THE MULTIMEDIA THESAURUS: ADDING A SEMANTIC LAYER TO MULTIMEDIA INFORMATION

By

Robert Hugh Tansley

B.Sc. (Hons)

A thesis submitted for the degree of

Doctor of Philosophy

Department of Electronics and Computer Science,

University of Southampton,

United Kingdom.

August 2000

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING

ELECTRONICS AND COMPUTER SCIENCE DEPARTMENT

Doctor of Philosophy

THE MULTIMEDIA THESAURUS: ADDING A SEMANTIC LAYER TO MULTIMEDIA INFORMATION

by Robert Hugh Tansley

The digital computer has greatly increased our capacity for storing and accessing information. The internet, and particularly the World Wide Web, have vastly increased the amount of information available to us. Information retrieval and hypermedia research has greatly reduced the time and effort required to fulfill a searcher's information need; however, problems still remain. To access *multimedia* information, text metadata must usually be assigned to the multimedia objects, requiring an (often prohibitively) large amount of time and effort. Alternatively, some systems use low-level features of the media to allow a searcher to find objects similar to a query object. Such features cannot always identify two media objects depicting or representing the same real-world concept; in the case of images, different camera angles and lighting conditions mean that semantically similar images are visually very different. Additionally, semantic relationships between objects may not be expressed.

This thesis presents a way of addressing these problems by drawing on the field of semiology, in which a symbolic object has two parts: That which is being represented (in the *plane of content*), and the symbolic object doing the representing (in the *plane of expression*). A semantic layer is introduced holding real-world concepts, and connections to the multimedia objects that represent them. Knowledge about these concepts can be introduced by connecting them with semantic relationships. A prototype multimedia information system incorporating a semantic layer feature, the *multimedia thesaurus*, is introduced. The construction and use of a museum application is described, and used to evaluate the semantic layer technique. Finally, some reflections on these findings and some suggested future directions for the work are presented.

Contents

List of Figures

List of Tables

Acknowledgements

Firstly I'd like to thank my supervisor, Professor Wendy Hall, for her expert guidance and keeping me focussed. Gratitude is also due to Paul Lewis for constant support and encouragement, and for many enlightening debates.

I'd also like to thank Mark Weal, for proof reading, useful comments, being the voice of reason and for countless beers; Mark Dobie, for co-writing MAVIS 2, and nights out at Ikon; Danius Michaelides, for being a LaTeX and UNIX guru, and entertaining co-drinker (a rare combination); Dan Joyce, always willing to share his extensive knowledge, for useful comments on parts of my thesis, and never allowing a social event to be dull; Stephen Perry, Joseph Kuan, Ian Heath, Dave Dupplaw, Simon Kampa, Tim Miles-Board, Steve Blackburn, Guillermo Power and Gareth Hughes for keeping the workplace (and various pubs) interesting; Colin Bird, for encouragement, always being willing to help, and for great assistance with integrating QBIC; and Stevan Harnad, for giving me a job so I can pay my bills while I finish up.

Thanks must also go to Alex Bailey, whose presence guarantees a good night out, for many much-needed laughs, and my sister Claire, who's also done the PhD thing, for excellent nights out in Reading. Special thanks are also due to my girlfriend, Julie, for making weekends a true escape.

Finally, the biggest thank-you of all must go to my parents, always there for me, for making everything possible.

Chapter 1

Introduction

1.1 Computers and Information

Since the inception of the computer, its potential for easing tasks has been well recognised and exploited. It is intrinsicly suited to repetitive, methodical, formulaic tasks, and can perform these at high speeds. The time many tasks used to take has been dramatically reduced by the computer, as has the amount of time people have to spend on tedious tasks. A computer can assist and in some cases replace a human worker. In addition, the computer can perform many functions never before possible at all.

Computers are used to store and access vast amounts of information. Virtually every modern business uses computers extensively in its day-to-day running, and research institutions and governments hold large collections of scientific, social and economic information. Since a computer is an essentially numerical machine and can only work with numbers, the media that conveys the information itself must be stored as encoded sequences of numbers. These encoded sequences are referred to as *data*. Thus, in order to store information, it is necessary to numerically encode it.

Commonly, a human user requires some piece of information from the computer, and wishes to retrieve that information from the computer. This task has traditionally been known as *information retrieval*. The computer has some store of information, and holds an index to it. The user specifies some criteria for identifying relevant material; this is known as issuing a *query*. The computer then uses that query to determine what in its store of data is *relevant* to that query, and what is not.

As computers have become exponentially more powerful and capacious, their effectiveness for this task has also increased. A growing body of research had resulted in the availability of a large number of highly effective text information retrieval techniques.

As well as capacity and speed of computation, two further revolutions have occurred, increasing the usefulness of the computer.

1.1.1 Multimedia Information

Text, being a symbolic and discrete medium, is the most easily stored and manipulated form of information besides numerical data. Currently, information retrieval systems largely operate solely on text, and it is still by far the most common medium in which to store information in a computer.

However, modern computers now have considerable capacity for storing and presenting information in other media. This phenomenon has been called *multimedia computing*. As an example, a museum may now store not only textual and numerical information about their exhibits and archives in a computer system, but they may also store photographs and videos of the exhibits themselves. In this way, such an institution may build up a far more comprehensive base of information, and enjoy some of the versatility and speed of access that a computer system allows.

1.1.2 The Internet

Computers can now also intercommunicate to perform tasks. Data can be transmitted between computers to provide access to data held at a different geographical location to the user. Although the idea is hardly new, it is only recently that communication standards and infrastructure have allowed computers to network on a massive, global scale.

Known as the *internet*, this set of standards and infrastructure allows the rapid transmission and receipt of data, including multimedia data. A number of protocols and standards have been produced to exploit this network:

Electronic Mail is received minutes or seconds after is has been sent, rather than taking days as with surface or air mail;

The File Transfer Protocol allows the transmission and receipt of arbitrary computer files;

The World-Wide Web is a standard for the storage, transmission and navigation of primarily text-based information, that can include multimedia content.

The latter of these, the World-Wide Web (Bernes-Lee, 1996), or Web for short, is based on a slightly different information discovery paradigm to the information retrieval model. This paradigm is known as *hypertext*, or in the more general multimedia case, *hypermedia*.

The Web consists of *documents*, that can exist on a computer anywhere on the internet. Documents may be connected to other related documents via *links*; a user *follows* these links, *navigating* around the documents until they find a document that fulfills their need.

The information retrieval paradigm is still heavily used in the Web. *Search Engines* are text information retrieval systems that index documents from the Web. Searchers query these engines, as they would any other information retrieval system, and documents satisying that query are then presented to the searcher as a dynamically-produced Web document. This is known as *content-based retrieval*, or CBR, since documents are offered based on the content of those documents.

1.2 The Problem

Even though there is a huge amount of information available on an unprecedented scale, it is still frequently hard to find the precise piece of information needed. If the user issues a simple text query, the one relevant document may be buried within 10,000 irrelevant ones, and the user has to issue several progressively more refined queries to cut the list down to a palatable size. If the information the user requires is not held textually, it is even harder to find.

The problem is twofold. Firstly, how does one specify ones information need to the computer? Secondly, given that information need, how can the computer determine what will satisfy it?

Text retrieval systems are mature in this area. Relatively complex queries can be specified, and scalable algorithms used to retrieve appropriate documents based on their textual content with a reasonably high degree of success. The challenge in this case is the construction of the query.

Multimedia information imposes far greater demands on computer systems than text alone, both in terms of storage and computation. Many forms of media, for example images and audio, are inherently analogue media. In order to capture such media in a digital computer, an approximation must be made. Images are stored as matrices of brightness and/or colour values, known as *pixels*. Audio is stored as sequences of samples. In order for the approximation to be undetectable enough by humans to be usable, the matrices of pixels and sequences of samples must be very large, and hence require a large amount of storage space.

Computation of multimedia information is also considerably more complex than in the case of text. Text is a discrete, symbolic medium; other media, such as images, are non-discrete and require significant cognitive work to gain information from. While there is a thriving research community working on image processing and understanding, the problem remains largely unsolved. Those techniques for retrieving multimedia information that achieve any significant degree of success are always dependent on a particular domain and set of preconditions. A more common practice is to treat multimedia objects as 'black boxes', and associate with each some text *metadata*. Text retrieval and manipulation techniques may the be applied to the metadata. However this requires that at some point the metadata is manually assigned to the multimedia objects by a human, in many cases a prohibitively lengthy and laborious task.

There are also problems with the Web system. Links between related documents must be authored individually, and the documents themselves must be held in a specific format. Additionally, only text may be used to locate relevant information; thus if the appropriate information is held in another media, the searcher relies on appropriate text being associated with it.

The Microcosm system (Fountain *et al.*, 1990; Hall, 1994) developed at the University of Southampton addresses the former of these problems by abstracting links from documents. To use the ubiquitous example, an author may decide to create a link from the word 'Southampton' to a document or video about Southampton. Rather than having to explicitly author the link on every occurrence of the word Southampton in a collection of documents, the link is created from the *word* Southampton; thus the link may be followed from any occurrence of the word Southampton in any document. The mechanism is known as the *generic link*.

Navigating using a generic link involves matching the content of the user's selection or query, in this case the word 'Southampton', to the source anchors of links in order to find a link or links that are relevant. This form of navigation is referred to as *content-based navigation*, or CBN.

This mechanism greatly reduces the authoring effort required to construct a useful network of hypermedia links. Nevertheless, in Microcosm, generic links must be authored from text. Links from other media can be authored but only in specific point-to-point ways, in a similar manner to the Web. A later system extended

Microcosm to enable generic links to be authored on multimedia data. This system is called MAVIS, standing for the Multimedia Architecture for Video, Image and Sound.

MAVIS allows links to be authored based on the content of multimedia objects (Lewis *et al.*, 1996a; Lewis *et al.*, 1997b). This is not a simple task. In Microcosm, just the word 'Southampton' needs to be stored, and can be easily compared with text source anchors, whereas in MAVIS an image selection cannot be compared with image link source anchors as easily. With text it is easy to detect when a link source anchor and a piece of selected text match, but with other media, matching is a fuzzy notion.

To address this, MAVIS extracts media-specific *signatures* from the media objects. These are low-level features calculated and compared by *signature modules*. Signature modules may be added and removed as appropriate, so any media type supported by at least one signature module may be used to author generic links.

MAVIS still is not a complete solution. A link authored on the word 'Southampton' can be picked up and followed from any instance of the word 'Southampton'; a link authored on the image of a car might not be picked up and followed from another image of a car, which may depict a significantly different view of a car. Additionally, why should links authored on an image of a car not be offered from the word 'car'? They both concern the same high-level subject or *concept*. This is a similar problem to the query specification problem in information retrieval: Exactly what is it that is the source anchor of the link or subject of the query?

An additional shortcoming of the approach is that the links are often too specific or too general for the user's needs. For example, a document about a specific region in Southampton, say 'Highfield', may well be useful for a user who has encountered the word 'Southampton' in a document. There may be a link from the word 'Southampton' to a document about Southampton, and a link from the word 'Highfield' to a document about Highfield, but the semantic relationship between 'Highfield' and 'Southampton' is not expressed in the system. Thus, the user may not easily be able to access the document about Highfield. Some mechanism for expressing such a relationship could therefore be a very useful navigational aid.

The problems that are to be addressed by this work are summarised below:

1. In order to be effectively retrieved and navigated, multimedia data often has to have text metadata associated with it. This can often be a prohibitively

lengthy and laborious task.

2. Media processing techniques are often not sufficiently flexible to recognise two different views of the same object, or category of object, if the two views are visually different.

3. Even using the generic hypermedia link facilities of Microcosm and MAVIS, a link authored on a particular view in a particular medium (as all links must necessarily be authored) will not be accessible to a user who has access to a different view of the object, or a view or representation of the object in a different medium.

4. In most multimedia and hypermedia systems, there are no mechanisms for expressing semantic relationships between objects. Those relationships may be extremely useful for the user.

1.3 The Proposed Solution

In broad terms, the problem is one of *understanding*; the computer does not understand what the source anchor or query means. So how can a computer be made to have some practical degree of understanding?

To answer this question one must return to basic principles. How does a human understand media? Smoliar *et al.* identify two components of media objects (Smoliar *et al.*, 1996): The media object itself is *representing* a real-world object or concept. This representation is said to lie on the *plane of expression*. Correspondingly, the real-world object or concept being represented is said to lie on the *plane of content*.

This distinction between that being represented, and that doing the representing, is a central idea to this work.

