D., and Cunliffe, D. (1999) Semantically Indexed Hypermedia: Linking Information Disciplines. *ACM Computer Surveys.* Vol. 31. No. 4. Available from World Wide Web: <http://www.cs.brown.edu/memex/ACMCSHT/6/6.html>. [Accessed Mar 2006]

Tudhope, D., Alani, H., and Jones, C. (2001) Augmenting thesaurus relationships: possibilities for retrieval. *The Journal of Digital Information JoDI.* Vol. 1. Issue 8. Article No. 41. Available from World Wide Web: <http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Tudhope/>. [Accessed Mar 2006]

Turban, E. (1995) *Decision Support and Expert Systems- Management Support Systems*. Englewood Cliffs: Prentice Hall

Twidale, M., and Nichols, D. (1998) *A Survey of Applications of CSCW for Digital Libraries*. Technical report VSeg/4/98 Computing department. Lancalster: Lancaster University. Available from World Wide Web: <http://www.comp.lancs.ac.uk/computing/research/cseg/projects/ariadne/docs/survey.html>. [Accessed Mar 2006]

Tzitzikas, Y., Spyratos, N., Constantopoulos, P., and Analy, A. (2002) Extended Faceted Taxonomies for Web Catalogs. *ERCIM News*, Special Theme: Semantic Web. No. 51. Available from World Wide Web: <http://www.ercim.org/publication/Ercim_News/enw51/tzitzikas.html>. [Accessed Mar 2006]

van Heijst, G., van der Spek, R., and Kruizinga, E. (1996) Organizing Corporate Memories. In *Proceedings of KAW'96*. Banff, Canada

Wen, J.-R., Nie J.-Y., and Zhang, H.-J. (2000) Clustering User Queries of a Search Engine, In *Proceedings of the Ninth International World Wide Web Conference (WWW9)*. Amsterdam, Netherlands

Wilson, M., Russell, A., Smith, D., Owens, A., and schraefel, m. (2005) mSpace Mobile: A Mobile Application for the Semantic Web. *End User Semantic Web Workshop ISWC2005*. Galway, Ireland

Winston, P. (1993) *Artificial Intelligence*. Reading: Addison-Wesley

Wohl, A. (2005) Interface Lift. *IEEE Spectrum.* Vol. 42. No. 11

Wolber, D., Kepe, M., and Ranitovic, I. (2002) Exposing Document Context in the Personal Web. In *Proceedings of the 7th ACM International Conference on Intelligent User Interfaces*. San Francisco, California, USA. Available from World Wide Web: <http://portal.acm.org/citation.cfm?id=502740>. [Accessed Mar 2006]

Wooldridge, M. (1998) Agent based computing. *Interoperable Comm. Networks*, Vol. 1. No. 1

Yang, Y. (1999) An Evaluation of Statistical Approaches to Text Categorisation, *Journal of Information Retrieval,* Vol. 1 No. 1-2

Yang J., Lin T., and Wu K. (2002) An Agent-Based Recommender System for Lesson Plan Sequencing. In *IEEE International Conference on Advanced Learning Technologies (ICALT 2002).* Kazan, Tatarstan, Russia

Yeates, S. (1999) *Novel Index and Metadata Sources in Digital Libraries*. Thesis Proposal. Waikato: University of Waikato. Available from World Wide Web: <http://citeseer.ist.psu.edu/yeates99novel.html>. [Accessed Mar 2006]

Online References

Berners-Lee, T., Masinter, L., and McCahill, M. (1994) *RFC1738: Uniform Resource Locators* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://rfc.net/rfc1738.html>

Berners-Lee, T. (1998a) *What the Semantic Web can represent* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.w3.org/DesignIssues/RDFnot.html>

Berners-Lee, T. (1998b) *Realising the Full Potential of the Web* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.w3.org/1998/02/Potential.html>

Berners-Lee, T. (2004) *How It All Started*, presentation materials from the W3C 10th Anniversary Celebration [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.w3.org/2004/Talks/w3c10-HowItAllStarted/>

Bray, T. (2003) *On Search, the Series* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.tbray.org/ongoing/When/200x/2003/07/30/OnSearchTOC>

Brickley, D. and Guha, R. (2004) *RDF Vocabulary Description Language 1.0: RDF Schema* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.w3.org/TR/rdf-schema>

Carr, L. (2000c) *Links and Queries in the COHSE Project*, Internal Project Report [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://cohse.semanticweb.org/discuss/000504_Links_Vs_Queries_Paper/links_v_queries.html>

Carr, L. (2001a) *ZigZag for Web Browsers* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.ecs.soton.ac.uk/~lac/zigzag>

Ervasti, K. (2001) *Instructions for the Use of GzigZag* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://gzigzag.sourceforge.net/ug/newug.html>

Girschweiler, B. (eds) (1992) *What is HyperText*, World Wide Web Consortium [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.w3.org/WhatIs.html>

Hillmann, D. (2005) *Using Dublin Core* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://dublincore.org/documents/usageguide>