The proposed solution involves explicitly representing the *plane of content* in the computer; existing systems largely only hold the *plane of expression*. This plane of content is considered to lie 'above' the plane of expression, and in this work is referred to as the "semantic layer".

Media objects are connected to corresponding concepts in the semantic layer, and this increases the accuracy and flexibility of the available information. The more media objects connected with an individual concept, the more comprehensively represented it is, and the more likely that additional representations of the same concept can be correctly identified using low-level media features or other means.

Additionally, relationships between these concepts can specify knowledge about those concepts, and this too can be used to augment the information and methods of access available to a user. This semantic layer does not need to intefere with the media objects at all; thus it can coexist with other hypermedia and information retrieval systems.

This thesis describes this idea in detail, and how it can be achieved in practise. The various techniques that can be employed to exploit the semantic layer are described.

A system with a semantic layer has been implemented. This system is the successor to the first MAVIS project, MAVIS 2. It is an experimental system allowing generic hypermedia links to be authored in any media, and allows multimedia information retrieval. The *signature module* idea has been inherited from the first MAVIS project, since its virtues still hold; medium and problem domain specific techniques can be used to retrieve and match media objects, and they can be updated and supplemented as required.

Additionally, MAVIS 2 can run in a heterogenous distributed environment. MAVIS 2 processes can run on different platforms, and use internet protocols to communicate; thus MAVIS 2 processes can interoperate on a global scale.

MAVIS 2 allows the authoring of specific and generic hypermedia links between any media objects. It holds link information separately from the media objects themselves, so no restrictions are imposed by storage location or format.

As with MAVIS 1, signature modules are used to compare and retrieve media objects based on low-level features. These are used both for *content-based navigation* (generic links) and for *content-based retrieval*. By mapping these low-level features to high-level semantic concepts, judgements can be made about to what *real-world concept* a query or selected link anchor pertains.

Once this is achieved, the following become possible:

- Asking 'what is this?'
- Viewing synonymous media representations
- Expanding of content-based retrieval and navigation queries with media representations in other media
- Indexing media objects (documents) on the concepts themselves
- Supplementing available links with those from synonymous representations
- Navigating around the concepts themselves

- Narrowing the scope of queries by specifying a subset of concepts.

In MAVIS 2, the implementation of the semantic layer and media associations is called the *Multimedia Thesaurus*, or MMT for short. It is so called because its structure is based on classic thesaurus structure, with an hierarchy at its core. This allows existing thesauri and other hierarchical classification systems to be directly constructed in MAVIS 2.

The MAVIS 2 system has been used to hold and allow access to a collection of museum artefacts from the Victoria and Albert Museum. The artefacts all have pictorial representations, and additionally, some of the artefacts have associated textual descriptions.

This application is used in the thesis as a basis for evaluating the semantic layer technique. Due to the lack of applicable hypermedia evaluation metrics, the techniques have been evaluated heuristically; this is explained and justified.

1.4 Thesis Structure

Chapter 2 describes existing systems and techniques for retrieving and navigating multimedia information. These systems and techniques do not explicitly deal with the *meaning* of the information they hold.

Chapter 3 describes those systems and techniques that do attempt to explicitly capture the semantics of multimedia information.

Chapter 4 introduces the semantic layer technique, and describes the enhanced possibilities for navigational and retrieval of multimedia information it provides.

The MAVIS 2 experimental multimedia information system, and the implementation of the semantic layer technique, the multimedia thesaurus, are described in detail in Chapter 5.

Chapter 6 introduces the Victoria and Albert Museum collection, and the associated textual data created as part of the ELISE I virtual museum project. How this data was used to construct a multimedia thesaurus is explained. The chapter concludes with some reflection on how the techniques used demonstrate the viability of the semantic layer technique.

The usefulness of the multimedia thesaurus is evaluated in chapter 7. This takes the form of a number of scenarios of use of the multimedia thesaurus, since time constraints did not allow user trials.

Finally, chapter 8 reflects on the work in the thesis, and discusses how this work could be taken further.

1.5 Declaration

With the exception of work concerning the MAVIS 2 system in chapter 4, this thesis represents entirely my own work. The MAVIS 2 system was developed as part of a collaborative effort to produce a distributed open multimedia information system. The aspects of the system which represent my contribution are:

- The four-layer data model
- The Multimedia Thesaurus
- The Selection Expression Matcher
- The Results Viewer and associated result sorting algorithm
- The 'Control Room' process control tool
- The integration of the QBIC signature module (with considerable assistance from Colin Bird)
- The mechanism for 'qualifying' selections (i.e. specifying and storing which signature types are appropriate for a selection)
- Hypermedia linking functionality.

This work has been carried out as part of the MAVIS 2 project, funded by EPSRC grant number GR/L03446.

Chapter 2

Information Systems

2.1 Introduction

Of the large number of information storage and access systems available, the majority follow one of two paradigms.

The first is an older paradigm in terms of computer systems, and systems that follow it are known as *information retrieval* systems. These assume that one has a large set of documents or objects in a store, and wishes to access one or more documents in that store effectively by asking questions. This asking process is known as *querying*.

The basic principles of the second paradigm were conceived centuries ago. In early versions of the bible, passages of text referred to other, related passages. This idea has evolved in the computing age into the *hypermedia* system. The hypermedia paradigm is based around the idea of connecting related pieces of information. A searcher *explores* information by *navigating* around the information.

Only recently has it become practical to incorporate non-text information to either type of system to any significant extent.

These paradigms are summarised in the following sections.

2.2 Information Retrieval

The fundamental theory behind information retrieval is this: We have a large body of information available, and an individual who requires some information within this body. The individual's information need is most likely a very small subset of the entire collection. It is the identification and extraction of this subset of information with which the information retrieval discipline is primarily concerned.

van Rijsbergen describes this problem well in his seminal book on information retrieval (van Rijsbergen, 1979):

> "Suppose there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by his question. He can obtain the set by reading all the documents in the store, retaining the relevant documents and discarding all the others. In a sense, this constitutes 'perfect' retrieval. This solution is obviously impracticable."

This passage introduces the idea of *relevance*. Relevance is a subjective notion, since only the searcher can give the authoritative answer to the question "is this object relevant to the searcher's information need?" For a computer to give some prediction of this answer, it is necessary to construct a model that allows us to quantitatively estimate this judgement. The basic aim of the information retrieval system is to discriminate between those documents that are relevant to the user's information need, and those that are not.

The following section outlines three broad categories of retrieval model. Sections 2.2.2 and 2.2.3 give some details of existing techniques used and under research in the text and multimedia information retrieval domains respectively.

2.2.1 Retrieval Models

Since their inception, a variety of models for retrieval systems have been developed. Given an information need (specified in the form of a query) the classic retrieval system somehow determines the likelihood that a document (or, more generally, retrieved object) fulfills that need. Such retrieval systems can broadly be categorised into one of the following three models (Salton & McGill, 1983):

Boolean. If all terms (or atomic objects) in a document collection are represented as a set $\{r_1, r_2, \ldots, r_n\}$, it is possible to specify any document in the system as a vector of length n where the ith element is *true* if term r_i is present in the document. Queries can be represented as a set of these terms, and any document for which these terms are *true* can be returned.

It is quite simple using this model to express relatively complex queries, since the searcher (or application) may apply Boolean logic to the individual query terms. For example, a query may insist that retrieved documents contain term

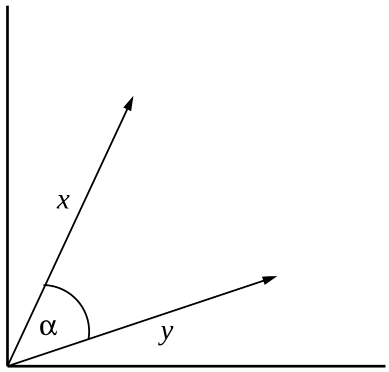


Figure 2.1: Dot Product Similarity Measure

r_j AND r_k, but NOT term r_m. It is possible to use this mechanism to formulate very specific queries; whether these queries are semantically meaningful is debatable.

This is the most common type of model, especially in commercial text retrieval systems (Ortega *et al.*, 1997) such as the Yahoo! web search engine (Yahoo! Inc., 1994). Extensions to this model exist, such as the application of fuzzy set theory (Salton, 1989).

While this method has proved useful for text information retrieval, due to the discrete nature of the text media (it is trivial to determine whether or not a term is present in a document or query), this technique is difficult to apply to the multimedia domain without some modification.

Vector Space. Let $\{r_1, r_2, \ldots, r_n\}$ once again be the set of terms present in a document collection. Documents and queries are both represented as vectors of dimension n, but this time the ith element is a weighting value corresponding to term r_i. Some measure is then used to estimate the similarity between documents and queries. One common method is to use the dot product

$$x \cdot y = |x||y| \cos \alpha \tag{2.1}$$

where $|x|$ is the length of the vector x (representing some document or query), and α is the angle between the two vectors. This is illustrated in figure 2.1.

Another method is to treat each document or query vector as a point in an n dimensional space. The similarity between two documents or a query and a document is estimated by taking the Euclidean distance between each point, as shown in figure 2.2. This method is frequently used by classification and image processing systems (Sonka *et al.*, 1993).

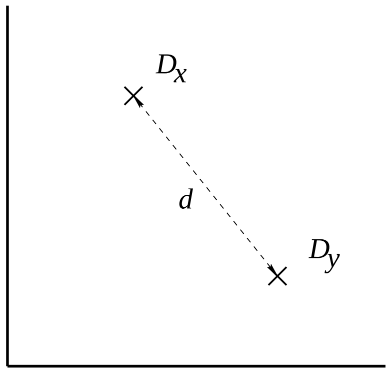


Figure 2.2: Vector Space Distance Similarity Measure

Probabilistic retrieval methods (van Rijsbergen, 1979; Sparck Jones *et al.*, 1998) are still largely of an experimental nature. What follows is a cursory overview; a more detailed description of the theory is outside the scope of this thesis.

As in the Boolean model, each document is represented as a set of Boolean values $x = \{r_1, r_2, \ldots, r_n\}$ corresponding to terms in the document collection. The probability of relevance (w_0) or non-relevance (w_1) for a document is $P(w_i|x)$. A value for this is estimated using Bayes Theorem:

$$P(w_i|x) = \frac{P(x|w_i)P(w_i)}{P(x)} \quad i = 0, 1 \tag{2.2}$$

$P(w_i)$ is the initial probability of (non)-relevance, $P(x|w_i)$ is the probability of relevance given x, and $P(x)$ is the probability of observing x on a random basis. A *retrieval function* is applied to this equation for each document, and the probability for each x (document in the collection) calculated. This produces a ranked list.

The main problem with this approach is that the probability distributions $P(w_i)$ and $P(x)$ must be known or estimated *a priori*.

2.2.2 Text Information Retrieval

The field of text information retrieval is not a new one. People have been organising and facilitating access to collections of literature for centuries. Even early on the problems of storing and accessing large amounts of (textual and/or numeric) information were apparent. Research into using computers for this purpose began in the 1940's. Hence, a wealth of techniques have been developed for high performance retrieval of text, both in terms of quality of results and computational efficiency. The most significant developments and techniques are summarised in this section.

Indexing

While it may be possible for a computer to trawl sequentially through all the text documents in a collection of documents, this is hardly an optimal solution. A keyword index is a 'reverse index' of the document collection, which holds a list of the terms in a collection together with list of documents each term appears in. In this manner, documents containing a particular term (or terms) can be retrieved simply by looking up the term(s) in this list.

The words (and/or terms) included in the index constitute what is called the *index language*. The ideal index language for a particular collection of documents needs to be good at discriminating between documents relevant to the user's request and those that are not. Index languages have the following features:

Exhaustivity. The number of topics indexed.

Specificity. The ability of the index language to describe topics precisely.

In some cases, it is not useful or appropriate to include particular words or terms in an index. For example, certain words are likely to appear in virtually every document in the system, such as *the*, and are usually excluded from the index and are termed 'stop words' (van Rijsbergen, 1979).

Another problem with a straight extracted-term to document index is that different forms of the same word appear separately. For example, if a document is indexed with the term 'tabulation', it will not be picked up during a search for the word 'tabulate', even though intuitively the document is very likely to be relevant to that search. A technique commonly used to alleviate this is *stemming*.

Stemming is a technique whereby each word in the index is reduced to a stem. Both 'tabulation' and 'tabulate' would both be reduced to 'tabulat.' Search terms are similarly treated, and so a search for any form of the word 'tabulation' will find documents containing any (other) form of 'tabulation.' A number of algorithms exist for working out the stem of an arbitrary word, such as Porter's algorithm (Porter, 1980). Research is still ongoing in the area (Xu & Croft, 1998).