Kelly, M. (1998) *Z39.50 Resource Page* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.niso.org/standards/resources/Z3950_Resources.html>

Klyne, G., Carroll, J., and McBride, M. (eds) (2004) *Resource Description Framework (RDF): Concepts and Abstract Syntax* [online]. [Accessed Mar 2006]. Available from World Wide Web World Wide Web: <http://www.w3.org/TR/rdf-concepts/>

Lagoze, C., Van de Sompel, H., Nelson, M., and Warner, S. (eds) (2002) *The Open Archives Initiative Protocol for Metadata Harvesting v2.0* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.openarchives.org/OAI/openarchivesprotocol.html>

Lukka, T. (2002) *A Gentle Introduction to Ted Nelson's ZigZag tructure* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.nongnu.org/gzz/gi/gi.html>

Lukka, T. and Ervasti, K. (2003) *GzigZag – A Platform for Cybertext Experiments* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://gzigzag.sourceforge.net/ct/ct.html>

McGuffin, M. (2004) *A graph-theoretic introduction to Ted Nelson's zzstructures* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.dgp.toronto.edu/~mjmcguff/research/zigzag/>

McLellan, T. (1997) *An Introduction to Usenet News* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.islandnet.com/~tmc/html/articles/usentnws.htm>

Microsoft (2006a) *Microsoft SharePoint Portal Server 2001 Resource Kit* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.microsoft.com/technet/prodtechnol/sppt/sharepoint/reskit/part7/z04spprk.mspx>

Microsoft (2006b) *Microsoft Office 2000/Visual Basic Programmer's Guide: Working with Document Properties* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odeopg/html/deovrworkingwithdocumentproperties.asp>

Nelson, T. (1999) *Ted Nelson's Computer Paradigm, Expressed as One-Liners* [online]. [Accessed Feb 2002]. Available from World Wide Web: <http://www.sfc.keio.ac.jp/~ted/TN/WRITINGS/TCOMPARADIGM/tedCompOneLiners.html>

Nelson, T. (2000) *ZigZag Tech Notes* [online]. [Accessed Mar 2001]. Available from World Wide Web: <http://www.xanadu.com/zigzag/zzTech.html>

Nelson, T. (2003b) *ZZZ, Our Working Version of ZigZag®* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://xanadu.com/zigzag/ZZdnld/zzzWriteup.html>

Raggett, D., Le Hors, A. and Jacobs, I. (eds) (1997) *HTML 4.0 Specification: HTMLs and URLs*, World Wide Web Consortium [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.w3.org/TR/WD-html40-970708/htmlweb.html>

Sherman, C. (2002) *Why Search Engines Fail*, In SearchDay [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://searchenginewatch.com/searchday/article.php/2160661>

Wolf, G. (1995) *The Curse of Xanadu*, In *Wired Magazine* [online]. Available from World Wide Web: <http://www.wired.com/wired/3.06/xanadu.html>

Weibel, S., *et al.* (1998) *Dublin Core Metadata for Resource Discovery (rfc2413 memo)* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://rfc2413.x42.com/>

Whitehead J. (1996) *Orality and Hypertext: An Interview with Ted Nelson* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.ics.uci.edu/~ejw/csr/nelson_pg.html>

Ziff-Davis (1998) *Ted Nelson, Hypertext Pioneer* [online]. [Accessed Mar 2006]. Available from World Wide Web: <http://www.g4tv.com/techtvvault/features/4605/Ted_Nelson_Hypertext_Pioneer.html>

Appendix A

A.1 Overview of Commercial Enterprise Content Managament Systems

Open Text was a pioneer in Web-based document management systems, with its software called **LiveLink**. The system originated from the academic environment. Open Text was founded in 90ties as a spin-off from the University of Waterloo in Ontario, Canada. Livelink includes a range of collaborative and knowledge management features, such as on-line meetings, shared workspaces and integrated e-mail. It is well know for a very high scalability. Also, Livelink supports a range of people-to-people and people-to-information tools, working with structured and unstructured data.

Documentum (now **EMC**), with a product called Documentum 5, is one of the oldest and most renowned document management solution providers since the 90ties. Currently, they are developing a concept of Information Lifecycle Management, enabling the management of unstructured content from creation through to archiving and disposal. Documentum 5 comprises a very broad range of technologies together with unique features: strong API and links to enterprise applications such as SAP and Siebel.

Lotus Notes comprises a product family of groupware collaborative applications. Its Web server **Lotus Domino** provides features for document management on the Web including a range of format conversion and publishing tools.

Fujitcu, one of the largest IT products and services company, offers a content management solution **Interstage Content Integrator** that uses object repository called Enabler. Enabler has built in metadata layer and features integrated relationship management service that allows users to create and store links between documents. Also, the Content Integrator includes a range of standard document/content management functionality such as change control, workflow, document search and federated Web search.

Hummingbird DM is a multi-tiered Hummingbird's document management solution with the integrated collaboration environment. Documents can be placed into the hierarchical structures, folders or nested folders. The complementary software product **Hummingbird KM** offers information search and categorisation solution. It uses taxonomies and neural network classifiers in order to reduce manual categorisation. The search features unified seeking into file repositories, document management system, Websites and multimedia libraries. Searching features 'find more like this' capability. The typical attributes to search by comprise: author, title, subject, keyword, date, size, as well as relevance, source and by concept. The tool features search term highlighting and saved searches.

Red Dot's **Content Collaboration Server (CSS)** comprises collaboration, document management and workflow solution. Document management solution comprises virtual hierarchy of folders while physical files are in so called vaults spread over the multiple disks and machines. The system features document cataloguing, version control, check in/out, automated trails and access control. Four types of search are available:

- Content search (search text, Boolean conditions and proximity searches)
- Item search (by author, owner or date)
- User and group search (search for users or groups)
- Catalogue search (search for files using any associated user definable metadata)

Autonomy is an established leader in the area of automatic classification of content against a predefined taxonomy. Autonomy have merged in 2005 with Verity, who were another leader in the unstructured data management. Their system called **Autonomy** performs automated cross referencing and hyperlinking into the existing content on the Website. A component called Intelligent Data Operating Layer uses a range of artificial intelligence techniques to analyse the content and provide add-on knowledge management facilities, such as Bayesian inference and neural networks.

It supports conceptual search, automated clustering and data visualisation. Autonomy claim that their system can identify concepts and find related documents not only based on the word similarity. This provider has a strong links to another well known content management provider – Vignette. Vignette's system **Vignette V7** provides Website personalisation and interaction management in addition to standard CM features.

The company called Hyperwave provides and integrated management framework, **Information Server release 6 (IS/6)**. IS/6 integrates many disparate, yet related functions, such as content management, document management, information discovery and retrieval, as well as a unique feature of e-learning, into one collaborative environment.

Cambridge (UK) based company Neurascript's system **INDICIUS** focuses on converting electronic document images into meaningful information. It is capable of extracting information from practically any type of document.

Vamosa's **Content Migrator** system enhances the content by automated addition of extended metadata, addition or removal of hyperlinks and de-duplication of content.

Appendix B

B.1 AWOCADO User Interaction Walkthrough

After logging into the AWOCADO's User Website, the user is presented with a list of documents in his/her private, working zone, called "WorkZone". Documents are listed in a tabular form with a fixed number of the most important attributes like "Document Type" (document class) and Priority, as given in figure B-1.

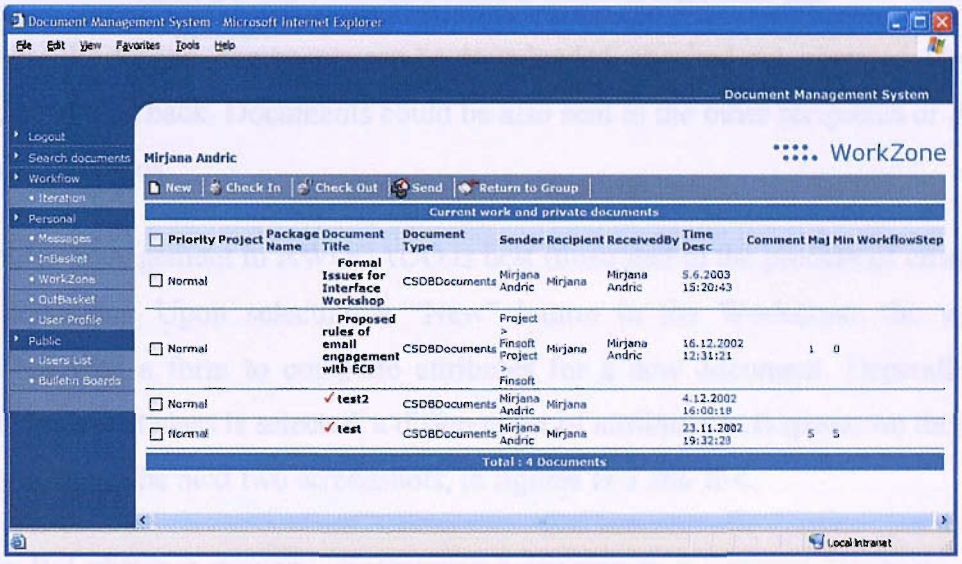


Figure B-1: The AWOCADO example: list of documents in the WorkZone

Following a link by clicking on a "Document title" attribute brings up the document details page where the full list of attributes is shown. Attributes of a selected document are presented in the upper half while the document's manipulation history is shown in the lower half of the page in figure B-2.