One final technique of interest used in keyword indexing is the introduction of a weighting factor, giving some measure of to what extent the keyword appears in (or is relevant to) a document. This can be used when ranking retrieval results.

Document Clustering

The basic premise of document clustering is that similar documents are likely to be relevant to the same queries. The idea is that by *clustering* similar documents together, one can treat the cluster as a single entity during search and retrieval, thus greatly reducing the computational demands of searching. This is known as the *cluster hypothesis* (van Rijsbergen, 1979).

Some disadvantages are identified by Crouch *et al.* (Crouch *et al.*, 1989):

- Some features such as query-document similarity functions are decided only by the system designer.
- Terms are assumed to be independent of one another.
- Relationships between terms are not expressed.

A detailed discussion of this technique is outside the scope of this review.

Relevance Feedback

Relevance feedback is a long-established technique in the text information retrieval world (Salton & Buckley, 1990; Walker & Vere, 1990). The principle is that by flagging retrieved documents as relevant or non-relevant, a better query can be formulated.

It is the process whereby once the initial results of a query are obtained, the user indicates to the system those retrieved objects that interest them, and those that do not. These are used to *expand* or *refine* the query and produce a new set of results that should better meet the user's infomation need.

Typically this relevance feedback is used when the initial results of a query fail to exectly match the searcher's information need. This is likely to occur in the majority of cases (van Rijsbergen, 1979; Salton & Buckley, 1990; Walker & Vere, 1990). Through some mechanism, the user adds or removes terms from the query, the query is re-processed, and the results again presented.

The user can be given varying levels of control over this, the extremes of which are:

Opaque. The user can tag retrieved documents as good (relevant) results or bad (irrelevant) results. From these decisions the system determines which terms to add/remove from the query.

Transparent. When tagging retrieved documents, the system will suggest to the user terms to add/remove from the query. The user can select which to actually add/remove and may add their own words too.

Work by Koenemann *et al.* suggests that even for inexperienced users, the greater the user control over the feedback the more effective the feedback is (Koenemann & Belkin, 1996).

Relevance feedback is usually used in systems where a text query is used to retrieve a ranked set of documents, though the technique has been applied with some success to a system using Boolean operators (Smith & Pollit, 1995).

2.2.3 Multimedia Information Retrieval

Multimedia information retrieval (*MMIR*) shares the same aims as text retrieval. Given a query representing an information need, an MMIR system will attempt to find multimedia objects relevant to that information need. While numerous ways of performing this task have been conceptualised and implemented, the area is still very much in its infancy, and most of the associated problems remain unsolved.

The principle problem with MMIR is that a computer cannot "understand" multimedia information such as images. It has been noted that most information in real-world data is ambiguity, uncertainty and noise (Oka, 1998), thus only a small amount of the actual information available is relevant. A computer lacks the ability of a human to *perceive* this information, and to *recognise* and *comprehend* an object or sound. The same is true of text, to an extent; text, however, is encoded 'perfectly' with no loss of information. A letter is the finest level of granularity of the text medium, and words are delineated by spaces and punctuation, and both of these are represented perfectly inside the computer. Thus, it is a discrete medium that is easily indexed and compared in a computer's memory automatically. This is due in part to the fact that text is a symbolic medium, and is hence already an abstraction from the real world.

There is only a finite (though possibly very large) number of ways that a real world object or concept can be represented using text. A large number of these ways can be determined by a set of syntactic rules and transformations (Salton, 1989).

Certain media types (such as vector graphics in a small application domain) can also be analysed in this way; however this problem of computer *understanding* is far from solved in the general case.

Smoliar *et al.* identify a distinction between the real world object a media object represents, and the media object itself (Smoliar *et al.*, 1996). They describe these in terms of semiology, and the proposition that a symbolic object is composed of two parts. One lies in *plane of expression* (the object used to communicate the concept) and the other in the *plane of content* (the concept being expressed.)

Similarly, Gupta *et al.* describe the two parts as the *semantic value* of a visual object (what it depicts) and the *appearance value* (the raw features of the image) (Gupta *et al.*, 1997).

An ideal MMIR system, then, would be able to map between the planes. The *content* (concepts represented by) a query could then be used to evaluate that query, and the results presented using the plane of expression. This ideal system is still a long way off; the majority of systems consider only the plane of expression.

A compromise many systems use is to allow the assigning of keywords or descriptive text to images or other media objects. Text is already a symbolic medium; it is already an abstraction of concepts from the real world, and so is a step closer to a conceptual representation of the object anyway. The objects can be retrieved using traditional retrieval techniques on the associated text. However, this requires a huge and often prohibitive amount of authoring effort, especially in large data sets (Ortega *et al.*, 1997). A solution is required where purely the *content* of the media objects is used during the indexing and retrieval process.

A description of some of the methods that current MMIR systems employ is given in the following sections. A number of existing systems are then explored, with a description of the key features of each.

The Matching Problem

One of the principle problems encountered in multimedia information retrieval and navigation systems is the *multimedia matching problem.* 'Matching' refers to the process of determining how similar two pieces of media are, with a view to determining whether or not the two media objects represent the same real world object. In the case of video, image and sound, two media objects representing the same real world object will not be exactly the same, even if the difference is insignificant to the human eye or ear. This presents a number of problems.

(Dis)similarity Measures

There are many ways in which two pieces of media can be considered similar. In the case of images, they may be similar in colour or shape for example. In the case of video, they may depict objects moving in similar manners. Two sounds may be of a similar pitch. Each such characteristic of a media type is known as a *property* or *feature* of that media type.

Not only can each media type have a number of features, but different methods exist for measuring and comparing these features. For example, there are many ways of comparing the colours of two images, such as RGB histograms (Swain & Ballard, 1991) or spatial colour distribution (Stricker & Dimai, 1996; Vellaikal & Kuo, 1995).

Content-based retrieval systems attempt to address this problem by using algorithms to estimate the *similarity* of two pieces of media (Jain *et al.*, 1995; Smoliar & Zhang, 1994; Amato *et al.*, 1996; Foote, 1997). Typically, given two media objects (of appropriate type), an algorithm can give some mathematical indication of how similar the two objects are in terms of a property. Quite often, this will be a measure of *dis*similarity rather than similarity, since the commonly used vector space retrieval model lends itself to the measuring of distances between objects in the feature space.

Retrieval systems usually require the user to provide an example media object and specify a property or properties of the example. The system then uses the results of applying the algorithm to this example to find similar objects from an index. This produces a ranked (according to the results of the algorithms, and any weightings provided by the user) list of objects in the system. However since the result is fuzzy and not Boolean, it is likely that *every* object of the same media type as the query in the system matches the query to some extent. Thus the list is usually thresholded, either by only considering the top n matches, or selecting a cut-off value for the (dis)similarity measure.

Some other techniques allow the searcher to specify some abstract property to search multimedia objects for directly, for example an RGB histogram. Other techniques allow the specification of a query 'from scratch'. For instance, a number of researchers have worked on techniques for the retrieval of music by humming a query (Ghias *et al.*, 1995).

There is a wealth of research in the field of media matching technology (Caetano & Guimarães, 1998; Bird & Elliot, 1998; Amato *et al.*, 1996; Jain *et al.*, 1995; Smoliar & Zhang, 1994; Idris & Panchanathan, 1997). Most is application specific, and some is geared towards finding quicker ways of finding a similar object from a large database of candidates. Allowing the application of these in a modular fashion will greatly enhance the flexibility of any multimedia information system.

Different Views of an Object

Even though two pieces of media may represent the same object, the media may be significantly different. For example, two images might depict a single object viewed from different camera angles, or two sounds might be of different voices saying the same thing.

A handful of algorithms attempt to address this problem. A three dimensional model of an object can be used to identify an object at any orientation. This method is however a fledgling one, only works for relatively simple objects, and is computationally expensive (Terzopoulos & Metaxas, 1991; Sonka *et al.*, 1993; Trucco & Verri, 1998).

Existing techniques rely on the production of some representation of the real world object without media examples, such as a three dimensional model (Chen & Stockman, 1996). An alternative approach is to store some metadata indicating which media objects represent the same real world object. A good match with a media object then indicates that any of the associated media objects may also be relevant to the query (Smoliar *et al.*, 1996).

Recognising Objects in a Larger Context

Most content-based retrieval systems will only retrieve a media object that matches the query well (using whatever similarity measures are appropriate) in its entirety. Consider a query consisting of an image of a car; the user may wish to retrieve scenes or videos *containing* cars, but a scene or video as a whole might not produce a strong match with the query object. A scene will not be retrieved unless any cars depicted in it is entered into the system as a media object in its own right.

There are methods for locating objects in scenes such as *template matching* (Sonka *et al.*, 1993). These methods are usually computationally expensive. Currently, some (possibly minimal) form of user interaction is usually required when selecting objects in a scene.

2.2.4 Existing Multimedia Retrieval Systems

Several retrieval systems already exist allowing the retrieval of non-text media. Most allow the assigning of keywords to images too, however this often requires excessive authoring effort in large data sets (Ortega *et al.*, 1997).

QBIC - Query By Image Content

IBM's *QBIC* system, as the title suggests, allows images to be retrieved using their content alone (Faloutsos *et al.*, 1994). The assignation and searching of keywords can also be used to augment the system.

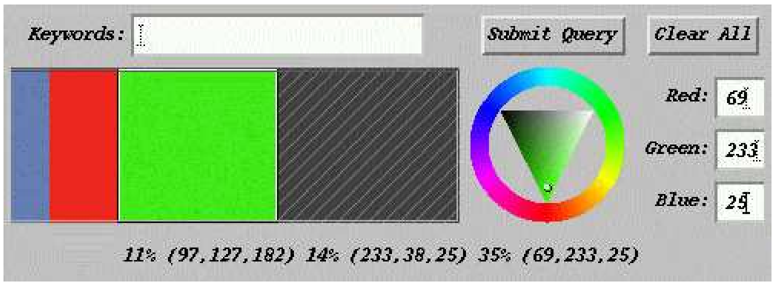


Figure 2.3: The QBIC Colour Percentage Query Interface

Images must be indexed by the system prior to querying. Images are imported into the database, and some feature vectors calculated. The feature vectors calculated by QBIC are:

- The average RGB colour over entire images.
- A 256-colour histogram. Similarity is calculated based on the distribution of image colours.
- The dominant colour in predetermined regions of each image. Similarity is calculated based on the similarity of each region, so the position of an area of colour in an image is a factor.
- Texture attributes such as directionality, coarseness, and contrast.
- A combination of heuristic shape features: Area, circularity, eccentricity, major axis orientation and algebraic moment invariants (Niblack *et al.*, 1993).

Queries can take three forms:

Query by example. A query image can be submitted to the system, and the top n images similar to the query image based on one of the above features can be returned, together with a distance between each of those images and the query image. The system makes no attempt to understand the semantic meaning of images, other than through use of the keywords.

Query by colour percentage. Queries can be formulated by specifying a colour histogram using the interface shown in figure 2.3. With this interface the searcher dictates the rough proportions of colours that the images should contain.

This form of query is rather abstract and concerned wholly with the plane of expression. It is very hard for a human to map something as abstract as a colour histogram to a real-world concept, and so the usefulness of this type of query is limited. This technique is also rather harder to integrate with other systems than the straightforward "query by example" mechanism.

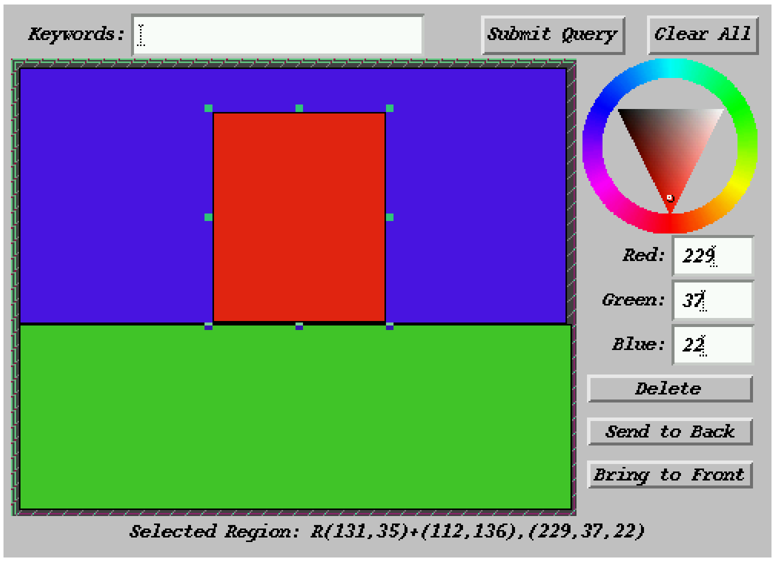


Figure 2.4: The QBIC Colour Layout Query Interface

Query by colour layout. The query if formed by drawing and moving regions of colour around a 'scratchpad' (Flickner *et al.*, 1995). This is shown in figure 2.4. This form of query is considerably more useful than the colour percentage form, and in some circumstances may even be more useful than query by example, since the searcher does not require an example image from which to start their search.