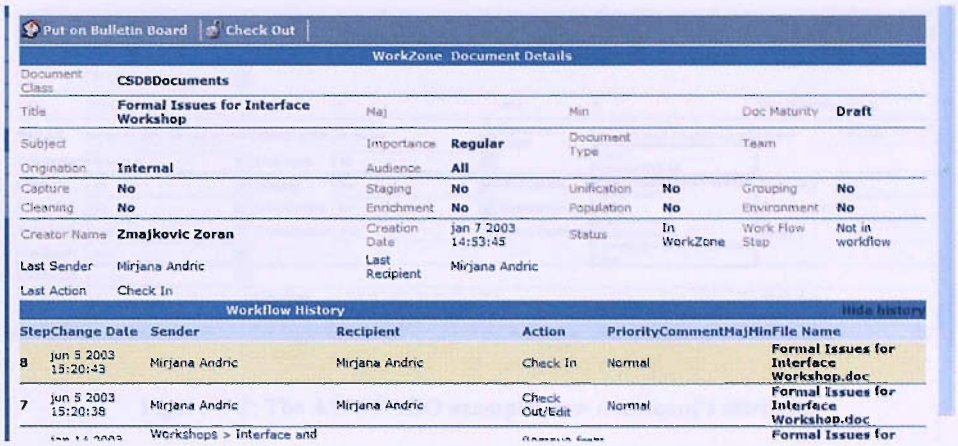


Figure B-2: The AWOCADO example: document's attributes page

From the WorkZone, documents can be downloaded, checked out, changed, checked in and uploaded back. Documents could be also sent to the other recipients or placed on bulletin boards.

Attributes management in AWOCADO is best illustrated in the process of creating a new document: Upon selecting a "New" button in the WorkZone, the user is presented with a form to complete attributes for a new document. Depending on which document class is selected, a different set of attributes will appear on the form, as is shown on the next two screenshots, in figures B-3 and B-4.

Figure B-3 shows a number of attributes belonging to a selected document class, which is in this example "EDQMRiskControl". The figure also shows how the controlled vocabulary attribute "Document Type" can be selected from a lookup list.

Figure B-4 illustrates that when another document class, in this example "Iteration Package" was selected, a set of attributes was also changed. This class has 13 attributes, some of the same type and some different comparing to the document class in figure B-3.

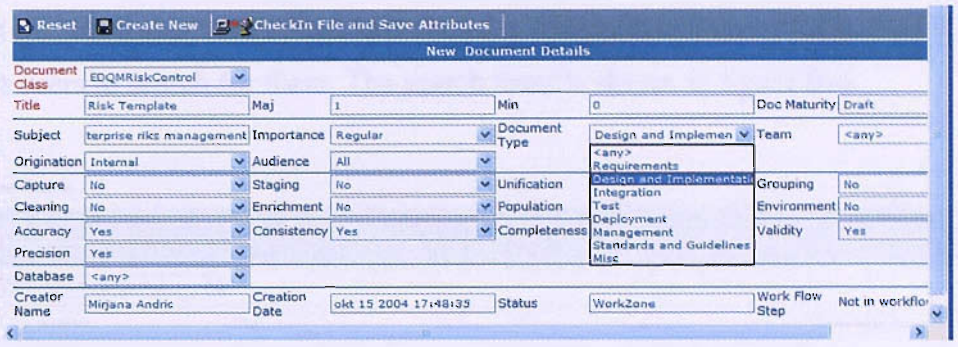


Figure B-3: The AWOCADO example: new document's attributes

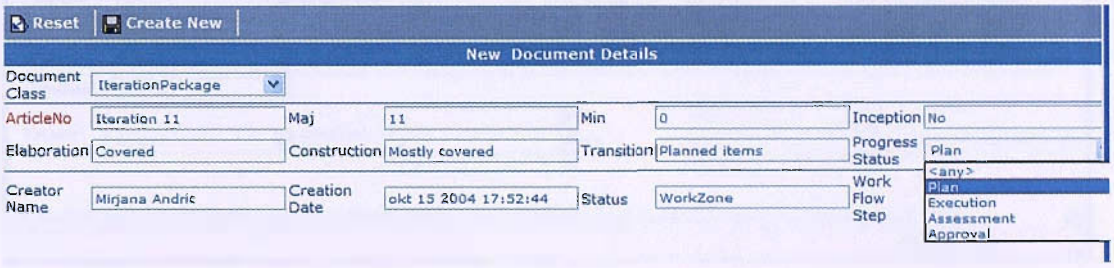


Figure B-4: The AWOCADO example: new document, another document class

The number and type of attributes for each document class is set via the admin site as shown in figure B-5, in this example a document class "EDQMRiskControl".