Query by sketching Later versions of QBIC allow the user to draw a freehand sketch. The shape of this sketch is extracted, and used to search for images with a similar shape feature.

QBIC at present can work only with entire images; it is not possible to retrieve or specify as a query a sub-part of an image. However, it is available for a wide range of platforms and is relatively easy to use from within another application.

QBIC has also been adapted to work with video images (Niblack *et al.*, 1998).

CHROMA

The CHROMA image retrieval and navigation system has been developed at the University of Sunderland by Lai *et al.* (Lai *et al.*, 1999). The system allows both query-by-sketching and query-by-example methods of image retrieval. It also allows the browsing and navigation of images, by means of an hierarchical classification of the images.

Images are initially classified automatically based directly on their image features, specifically colour. A set of colour categories has been devised, dividing the HSV colour space into ten. Each division has been given a digit between '0' and

‘9’. These are used to give each image a class number that indicates the dominant colours in that image. For example, if an image’s class number is ‘87’, then the most dominant colour in the image is blue (represented by the ‘8’), and the next most dominant colour is green (represented by the ‘7’). The class number can have as many numbers as there are significant colour regions in the image.

This class number mechanism provides a simple way of allowing access to clusters of similar images. For example, specifying ‘87*’ encompasses all of the images which have blue as the most dominant colour, and green as the next most dominant. Using this mechanism, hierarchies of images can be specified. For example a class ‘87*’ is a specialisation of the class ‘8*’.

CHROMA provides a tool for browsing this hierarchy and using it to refine the scopes of queries. The technique is tied into a particular visual similarity measure, colour. Additionally, and does not take into account the *semantics* of the images; images are only grouped together by a specific *visual* characteristic. However, the system demonstrates the potential of organising images into a hierarchy for improving the usefulness of an image storage and access system, and the feasability of a hybrid retrieval and navigation approach.

WebSEEk

VisualSEEk and WebSEEk, a more open web-based version (Smith, 1996b) are both content-based image (and video) retrieval systems. They both work on images alone, but images can be indexed and used as queries from anywhere on the web (Smith & Chang, 1996).

Later work by Smith and Chang resulted in the *SaFe* (Spatial and Feature Query) system (Smith, 1996a). This allows the combination of spatial and other image features. When images are introduced into the system, regions with prominent features are extracted (Smith & Chang, 1999). The spatial relationship between these regions is also extracted and indexed.

The WebSEEk and SaFe systems use colour as the feature with which to identify and match individual regions in images. They use a derivative of colour histograms, ‘colour sets’. These are a kind of cut-down histogram that rely on that fact that a region will consist of only a small number of prominent colours. Since this is applied to regions rather than whole images, this holds reasonably consistently.

WebSEEk also allows images to be categorised in an extendible subject category taxonomy, and this information can be used during a query (Smith, 1997).

A searcher can restrict their search to images in a particular category, producing a useful rise in retrieval precision. The category taxonomy is a strict hierarchy, specified in the form `broad-category/sub-category/sub-sub-category`. A category subset can be simply specified with the broadest categories, for example `broad-category`.

This categorisation can be done semi-automatically using text-only analysis (such as analysing HTML tags, and the web page that the image appears in) with some degree of success. This does rely on their being suitable surrounding HTML text with images, and the content of the image itself is not used during the categorisation. In cases where a category cannot be found automatically, a category must be assigned manually.

Another interesting feature of the WebSEEk system is the automated collection of images and video. Web 'spiders' automatically follow web links, collecting images and video clips. These images are categorised (using any associated text), processed and indexed. This enables a large dataset to be built up with very little operator effort.

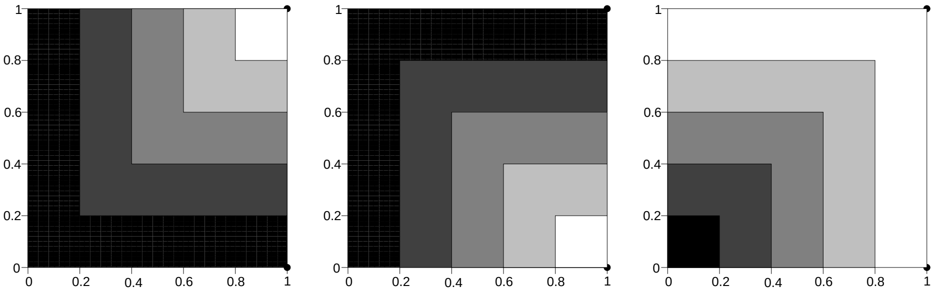
In summary, WebSEEk is a content-based image retrieval system, though no attempt is made to map the semantics of images (the plane of content) to image content itself (the plane of expression.) The emphasis is still on the location of images similar to a particular example image. However, the image matching techniques used are quite powerful, and the novel method by which images and videos are automatically gathered and categorised is of special note. Also useful is the way in which the scope of a query can be specialised by specifying a sub-category.

MARS

The Multimedia Analysis and Retrieval System (*MARS*) is an image retrieval system that uses two extensions to standard Boolean retrieval, called *fuzzy Boolean retrieval* and *probabilistic Boolean retrieval* (Ortega *et al.*, 1997). These provide a rank order for retrieved objects.

MARS uses the following image features for retrieval:

- Colour HSV Space
- Texture
- Shape
- Layout



a) *and* operator b) *and not* operator c) *or* operator

Figure 2.5: MARS Boolean Operators. The lighter areas indicate a stronger overall match.

The layout technique works by splitting an image into a 5×5 grid and each feature calculated for each area on the grid.

One novel aspect of MARS is the ability to retrieve images based on more than one feature in one operation. To this end, the vectors representing features are *normalised*, so that equal emphasis is put on each. The searcher can alter the weights put on each feature, so for a composite query, the searcher can assign 90% of the weight to the colour histogram feature and 10% to the texture feature. Whether this weighting has to be assigned during the query formulation, or can be applied dynamically to results is unclear.

A query can specify a number of different features to search for using Boolean operators (Ortega *et al.*, 1998). The contour graphs shown in figure 2.5 show how these operators are used to give two features an overall similarity measure in the *fuzzy Boolean model*. Each axis of the graph represents the similarity value calculated for one feature. The lighter the point of intersection, the stronger the overall value of the match. During retrieval operations, the number of retrieved images can be limited by defining a bounding box in one corner or edge of the graph. This is referred to as the *Observed Area Bounding Box* (OABB).

Similar graphs apply to the probabilistic model, but the contours are a more curved shape. Specific details of this are beyond the scope of this review.

Evaluation of the MARS system concluded the following key points:

- Varying the weighting of the different features can dramatically improve performance for simple queries. This is probably because different features may be different to different information needs. For example, if one is searching an archaelogocial image database for a particular type of fabric, the texture feature is likely to be more relevant than the shape feature. If one is searching for a particular type of axe, then the shape metric is likely to be the feature that best discriminates images that are relevant from images that are not.

In the MARS experiments, the weightings of different features were changed subjectively by the user. Work is also in progress to determine ways in which this can be performed automatically (Rui *et al.*, 1998).

- Relevance feedback is a powerful mechanism for capturing the subjective perceptions of the human user, and using them to improve retrieval performance (precision and recall). (Rui *et al.*, 1998; Porkaew *et al.*, 1999)
- Varying the weights of features in complex queries (queries involving multiple terms, for example $shape(I_1)$ AND $colour(I_2)$ OR $shape(I_3)$, did *not* significantly increase retrieval performance.

Informedia

Informedia is essentially a video library system with indexing and retrieval capabilities (Christel *et al.*, 1994). Retrieval of video presents a variety of new problems. As well as the image analysis problems addressed by image retrieval systems, a video database system must also take into account the temporal nature of the medium, and in addition must handle audio information.

The life cycle of the Informedia system is divided into two parts: Creation, and exploration. The creation stage involves *segmenting* videos into small chunks, allowing faster and more efficient access to video content. It is considered likely that the whole of a video 'clip' is likely to be relevant to a query — a similar assumption to the cluster hypothesis (section 2.2.2).

Videos are segmented into three levels:

Paragraphs. A paragraph is defined in this context as a "series of of related scenes with a common content." How this is determined is unclear from the literature.

Scenes (or 'shots', or 'clips') are automatically-determined segments consisting of footage of a single location and camera angle (or camera 'pan'.)

Frame Icons are still images representative of individual scenes, generally used for displaying the results of a query to the user, and other purposes where the displaying of temporal media is impossible or impractical, such as on a printout.

In order to perform the segmentation, three core technologies are employed:

Text Processing. Any textual information available about a video sequence may be processed. This may be in the form of closed captions, or producer's notes. These may contain structural text markers such as punctuation which can be exploited to identify video 'paragraphs'.

Also the associated text may be searched for keywords, allowing video segments to be retrieved using text queries (Christel & Martin, 1998).

Speech Signal Analysis. Text information can be extracted using speech recognition techniques, though these may produce inaccurate data. A signal-to-noise ratio (SNR) technique can also identify breaks in utterances, providing further clues to the extents of video paragraphs.

Image Analysis techniques are used to identify scene changes and suitable *frame icons*. Large changes in colour histograms indicate probable scene changes, and optical flow techniques can identify camera pans and zooms.

In later research, face detection techniques are also used to discriminate scenes that depict people from those that do not (Smith & Kanade, 1998).

Ma's Texture Thesaurus

At the University of California, Ma *et al.* have applied a thesaurus view onto a database of extracted textures (Ma & Manjunath, 1998). The textures are extracted from regions from a large number of digitised aerial photographs depicting various geographically salient features.

The textures are first partitioned in the feature space into a set of classes determined by a human user. These are then iteratively partitioned further, until a predefined number of partitions have been created. Each partition is given an automatically generated 'codeword', which is basically an identifier for the partition. It is not human-readable.

When further images are introduced into the database, they are automatically assigned the codeword corresponding to the best-matching partition. When an image is issued as a query, the best-matching partition is found, and all images with the corresponding codeword can be retrieved. A simple Euclidean distance measure is then used to rank the results.

The above process is classic clustered retrieval. The novel aspect of this work lies in the application of the thesaurus paradigm to the clustered image regions. The user can browse around the thesaurus hierarchy and 'beam down' to the original photographs as they wish. However, the 'terms' in the thesaurus have no meaning other than that they are visually similar; they are not necessarily semantically similar.

EVA

EVA is a multimedia information system based on the database paradigm (Golshani & Dimitrova, 1994). The query language is an extension of the 'Varqa' database querying language. The EVA extensions allow for the manipulation of multimedia information by providing operators of the following classes:

- Operations for querying and updating multimedia information. This includes four types of retrieval:
  - retrieval by identifier (object ID)
  - retrieval by traditional database-style conditional statements
  - retrieval by example
  - retrieval by semantic content
- Operations for screen management (such as "display x to the left of y")
- Temporal operators (e.g. for audio/video synchronisation operations)
- Operators for specifying rules and constraints (e.g. $month \in (1 \ldots 12)$)
- Computational operators (e.g. result aggregation)

Of particular concern to this research is the 'retrieval by example' and 'retrieval by semantic content' operations. It is admitted in (Golshani & Dimitrova, 1994) that EVA's capabilities for these two operations is limited. Regions of images can be specified, and the colour values in that region calculated. Other images with similar regions can then be retrieved. This mechanism can be used to retrieve by semantic content in a rather primitive way. For example the query 'find images where a man is standing by a tree' can be approximated by extracting the colour values of a region of an image containing a man, and those of a region containing a tree. These can the be aggregated and similar images retrieved.

AMORE

Mukherjea *et al.* at the C & C Research Laboraties at NEC USA Inc. have implemented an *Advanced Multimedia-Oriented Retrieval Engine* called AMORE (Mukherjea & Hirata, 1997; C & C Research Laboratories, 1997; Mukherjea & Cho, 1999). In a similar way to WebSEEk, AMORE can automatically index images on Web sites and use the surrounding HTML document to assign keywords to the image. Further to this, AMORE studies the surrounding web pages and web site to obtain more keywords. AMORE can retrieve images visually similar to a query.

AMORE has the option for finding *semantically*-similar images to a query. This is achieved through use of the automatically-assigned keywords. Categories can also be assigned to images through use of keywords. No attempt is made to map visual image content to semantic meaning; 'semantics' are achieved entirely through keywords.

The AMORE system also implements some novel result visualisation techniques. These include the 'scatterplot' visualisation, that places thumbnails of retrieved images on the screen. Each axis (including a Z-axis) indicates one measure of similarity, so images that are more similar are closer together on the screen. Another visualisation is the 'perspective wall' visualisation. Small cubes whose size represents a (visual) match strength are placed on 'walls' in a VRML environment. Each wall contains cubes representing images with similar keywords.

Other Systems

The following MMIR systems are also worth mentioning.

Virage Inc.'s Visual Information Retrieval Tool is an extension (called a *datablade*) for Informix Software Inc.'s DBMS that allows the indexing of images based on image features such as colour and texture. (Bach *et al.*, 1996). It also has been adapted for use in video.

RetrievalWare. Excalibur Technologies have produced a commercial set of image information retrieval tools, implementing a number of matching and indexing techniques (Feder, 1996).

Photobook was an early image retrieval system. A novel feature of Photobook was in its approach to storing information about images. Rather than store extracted features such as shape or texture, it stores information about how to extract those features (Pentland *et al.*, 1996). This allows new matching and indexing methods to be added transparently and at query time. There is a corresponding speed deficit because of this.

Chabot. The Chabot system is an early MMIR that allowed a collection of digitised images to be searching on the basis of text and colour features (Ogle & Stonebraker, 1995). This system gave rise to both the Cypress system

(Berkeley Digital Library Project, 1996), and Blobworld, a colour-based region extraction and matching system that has demonstrated some success at extracting objects from scenes (Carson *et al.*, 1999).

Sketch-drawing-tool. Nishiyama *et al.* have produced an image retrieval interface that allows users to place icons on a query 'sketch' in order to retrieve images (Nishiyama *et al.*, 1994).

The system allows users to formulate queries by using symbolic representations of real-world objects, rather than requiring an existing example image. However, each image in the system must be symbolically annotated in a similar fashion by a human before it can be retrieved by the system, and so the approach is not scalable to large sets of images.

Image Retrieval with Latent Semantic Indexing. Pečenović has applied the technique of latent semantic indexing (Pečenović, 1997) (closely related to latent semantic analysis, described in section 3.5.5). This involves the mathematical technique of reducing the dimensionality of a matrix of features. While the system produces good retrieval performance, it does not actually use the *semantics* of the images as such, it merely adapts the mathematical model used in latent semantic indexing to image features.

2.2.5 Evaluation

Formal evaluation of information retrieval systems has been an integral part of information retrieval research, and thus a large number of techniques have been developed. A good summary of many can be found in chapter 7 of (van Rijsbergen, 1979).

The most common of these use *precision* and *recall*. These are defined thus:

$$PRECISION = \frac{|A \cap B|}{|B|} \tag{2.3}$$

$$RECALL = \frac{|A \cap B|}{|A|} \tag{2.4}$$

where A is the set of all *relevant* documents in a collection, and B is the set of *retrieved* documents resulting from a query.

A variety of methods exist for combining these into a single measure. It is also common to plot these as precision-recall graphs, plotting the precision of a retrieval operation against recall. Recall is altered by varying the number of documents retrieved.

One requirement of this (and other) evaluation techniques is that of a 'ground truth'; that is, for a test set of documents, the 'correct' documents that should be retrieved given a certain query are known beforehand. For text retrieval, there are many large 'benchmark' document collections for which these ground truths are specified. The most well known of these are known as the TREC collections, after the Text REtrival Conferences (Smeaton & Harman, 1997).

Unfortunately, in the multimedia domain, no such collections exist (Smith & Li, 1998). This greatly complicates the evaluation of multimedia systems; it is up to individual researchers to provide their own collections and methodologies.

2.3 Hypermedia

Information retrieval systems tend to enforce a particular style of use or 'dialogue'. Information is held in a large 'data store'. The user specifies a query by some means, and the system returns a set of documents or media items that are deemed relevant to that query. A feedback or refinement process may be involved, but the form of the dialogue is the same.

An alternative method of 'information discovery' is to explore or *browse* documents (or media objects), *navigating* to other documents/objects by following *links*. This idea in essence is not new. One could argue that the idea of linking documents started as early as 3000 b.c.; papyrus scrolls and written clay tablets have been uncovered that refer to earlier ones.

A far closer ancestor was devised in this century. Vannevar Bush conceptualised a system that would enable this method of information exploration before the age of digital computers, a system based around mechanical devices such as microfilm. This work was published by Bush in 1945 (Bush, 1945), and the system was called MEMEX. The basic idea is of a unified store of books and other documents, and of a person's ability to make associations between pieces of information, and to *annotate and add new material* at will. The reason for the emphasis on this last phrase will become clear in the next sections; in short, aspects of Bush's vision have to some extent been lost (Nürnberg *et al.*, 1998).

The term used to describe this idea of associating related pieces of information these days is still over three decades old. Coined by Ted Nelson in 1965, the idea is called *hypertext* (Nelson, 1965). Of course, in recent years computers are capable of handling more than mere text — the linked information may include imagary, video and audio clips, and maybe even more types of information. This genericised version has been given the name *hypermedia*.

When looking over the history of hypertext, it is possible to see how the initial ideas were far ahead of what technology was able to provide at the time (Nelson, 1981). As technology caught up, people's thinking underwent a somewhat reverse trend, having to come up with ideas that were possible to realise given the available technology. Now technology is catching up, and people's ideas are once again heading forwards.

The following section described those ideas and systems that were implemented while technology lagged behind the ideals of hypertext. These systems are often called 'closed' hypermedia systems. The second category, 'open' hypermedia systems, are arguably closer to the original ideas of Bush and Nelson — in many cases (unlike in 'closed' systems) any user is free to make their own associations and add their own material at will. The reader will notice that the World-Wide Web, the most common hypermedia system in widespread use today, falls between these categories.

2.3.1 Closed Hypermedia Systems

Early hypertext/hypermedia systems tended to be proprietary, stand-alone systems that allowed the entry of text in a particular format. This text could be linked by defining a *source anchor* within a document, and a destination anchor in another. When a user views these texts, the links would be highlighted, and (usually) clicking on a highlighted link would display the text containing the destination anchor.

A searcher could then, from some starting point, navigate around the 'hyper-document' by following these links, until their information need (or curiosity) was satisfied.

These systems have been termed *closed* or *legacy* hypermedia systems. They impose a number of restrictions on authors and users alike:

- Documents must be authored in, or converted to, specific proprietary formats. Thus, they are difficult to edit or update once they have been imported to the system.
- Links are embedded in documents (a method usually referred to as *embedded markup.*) This means that when someone authors a link, they must edit the document itself, and all other users will see the link. This necessitated access

control to the documents, to stop unauthorised people from editing document content and links.

- Each and every link must be authored separately. Even though many links may be conceptually identical (for example, a link from an electronic component code number to a description of that component), each occurrence of that code must be edited individually and linked to the description.

These restrictions obviously annul many advantages of the ideas put forward by Bush and Nelson. Users cannot edit or annotate information at will, and cannot define their own links and trails through the information. It was clear that new approaches were needed (van Dam, 1988).

2.3.2 The World-Wide Web

The World-Wide Web (or *WWW*) is a hypertext system of special note. It is far and away the most widely used hypertext system in existence. It is used by millions of people each day. Few people in the developed world have not heard of 'the Web,' and in fact, the term has become synonymous with the internet, the global computer communications mechanism over which it operates.

It is becoming increasingly pervasive in society. Many consumer items, for example soft drinks cans, depict a *Uniform Resource Locator* (URL), the 'address' of a set of hyperlinked documents that contains information about that product, the company that manufactured that product, and other products manufactured by the same company. Over the last five years, the majority of organisations have developed a presence on World Wide Web, or *Web site*. Large companies have them. Small companies have them. Universities and libraries have them. Hospitals have them. Governments and countries have them. Cities have them. Schools and colleges have them. Churches have them. Also, with perhaps the biggest social impact, countless millions of individuals have them; their own, globally-visible, totally owner-controlled voice. It is the first time in history that any individual in developed nations can, with the aid of a PC and a modem, project a view of themselves that the entire world can see.

This fulfills a part of the Bush and Nelson vision; it is global, and everyone can have a voice. People anywhere can use this system to satisfy an information need, whether it is product information, bibliographic information, or tourist and travel information, or simply for entertainment.

This is all quite an achievement considering that the Web started life as a modest means for a small group of physicists at CERN in Switzerland to transport electronic documents (Bernes-Lee, 1996).

However, the Web does not fulfill the vision entirely. People cannot make their own annotations and associations to other people's or company's documents. Due to the point-to-point linking system, it is very difficult (even impossible) to link from an item of information to every other relevant piece of information there is. From a more pragmatic viewpoint, the problems can be summarised thus:

- Documents must be stored in HTML or 'Hypertext Markup Language' format. HTML lacks many features of other document and data formats, for example support for displaying mathematical formulae, a curious omission considering its originally intended use by physicists.
- Links are embedded in documents. This makes them difficult to maintain, and it is also impossible for a single anchor to lead to more than one destination.
- There is no system-supported concept of *link integrity.* The destinations of links are specified as URLs, which indicate the location of a document in a direct way, in much the same way as a filename indicates the location of a file on a filesystem. If the destination of a link is a location over which the author has no control, there is no guarantee that the destination will always be there, or that its content will always be appropriate for the link.

However, in many respects, the Web is also an 'open' system:

- Web servers (stores of documents) and web browsers (clients for viewing the documents and following links) may be heterogenously distributed over a variety of hardware and operating system platforms.
- While hyperlinked documents must be stored in HTML, other document formats (both text and other media) can also be downloaded using HTTP (HyperText Transfer Protocol.) The web browsers can use *plug-ins* to display (or play) these documents, or they can download the document and launch 'helper applications' in order to display or play them.

  These plug-ins do tend to be platform-specific, and require downloading and installation by the user before use.
- Additional flexibility can be gained through use of a standard called the Common Gateway Interface or CGI. This is a standard whereby a web server, when requested by a client (browser) for a certain page, executes a program

(or script) that generates HTML output. This output is then sent to the browser for displaying.

The most common use for this is for generating search results. The user fills out a search *form* (described below) and the CGI script performs the search and returns the result. The script can obviously embed links in the document. Interestingly, this allows the integration of an information retrieval element to the Web.

- HTML documents can include various input mechanisms, such as areas where text can be entered by a user or buttons clicked. This data can then be sent back to the server, where it is normally handled by a CGI script.
- A Web browser can supply many aids to navigation of links. The most common of these is the *history*, a chronological list of the documents the user has visited. This reduces the possibility of becoming 'lost in hyperspace'.

The World Wide Web, while not perfect, has introduced the world to the idea of hypermedia. Many people who use it are following links without even knowing what hypermedia means. Vast amounts of information are available, though due to the limitations described above, finding a specific piece of information is not necessarily easy.

Due to the enormous amounts of information on offer, it seems foolish not to make use of it somehow. Methods exist for automatic traversal and creation of HTML documents, and since the protocols and specifications are open, integration is fairly easy. Some issues such as integrity and persistence do require special attention.

2.3.3 Open Hypermedia Systems

The open hypermedia philosophy recognises the problems of the various 'closed' systems, and attempts to address them by removing reliance on any particular document formats or software or hardware platforms. The key features of the paradigm are listed here (Davis, 1995):

- There should be no limit on the number or size of objects or links entered into the system.
- Documents should remain in their native format, so that any information held in those formats is not lost through conversion. Consequently they may also be stored on read-only media.