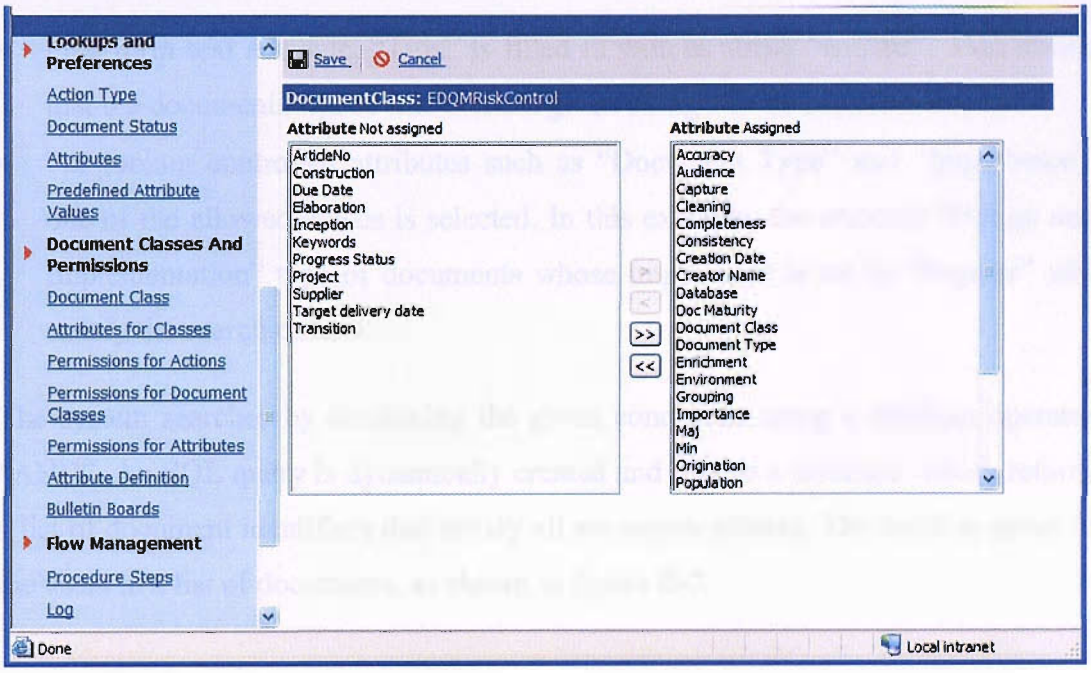


Figure B-5: The AWOCADO admin example: assigning attributes to a document class

When the documents are edited and placed in the system's repository, AWOCADO allows users to search for them. The search form is shown in figure B-6.

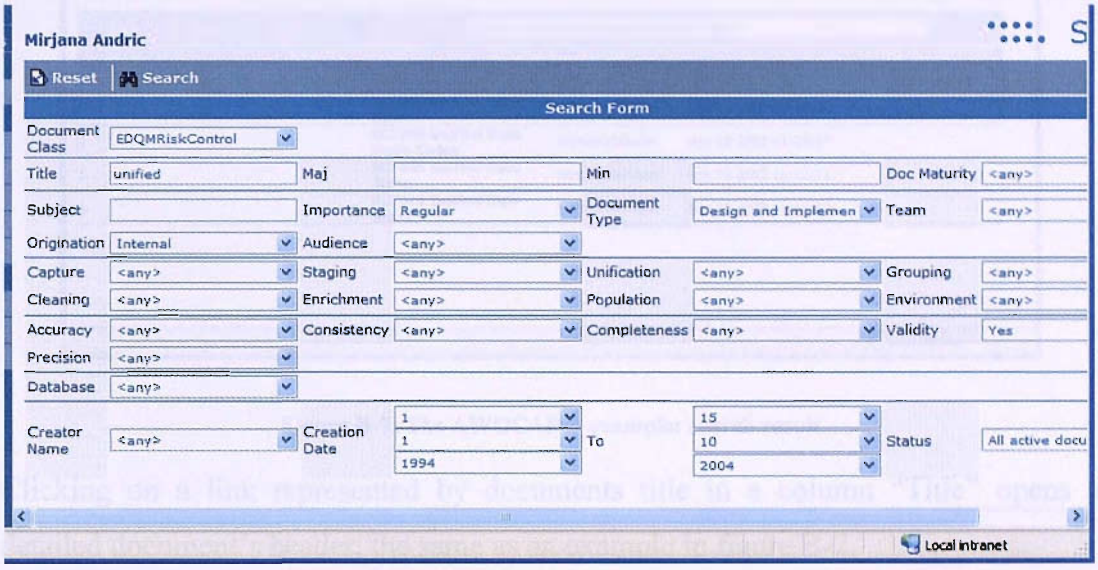


Figure B-6: The AWOCADO example: searching by metadata

When the document class is selected, a set of attributes by which to search is changed. In the example in figure B-6, the user has provided values for the following attributes:

- Free form text attribute, "Title" is filled in with as string "unified". This means that the documents whose title contain given string will be searched for;
- For lookup controlled attributes such as "Document Type" and "Importance", one of the allowed values is selected. In this example, the selected "Design and Implementation" type of documents whose importance is set to "Regular" will satisfy the search criteria.

The system searches by combining the given conditions using a Boolean operator "AND". An SQL query is dynamically created and sent to a database, which returns a list of document identifiers that satisfy all set search criteria. The result is given to the users in a list of documents, as shown in figure B-7.