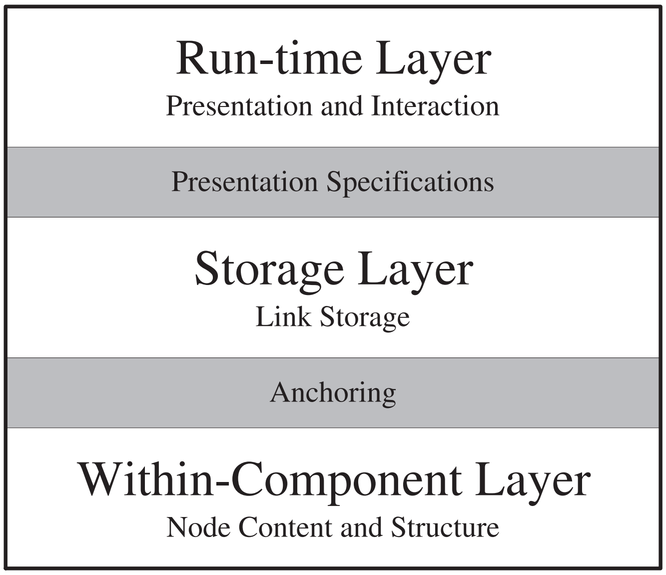


Figure 2.6: The 3-Layer Dexter Hypertext Reference Model

- No particular set of software packages or hardware should be required in order to make use of the system.
- The system should not have an in-built data model, but should allow new models to be incorporated.
- It should be possible to run the system on a variety of distributed platforms.
- The system should support multiple users and allow each to have their own view of objects in the system.

A number of systems, termed open hypermedia systems embrace this philosophy to varying degrees. Additionally, a formal model called the Dexter Hypertext Reference Model has been defined as a basis for both designing and comparing open hypermedia systems. This model is described below, followed by descriptions of some existing open hypermedia systems, with notable features highlighted. Omissions from this list are Microcosm and MAVIS, which are described in some detail in sections 2.3.4 and 2.3.5.

The Dexter Hypertext Reference Model defines a three-layer hypertext system (Halasz & Schwartz, 1994). The model divides an open hypertext system into three layers, the *run-time* layer, the *storage* layer and the *within-component layer*. This is illustrated in figure 2.6.

The model aims to provide a minimum set of operations and functionality that any modern open hypermedia system should provide. Most of the systems below have either been based on or compared to this model, and it remains an influential reference in the field.

Intermedia was originally developed as an environment for teaching systems. It is an attempt to integrate hypermedia functionality onto the user's usual desktop environment (Yankelovich *et al.*, 1988). The architecture is multi-user, and different sets of links ('webs') can be applied to documents. Anyone can author links, though these will always be visible and followable by anyone using the system.

The idea of allowing standard edit operations on hypermedia links within desktop applications is a useful one, however the architecture was implemented using A/UX, Macintosh's UNIX flavour, which has been discontinued.

Sun's Link Service is a simple open hypermedia architecture designed by Sun. Link information is held completely separately from document data (Pearl, 1989). A function library allows applications to integrate hypermedia functionality, and interoperate. New media types can be transparently introduced, though the links are untyped and the range of operations is limited. However, this was the first implemented open hypermedia architecture.

Multicard is similar in many ways to Sun's Link Service, with heavier-duty industrial applications in mind (Rizk & Sauter, 1992).

Chimera. Based around a client-server model, the primary feature of note in Chimera is the notion of separate views of the same object (Anderson *et al.*, 1994). It is on these views that links are authored. Chimera also supports n-ary links. Chimera is a multimedia system in that specific links can be authored between multimedia objects; no processing or matching of multimedia data is attempted.

DHM, or DEVISE Hypermedia, is an object-oriented hypermedia framework based on the Dexter model. It supports bi-directional links, that can have more points than source and destination (Grønbæk *et al.*, 1994). Additionally, links and documents from other systems (for example, the World-Wide Web) can be incorporated through the use of *locspecs*, abstract specifications of a location in a particular document (Grønbæk & Trigg, 1996).

PROXHY stands for Process-Oriented Extensible Hypertext Architecture, and is a four-layer open hypermedia architecture that again separates documents from links (Kacmar & Leggett, 1991). Independent processes (such as applications) communicate using a defined protocol. An object-oriented persistent store holds link information. Third-party applications not understanding the PROXHY architecture can be integrated to an extent using *proxy anchors* which contain information about how to display a particular document/object with a certain application.

The architecture is highly flexible, though high message traffic between processes can cause performance problems.

Hyper-G is a multi-user hypermedia system in which links are held separately from document data (Kappe, 1993; Andrews *et al.*, 1995a). It supports only point-to-point links. Documents (or media items) introduced into the system can be full-text or keyword searched. Hyper-G can also interact with the web and other protocols (Andrews *et al.*, 1995b).

Hyper-G is a client-server based architecture. Most of the work is done at the server. Documents are indexed as soon as they are introduced, and the server interacts with other Hyper-G servers as well as World-Wide Web servers. Link integrity is also maintained by the server. Individual users and groups of users can have different views (sets of links) and access privileges. This view information can include preferred language information.

The system is a powerful, distributed, multimedia system; however no processing, matching or interpretation of multimedia information is attempted, and the system does not support a computed link (generic link) mechanism, so the advantages of such links are lost. Hyper-G has been marketed commercially as "Hyperwave".

HyperDisco is another distributed, heterogenous open hypermedia system (Wiil & Leggett, 1996). It supports inter-tool linking, though how effective a tool can be depends on how integrated it is. HyperDisco is rather similar to other systems, differing in that the system can store the documents, rather than requiring tools to store documents themselves.

2.3.4 Microcosm

Microcosm is an open hypermedia system developed at the Multimedia Research Group in Southampton University. The system, started in 1989, was designed to alleviate some problems associated with hypermedia systems at the time (Fountain *et al.*, 1990; Heath, 1992):

- The difficulty of creating and maintaining links over large document collections.

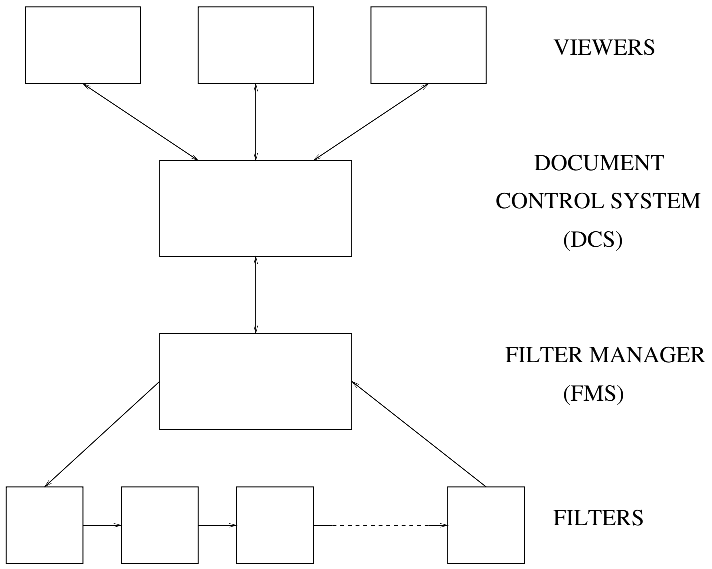


Figure 2.7: The Microcosm Open Hypermedia Architecture

- The monolithic application approach of many systems.
- The inability to link to and from documents on read-only media.
- The requirement that documents be held in a certain proprietary format.

The Microcosm architecture is an open architecture which makes a very clear distinction between documents and the links within them (Hall, 1994). Links are stored completely separately from documents in link databases, or *linkbases*. Hence, it is not necessary to be able to change or write to a document in order to be able to link from or to it. In addition, links may be held in several different linkbases, so that different links may be available or editable by different users or groups of users.

The architecture is depicted in figure 2.7. Each layer in the architecture is described below.

Viewers are the user interface components that display documents and links and interact with the user. The level of integration with the rest of the Microcosm system can vary, and falls into one of three classes:

Fully aware viewers are written specifically for Microcosm, and understand the Microcosm messaging protocol. Links can be authored in and followed from documents they display, and button links may be highlighted.

Partially aware viewers are usually existing applications that can communicate with Microcosm to some degree, for example through use of a macro language. Microsoft Word has been adapted as a partially aware viewer in this manner.

- **Unaware viewers** are third-party applications that do not understand the Microcosm protocol at all, and cannot be modified to do so. These may be started up in order to display a particular document. Some limited link-following capability may be possible through use of the system clipboard.

Document Control System. The DCS communicates with the viewers, spawning them if necessary. Any link highlighting, computation, following or authoring performed in a viewer is performed via a message dialogue between the viewer and the DCS.

Filter Manager. The filter manager or FMS manages the chain of *filters.* In order to work out whether a link is available (for highlighting or following), or to author a link, the DCS sends a message to the FMS. The FMS sends this message along the *filter chain.* The message is processed and modified by relevant filters as it is passed along the chain. When it emerges from the end of the chain, it will contain the results of whatever operation the message concerned. For example in a link following operation, the message will be filled in by each linkbase in the chain with available and relevant links.

Filters contain the core hypermedia functionality. They receive messages from the filter manager, and may react in a number of ways. A filter can 'consume' the message if it wishes, or in other words, the message does not get passed to any further filters. It may also modify the message before passing it on, and create new messages.
Filters may be linkbases, navigational aids such as histories, or other processes such as access privilege databases.

Some modifications have since been made to this architecture, including various different network topologies, and some more intelligent handling of the filter chain (Hill & Hall, 1994).

Microcosm supports typical hypermedia button-style links, provided that a viewer is 'fully aware'. In addition, it supports a dynamic linking facility known as the *generic link.*

A typical button link is a highlighted word or phrase in a document, and when clicked, the user is shown the destination of the link. For example, the word 'Barcelona' might be highlighted, and clicking on the button might open a document about the city of Barcelona. However, if the user comes across the word

‘Barcelona’ in another document, it might not be highlighted, and even though the document about the city of Barcelona may still be pertinent.

When a Microcosm generic link is authored, the location of the text ‘Barcelona’ is effectively discarded; only the fact that the link was authored on the text ‘Barcelona’ is remembered. Thus, whenever the user selects the text ‘Barcelona’, the link can be offered to the user, regardless of where the word occurenced.

Microcosm actually supports three kinds of link:

Specific links are typical point-to-point links.

Local links are authored on a term or phrase, and can be followed from any occurrence of that term or phrase within a certain document or set of documents.

Generic links are likewise authored on a term or phrase, and may be followed from any occurence of that term or phrase occuring in any document.

Microcosm has proved a usable system with some commercial success. For example, it has been used by Glaxo Wellcome plc and Pirelli Cabling. It has also been a test-bed for several experimental hypermedia techniques (Beitner, 1995; Goose & Hall, 1995). Work was also started on a distributed version of Microcosm, called Microcosm TNG (The Next Generation) (Goose *et al.*, 1996). The *Distributed Link Service* (Carr *et al.*, 1995) is a Web-based development of the ideas pioneered in Microcosm. It allows hypermedia linkbases to be distributed. Some multimedia content-based retrieval features are also being innovated (Blackburn & DeRoure, 1998).

2.3.5 MAVIS 1

Although recent versions of Microcosm allowed the authoring of links from media such as images and sound, they can only be point-to-point links in these cases. The capability for authoring *generic* links from these media is not there.

One very good reason for this is that, given a user’s selection, it is not nearly as simple to work out which links are appropriate to offer. Matching text is straightforward, whereas matching images and sound is not, for reasons explored in detail in section 2.2.3.

MAVIS 1 was an attempt to add *multimedia* generic link capability to the Microcosm open hypermedia system (Lewis *et al.*, 1996a; Lewis *et al.*, 1997b; Lewis *et al.*, 1997a), by using media processing and matching techniques to work out which links are appropriate to offer the user. MAVIS originally stood for Microcosm Architecture for Video, Image and Sound. Later on, the development and

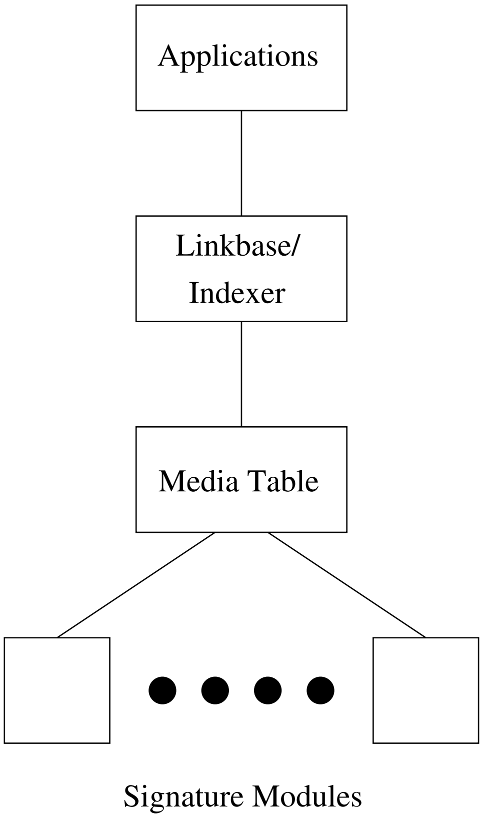


Figure 2.8: MAVIS 1 Architecture

requirements of each system diverged, such that the MAVIS architecture was significantly different from the Microcosm architecture. The name was then changed to Multimedia Architecture for Video, Image and Sound.