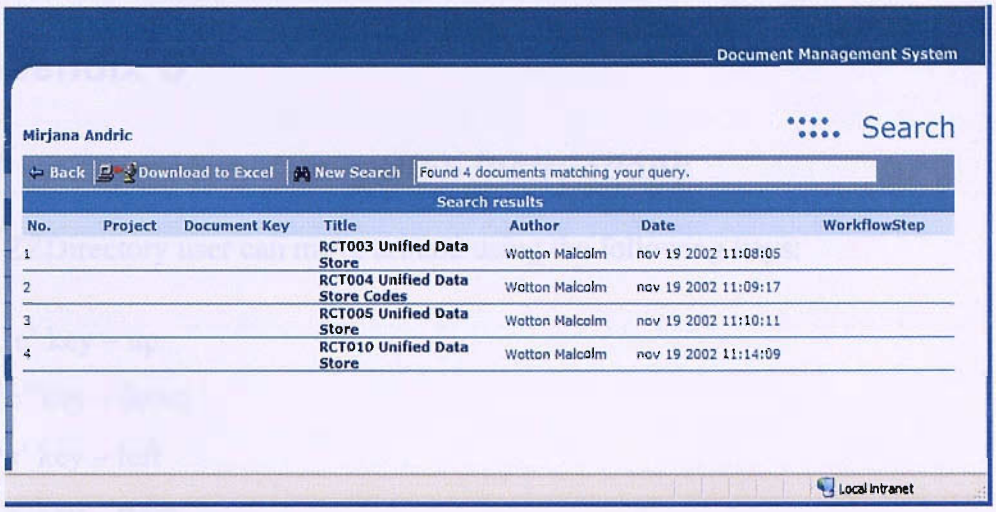


Figure B-7: The AWOCADO example: search result

Clicking on a link represented by documents title in a column "Title" opens a detailed document's header, the same as an example in figure B-2.

Appendix C

C.1 ZZDirectory Navigation Instructions

The ZZDirectory user can move around using the following keys:

- 'e' key – up
- 'c' key – down
- 's' key – left
- 'f' key – right
- 'x' – change to the next Across dimension
- 'y' – change to the next Down dimension
- Clinking the arrow below the cell – Change Across dimension and make the cell current
- Clicking the arrow left of the cell – Change Down dimension and make the cell current

The user can always return to the starting position if the page is reloaded (right click+Refresh).

C.2 ZZDirectory Demo Instructions

The ZZDirectory demo is prepared for the Windows and IE5+ browser environment. The usage instructions are as follows:

1. Place all files in the same directory

2. Double click ODPExample-2frames.htm>

3. Leave the left window on the main hierarchy (showing y:'Top' and x:'Sublevel') and change the Across dimension to become 'Newsgroups' in the right window by clicking on the leftmost black arrow just bellow the 'Top/Health'

4. Change the Across dimension to become 'Related' by clicking on the

rightmost black arrow just bellow the 'Top/Health'

5. Click again the rightmost arrow just below the same cell and you will be presented with the most interconnected dimension 'Symbolic'. You will have to scroll right to place again your current cell in the window centre

6. Observe symbolic cell with a title 'Reading', id: 'Top\Arts\Books', symbolically under 'Top\Recreation'. Let's see where is it in its main hierarchy. Click on any arrow left of the cell and you will see in the vertical dimension where is it placed in the 'Top\Arts', while still retaining horizontal 'Symbolic' dimension.

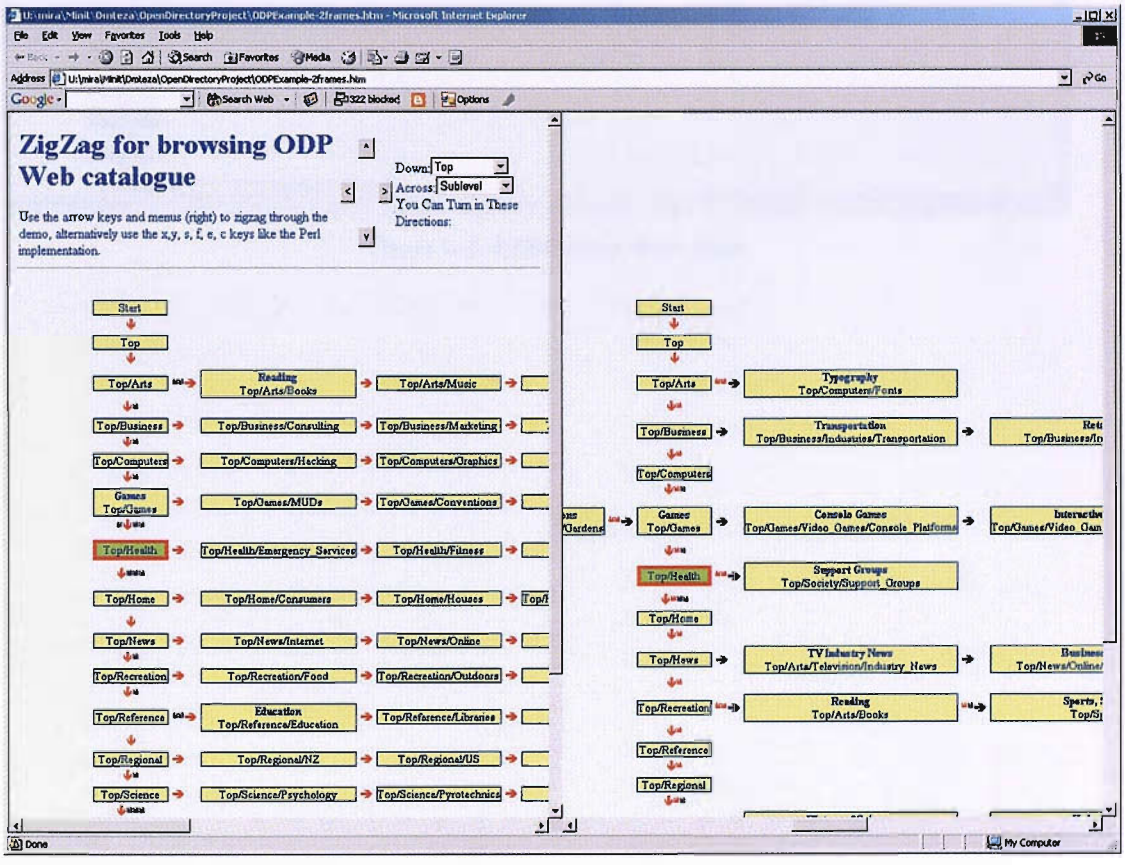


Figure C-1: ZZDirectory demo, starting step

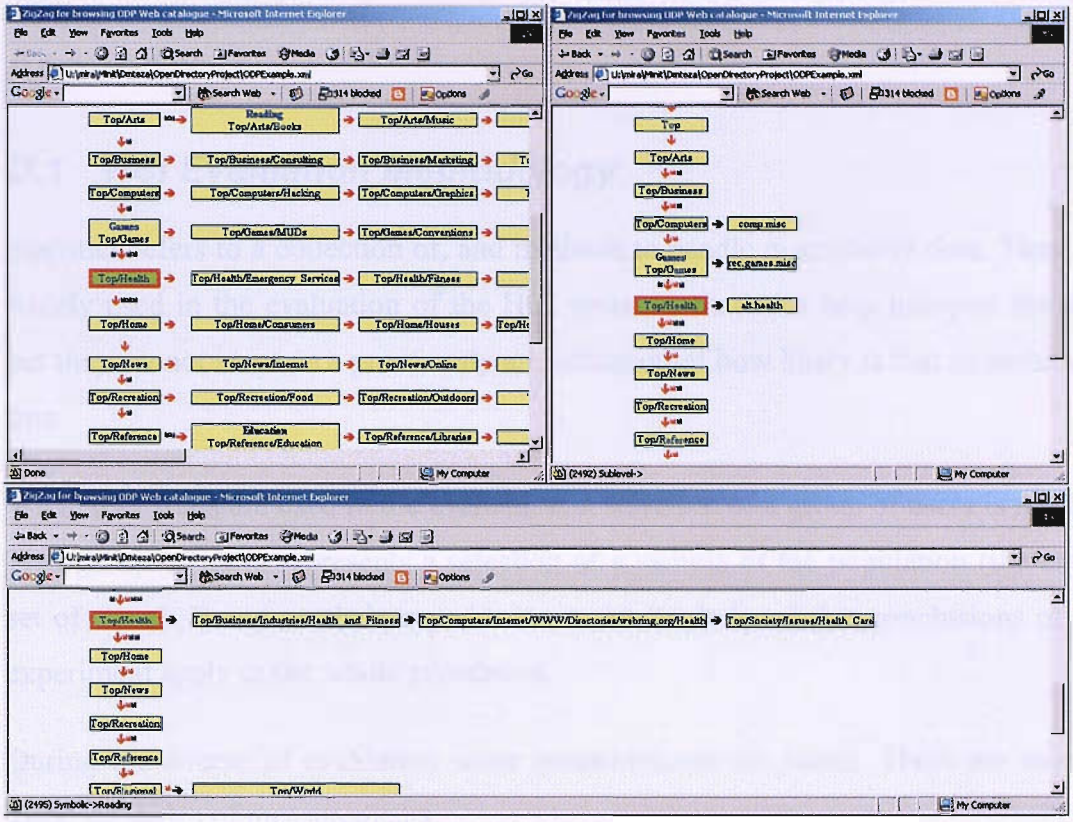


Figure C-2: ZZDirectory demo steps

Appendix D

D.1 HCI Evaluation Methodology

Statistics refers to a collection of, and methods to handle quantitative data. They are widely used in the evaluation of the HCI systems. Statistics help interpret the data but they cannot serve as a proof, only an indication of how likely is that something is true.

When user trials are used in the evaluation, a selection of a group of users is needed. That group of users represents a selection of a sample of the population (the entire set of users). Based on their experiences it can be deduced that conclusions of the experiment apply to the whole population.