The original Microcosm system works out which links to offer by searching linkbases for occurrences of that term. Effectively, text information retrieval is performed on those linkbases. MAVIS works out which links to offer by performing multimedia information retrieval on linkbases. Those links whose source anchors produce a high-scoring match are offered to the user, in an ordered list, the highest-ranking match shown first.

The core architecture of MAVIS 1 (shown in figure 2.8) is modular. The actual matching methods used are implemented as *signature modules*. Each signature module deals with a particular feature of a particular medium.

The main advantages of this approach are:

- It allows future amendment and addition of media matching and indexing methods. Since these methods may be both domain-specific and constantly improving with research, allowing such updating prevents the system from going 'out-of-date' unduly early.

- Media matching and processing techniques are usually application-dependent, so the user may may wish to search for links, or author links, using a particular image feature. For example, if authoring a link from an electronic component symbol to a description of that component, the shape is a useful measure for determining whether it is relevant. The colour is not, since the symbol may appear elsewhere drawn black-on-white, white-on-black, or white-on-blue, but the link would still be relevant. Therefore, MAVIS allows the signature modules to be switched on or off by the user during authoring and following.

In MAVIS the user can select a piece of media and have it offer a number of links for following (content-based *navigation*), and in addition can elect to find all pieces of media in the system matching the selected piece of media (content-based *retrieval*). The two processes are very similar, the difference lies only in the scope, i.e. which pieces of media are candidates, source anchors of links or everything in the system.

Lewis *et al.* have produced two applications within the MAVIS framework (Lewis *et al.*, 1998). The first is an archaelogical collection, in which shape is the primary feature used to find links. The second is a product catalogue, in which the colour and texture features are most useful.

The MAVIS system, whilst a definite step forward towards a true multimedia generic linking system, still has some problems:

- While queries can be started using rectangular image subparts, they may only be matched against whole images. Thus, the source anchors of image-based generic links must be entire images. This leads to problems where a user wishes to author a link from a particular object in a scene. The features of the entire scene will be used when finding links, rather than just the features of the particular object to which the image pertains.
- Multimedia matching technology is not, in many instances, able to recognise whether two images depict the same real world object. While instances of the word 'car' are easy to detect in any text document, instances of images of cars are very difficult for a computer to detect. For example, a front view of a car is significantly different than a side-on view. Thus, if a user asks for available links from the front view of a car, links authored on side views of cars, although likely to be pertinent, are unlikely to be picked up.

2.3.6 Evaluation

The majority of existing hypermedia evaluation methods require extensive user trials or interaction. Knussen *et al.* provide a useful summary of techniques (Knussen *et al.*, 1991), though the techniques are heavily biased towards educational applications. Hutchings extensively researched the styles of interaction between users and hypermedia systems (Hutchings, 1993), again in an educational context. One of the main conclusions was that different tasks required different hypermedia structures. In general, the more formal the subject matter the more rigid the structure that can be usefully imposed over the information.

In many cases, user trials may not be a viable option, due to lack of time or other factors. Additionally, user trials inherently evaluate the interface to a system; the underlying techniques employed cannot be evaluated in isolation.

A more theoretical approach, along the lines of quantitative information retrieval evaluation would therefore be beneficial. However, the two measures *precision* and *recall* (described in section 2.2.5) are not easily applicable to the hypermedia domain. In information retrieval, the dialogue is of a very particular type: A query is specified (or refined), and results returned. In addition, calculating these values requires that the authoritative set of actual relevant objects is known.

One or two IR evaluation techniques can however be adapted to hypermedia. Subject indices, which are navigable in an hypermedia-like way, have been evaluated with such criteria as search time and number of pages turned. Craven informally defined the notion of *predictability* of a subject index (Craven, 1986). Later still, Blair formally defined predictability by relating it to *logical decisions* (Blair, 1990). A logical decision in the subject index sense is a change of focus; that is, a refinement or expansion in the subject index, or a *beamdown* to an actual document. An hypermedia application, then, can be evaluated in terms of the number of logical decisions required to satisfy a certain information requirement or find a particular item.

Salampasis *et al.* have proposed another evaluation methodology, suited to information systems in which information retrieval-style querying and hypermedia-style browsing are possible (Salampasis *et al.*, 1998). In their work, the notion of relevance is not Boolean but fuzzy. The method assumes that the information need of the user can be fulfilled with a single (known) object in the system.

The hypermedia network is represented as a matrix, in which is held the distance between every node in the hypermedia and every other node. The relevance of an object (document) is a metric of the distance between it and the object that will fulfill the user's information need. This takes into account the fact that an object reached is still a 'good match' if it is close in the network to the object the user requires. This relevance is independent of how an object is reached (by querying or browsing).

An application is evaluated by logging the actions of an user. At each interaction with the system, the relevance of the node the user is focussed on is recorded. This stops when the user has fulfilled their information need. The relevance values so generated can be used to evaluate the effectiveness of the application, and can be used to compare systems. It is also possible to perform a calculation on the relevance values to produce a single measure of effectiveness.

This technique does rely on the hypermedia collection consisting of a completely connected network of nodes; that is, there must be no 'islands'. Additionally, it may be difficult to apply to a system using dynamically-generated links such as generic links; in this case, the distance between nodes is difficult to calculate.

Garzotto *et al.* propose another hypermedia evaluation methodology, which can be performed without user trials (Garzotto & Matera, 1997). A variety of aspects of the system are defined, and then evaluated heuristically. Their methodology centres on a *model* of the system being evaluated, from which the various *dimensions* of usability can be determined. This is in turn used to define various *attributes* of the system that can be evaluated by performing some *abstract tasks*. These tasks can then be performed by evaluators, and any problems noted. This approach requires a large amount of preparation time before the evaluation can take place.

These evaluation techniques are geared to estimating the effectiveness of a particular hypermedia *application*; is is difficult to apply them with any confidence to an information system *framework* without evaluating purely the application within that framework itself.

2.4 Hybrids

Some limited work has been done on combining the two main paradigms of information access. Specifying an information need in the form a of query is one of the key tasks in effective information retrieval, and one that takes experience to be able

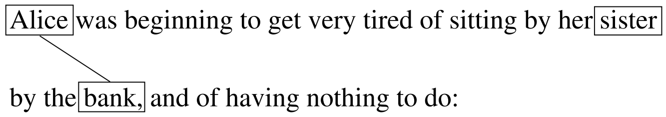


Figure 2.9: Query Formulation in Queries-R-Links

to perform effectively (Salton & McGill, 1983). The aim of a hybrid approach is to introduce the ease of hypermedia browsing to query formulation.

An interesting idea that makes use of hypermedia structure to allow text retrieval of images (and associated text) is presented by Dunlop *et al.* (Dunlop & van Rijsbergen, 1993). Clusters are formed of closely-connected hypermedia nodes, and the text within the clusters used to represent the cluster as a whole. Thus retrieval operations will retrieve the whole cluster, including multimedia content, and the hypermedia structure can be used to navigate around the cluster (and to nodes outside the cluster).

Descriptions of two systems that combine information retrieval and hypermedia follow.

2.4.1 Queries-R-Links

In Golovchinsky's *Queries-R-Links* system, the user can select one or more text terms, usually by clicking on them in a similar manner to hypertext link following (Golovchinsky & Chignell, 1993). Selecting a term adds it to the query, and the new set of documents satisfying the query are displayed. The user can specify boolean queries by drawing lines between the 'and' terms. 'Or' is implicit when no line exists. Figure 2.9 shows an example of this. In the example documents must contain both 'Alice' and 'bank', OR contain just 'sister.' A document containing 'Alice' is not returned; a document containing just 'sister' is, and so is a document containing 'Alice' and 'bank.'

The user can quickly add terms to the query by selecting new terms and connecting them to existing terms if appropriate. A derivative of this system has been used by Golovchinsky to prototype a newspaper archive information system (Golovchinsky, 1997).

2.4.2 TACHIR

Agosti *et al.* advocated an approach for the automatic authoring of an 'IR hypermedia' (Agosti *et al.*, 1995; Agosti *et al.*, 1996). This is performed by extracting index terms, and using thesaurus concepts as a higher-level means of navigation.

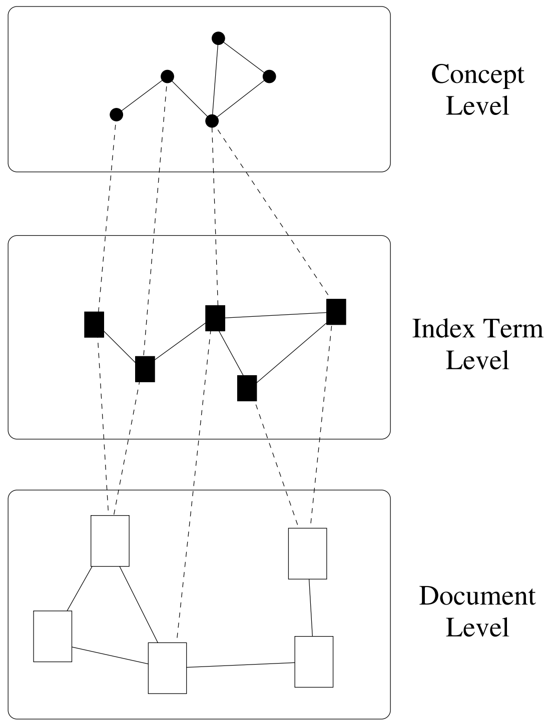


Figure 2.10: Three-layer Hypermedia Schema in TACHIR

These index terms and concepts become nodes in the hypermedia. A map of the resulting hypermedia is shown in figure 2.10.

A brief explanation of each layer follows:

Concept Level. This consists of thesaurus concepts that have effectively become nodes in the hypermedia. Thesaurus relationships are the basis for concept-concept links.

Index Term Level. This layer holds the extracted index terms. These are now hypermedia nodes, linked to each other based on statistical co-occurrence in the document collection. They are also linked to concepts based on their appearance as terms in the thesaurus, and to documents in which they occur.

Document Level. This consists of the original document collection. The inter-document links are either generated from citations, or from statistical similarity measures.

Thus, given a document collection and a relevant thesaurus, this technique allows the automatic construction of an hypermedia navigable at a variety of levels.

Although in the literature the architecture is referred to as a 'multimedia' architecture, it is only pseudo-multimedia in that non-text media objects are treated as opaque 'blobs'. No consideration is given to their content, and they appear only by virtue of being attached to a text document.

2.5 Summary

Two main methods of information access have been developed in recent years, information retrieval and hypermedia.

Information retrieval, especially of text, is the more mature field. Techniques for retrieving text have reached a high level of competence, and a variety of application-specific methods and retrieval models exist for a number of media types.

The vast majority of all methods of information retrieval produce, given a query and a set of objects from which to retrieve results, a ranked list of objects best matching the query. The techniques themselves are necessarily domain-specific, but abstracting above this level enables information retrieval to be considered in a general sense.

Notably, the following features in information retrieval system have proved effective:

- Fully user-controllable relevance feedback.
- User-controlled or automatic weighting of query terms (in simple queries).
- Probabilistic and vector-based models, yielding higher performance than Boolean models.

Empirical evaluation of such systems is well-researched, provided that a 'ground truth' is known.

While information retrieval enforces a query-result style of dialogue, hypermedia represents a more exploratory method of information discovery. An open approach, separating links from documents, provides greater flexibility, especially since different structures and methods of presentation are appropriate for different tasks.

Evaluation of hypermedia systems is tricky, usually relying on user trials and feedback, though a few empirical measures can be adapted from the information retrieval field.

The systems and techniques in this chapter have not attempted to explicitly process the *semantics* of the information they contain. While it may be argued that by constructing links, the user of a hypermedia system is introducing semantic information (Schnase *et al.*, 1993), it is not treated as 'knowledge' by the system, which still operates purely in the *plane of expression*.

The following chapter describes methodologies and systems that explicitly attempt to model and exploit the semantics of the information within them.

Chapter 3

Semantics

3.1 Introduction

Largely, systems described in the previous chapter paid little attention to the *semantics* of information held in the system. Measurable, quantifiable qualities of media objects were used to retrieve and explore information. Some hypermedia systems do provide semantic information, and the organisation of the hypermedia may be semantically based (Wang & Rada, 1998). However, there is still no *understanding* of the media by the computer in any sense; the semantics have just been created largely by human authors in the construction process.