During the course of evaluation some measurements are taken. There are several different scales of measurements:

- Nominal scales: a simple naming or classification scale (such as 1, 2, 3 or A, B, C), where mathematical operations can not be applied. 1 or 2, for example, are just codes for some class of events.
- Ordinal (rank) scales: Numbers on this scale can be ordered, so a greater number indicates magnitude of the feature measured.
- Interval scales: this scales takes the notion of ranking items in order one step further. The numbers are ordered and the intervals between consecutive points on a scale are of equivalent value.
- Ratio scales: it has all the properties of the interval scale in addition to having an absolute or a fixed zero point. Also, the ratio of numbers on this scale reflects the ratios of the feature measured.

Based on which scales are used, only certain kinds of statistical tests can be employed on the collected measurements. Statistical test designs can be classified into two broad categories: parametric and non-parametric. Parametric tests are used with interval and ratio scales. In other cases, non-parametric tests must be used.

The preconditions that need to be satisfied if parametric tests are to be used are:

- The selection of subjects from the population was random and independent, meaning that subjects have an equal chance to be selected and that selection of one of the subjects in no way influences the sampling of any other subjects.
- The observations were drawn form a normally distributed population.
- The variance of each set of scores must be comparable: this is known as homogeneity of variance.

The assumptions for non-parametric tests are:

- The selection of subjects from the population was random and independent.
- For scales of measurements ordinal or higher, the variance of each set of scores must be comparable.

Non-parametric tests make no assumptions about the particular distribution. They are more resilient to the outliers but less powerful.

User participation evaluations are divided into two categories:

- Laboratory studies (users are performing the cases in the predefined environment)
- Field studies (conducted in the natural users' environment)

Experiments involve two types of variables: independent and dependent. Typically, one or more independent variables are manipulated (changed) and the effect observed (measured) on the dependent variable(s).

D.2 Selecting the Appropriate Statistical Test

Dix and others (Dix *et al.* 1998) identify the following two rules of statistical analysis:

- Look at the data: Observe graphs and tables, identify outliers.
- Save the data: Preserve the original results in order to be able to try different statistical analysis.

The selection of statistical analysis depends on the type of data and the questions that are possibly asked about the data (Dix *et al.* 1998):

- Is there a difference? Is, for example, one system better than another? The technique addressing this is called *hypothesis testing* and the answers provided are in a form of for example 99% certainty that the answer is "yes" or "no".
- How big is the difference? This is called *point estimation*, often obtained by averages.
- How accurate is the estimate? Usually the standard deviation of the estimate or confidence intervals, give an answer to this question.

An aid to selecting the most appropriate statistical test for analysing data is given in the flow chart diagram in (Foster 2001).

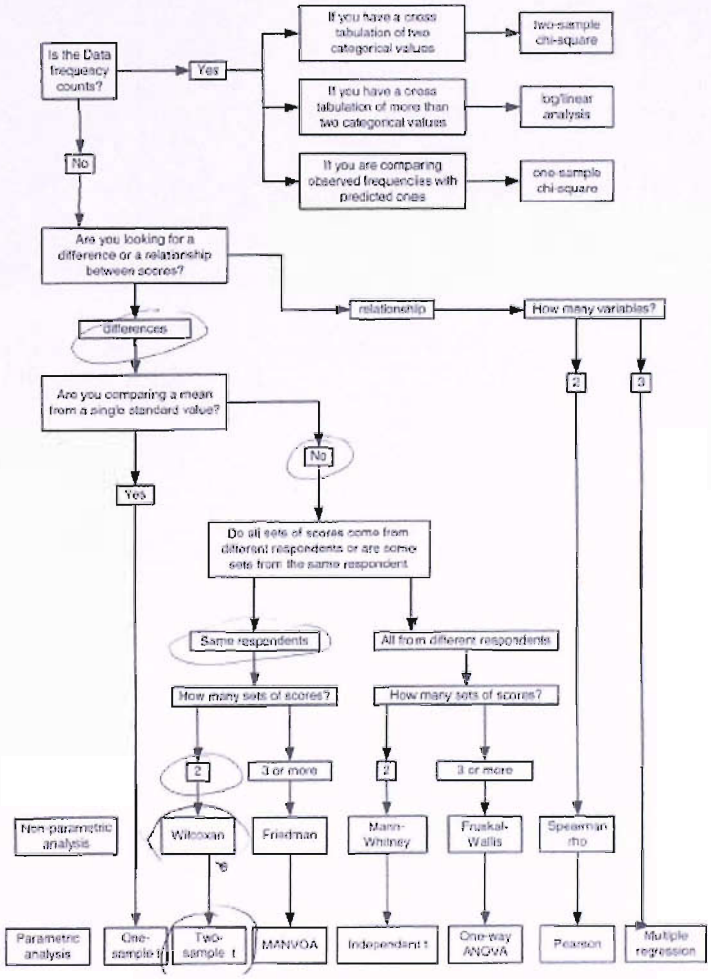


Figure D-1: Selecting the correct analysis technique, after (Foster 2001)