This chapter concerns systems that attempt to capture and make use of the *meaning* of data; that is, they operate in Smoliar's *plane of content* (Smoliar & Wilcox, 1997).

Thesaurus systems are investigated first, since these are the oldest and most widely-used and recognised semantic systems. There follows an overview of knowledge representation techniques. Finally, existing information retrieval and navigation systems using semantic content are discussed.

3.2 Thesaurus Systems

Broadly speaking, a thesaurus is a set of terms connected by a set of relations (Jones *et al.*, 1994). Most people's contact with thesauri will be in the context of a book or thesaurus feature in a word processing application. The reader or user looks up (or selects) a term in the thesaurus. They will be presented with a number of synonymous terms, and terms semantically related in some way. This is effective at helping people with 'word block' when writing documents. In an information retrieval sense they form the backbone of an indexing and retrieval system. Examples of commonly used thesauri are the INSPEC classification thesaurus (Institution of Electrical Engineers, 1991), or the Arts and Architecture thesaurus (The J. Paul Getty Trust, 2000).

Thesauri may be used in information retrieval systems in a number of ways:

- They can be used as a controlled vocabulary with which to describe documents. Queries may then be specified or translated to this controlled vocabulary, improving retrieval effectiveness.
- Documents may be indexed using the most specific available terms in the thesaurus. These documents are then automatically indexed on the generalisations (broader terms) of those terms. Thus a query can be expanded or refined by viewing documents indexed on broad (general) or narrow (specific) terms.
- Thesauri can be used for expanding free text queries, supplementing queries with synonyms of the original query terms.

3.2.1 Structure/Relations

The majority of thesauri use five main relations (or close equivalents):

Broader term. The *broader term* of term A is a term encompassing the scope of term A.

Narrower term. The complementary relationship of *broader term* - the scope of a *narrower term* B of term A is encompassed by the scope of the term A. If this relationship exists, term A is always the broader term of term B (Aitchison & Gilchrist, 1987).

Equivalent term. Term A is *equivalent* to term B if term A is synonymous with term B. In a thesaurus with a relatively low level of detail (specificity), term A may in fact be broader or narrower than term B, if the difference in specificity is small enough.

Preferred term. This relation is very much like *equivalent term*, however this relation is used to indicate the preferred (often the most common) form of the term.

Related term. Used when term A and term B have some semantic association, but their relationship cannot be adequately expressed in terms of one of the above relations.

If the thesaurus as a whole is considered as a concept map, the *related term* relationship can be considered as a mapping for all non broader/narrower relationships (Jing & Croft, 1994).

The above relations may be given weightings, depending on the strength of the connection. This is particularly useful for *related term*, where information about the connection is very limited.

A thesaurus, then, can be considered a map of some subject domain (Jones *et al.*, 1994). There are a number of different ways of visualising this map:

Arrowgraph. A graphical representation, with physical distance between terms on the diagram related to 'semantic distance' (how closely the terms are related). These diagrams are difficult and time-consuming to produce and maintain, although methods are emerging for automatic generation of these or similar diagrams (Gaines & Shaw, 1995).

Flat Map. Essentially the thesaurus is a collection of nodes (terms) and links (relations) with no structural restrictions. Thus it can be considered simply a flat concept map. This approach offers no advantage and does not aid navigation of the map in any way (Aitchison & Gilchrist, 1987).

Hierarchy. By far the most common visualisation, this approach assumes that the most useful relationships in the thesaurus are of the broader/narrower type. The terms are arranged as an hierarchy, moving up the hierarchy involving traversing broader term relationships, moving down the hierarchy traversing narrower term relationships. Typically the thesaurus has a number of broad subject headings (known as *facets*), below each of these is an hierarchy of terms encompassed by this heading. A set of equivalent/preferred terms are considered as single nodes in the hierarchy, or commonly only the preferred form of a term appears. Although related term connections mean the hierarchy is not strict, the hierarchical core of the thesaurus is used as the basis for visualisation.

3.2.2 Further Definitions

There are a number of other important aspects to thesauri. Terminology used to describe them are given below.

Specificity refers to how detailed the terms in the thesaurus are (Aitchison & Gilchrist, 1987). A term or part of a thesaurus with high specificity covers a relatively narrow subject field. A term or part of a thesaurus with low specificity covers a relatively broad subject field.

Controlled/Free Language. In a controlled language thesaurus, only a controlled subset of all terms in the thesaurus are actually used to index documents or represent the relationships between concepts (Aitchison & Gilchrist, 1987). Other terms in the thesaurus 'lead in' to the preferred term for a particular concept.

A controlled language thesaurus aids retrieval at the (possible) cost of specificity. This means that given a search term, a controlled language thesaurus is likely to retrieve a number of documents, though the chances of each of these documents being relevant to the searcher is reduced.

Faceted Classification. Each term in a thesaurus may also appear (usually hierarchically) beneath one of a group of *facets* (Vickery, 1960). Each facet reflects one aspect of the overall subject being covered. A facet may simply describe the fundamental type of the term, for example 'Process' or 'Material,' or may describe the term in more detail. For example in a thesaurus describing chemical manufacture, there may be facets 'Raw Materials,' 'Catalysts,' 'End Products,' 'Apparatus,' and 'Processes.' For a more general thesaurus, a more generic classification schema may be adopted such as Ranganathan's *PMEST* formula (Ranganathan, 1959) ('Personality,' 'Matter,' 'Energy,' 'Space' and 'Time'). However, in all cases this classification schema is predetermined and considered too time-consuming to alter once the initial thesaurus has been constructed (Vickery, 1960).

By far the most common form of thesaurus is a controlled language, faceted hierarchical thesaurus. Other structures exist such as multitrees (Furnas & Zacks, 1994) but these tend to be suited to particular applications.

In the vast majority of cases, documents relevant to a term are indexed on the 'preferred form' of that term. Other relations such as 'broader term' are indicated only between preferred forms — the relationship exists implicitly between synonyms of those terms.

A query term that appears in the thesaurus 'leads in' to the preferred form of that term, and from there documents concerning that term can be simply retrieved. Terms other than the preferred term are thus called *lead-ins*. An example of this can be demonstrated in figure 3.1. If the query term is 'automobile', the lead-in

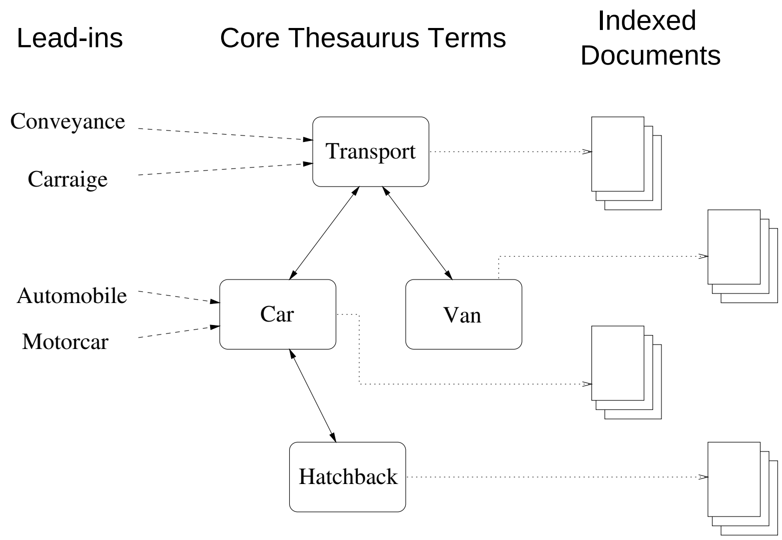


Figure 3.1: Leading in to a Preferred Term

'automobile' is used to find the core term 'car'. From that term indexed documents can be retrieved, or other related terms explored.

It should be noted that a 'lead in' term may actually lead in to more than one preferred term (Jones, 1993). Most electronic systems will offer the possibilities to the user for their selection in such a case, or use further evidence such as other terms in the query to make a choice of preferred term.

Obviously, to be useful, a query term must appear in the thesaurus in some form, so the language in the thesaurus must be comprehensive to adequately cover the subject area.

3.2.3 Thesaurus Construction

Construction of a thesaurus, or thesaurus system, can be divided into two sections. The first section is the construction of the underlying terms and structure; the second is the indexing of documents in that structure.

Automatic generation of thesaurus structures is not a new field, research in the automatic classification of terms goes back as far as the 1960s and earlier (Salton, 1968; Sparck Jones & Needham, 1968). However, with the advent of widely used electronic information retrieval systems and high-power computing, far more complex methods can be used prompting more recent research in the area.

A number of methods exist to automatically create thesauri, or similar semantic structures. All investigated are text-based, using text analysis techniques to identify candidate terms (Cunliffe *et al.*, 1997; Jing & Croft, 1994). An iterative process known as relevance feedback, involving analysing the context of a term in many different documents, can be used to refine this process (Jing & Croft, 1994; Golovchinsky, 1997). These methods may prove useful as a method of constructing an initial thesaurus for an application, but are outside the scope of this research.

More commonly, a set of terms and relationships is chosen by human authors. Many such sets have existed and been researched and extended for many years. For example, the Dewey Decimal Classification system, while not strictly a thesaurus, is a similar structure that was conceived in 1873 and first published in 1876.

Once this set of terms is decided, documents are usually indexed using some automatic process (Aitchison & Gilchrist, 1987). A common and simple way to do this is to index (text) documents with all (preferred forms of) terms appearing in the document.

3.3 Knowledge Representation

There are a number of methods for representing semantic knowledge in a computer system. These have been extensively researched in the artificial intelligence field (Sowa, 1984).

By far the commonest form of explicitly stored knowledge representation is written natural language. In common with all knowledge representation schemes in common use in information systems, it is a symbolic medium. There have been many attempts to analyse its structure (Vickery, 1986), with some success, but still a computer cannot analyse and intepret the knowledge implicit or conveyed in natural language prose except in very restricted domains.

One alternative method of storing knowledge that is more suitable for a computer is predicate logic, or 'predicate calculus' (Charniak & McDermott, 1985). Knowledge is represented symbolically as a set of *predicates*, which state facts. The ubiquitous example is the syllogism:

$\forall x \mid human(x) : mortal(x)$

$human(Socrates) \rightarrow mortal(Socrates)$

This states the fact that if x is human, then it can be *inferred* that x is also mortal. Thus, if Socrates is human, Socrates is mortal.

An alternative, but formally equivalent method of representing knowledge is the *semantic net*. An example of a semantic net is shown in figure 3.2.

Interestingly, a thesaurus may also be visualised as a semantic net with a small number of relationships, and has been explicitly by Jones (Jones, 1993). It also follows that a thesaurus could be expressed in predicate calculus. So, when designing

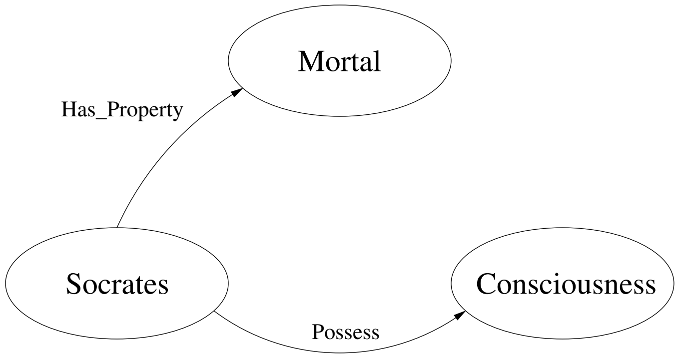


Figure 3.2: Example of a Simple Semantic Net

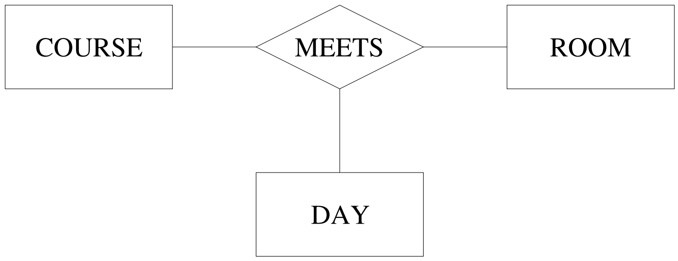


Figure 3.3: Example of an n-ary Relationship

a system to hold knowledge, a system that can hold knowledge in either predicate calculus or semantic nets will also be able to hold thesaurus information.

Mathematical set theory is the basis of the *relational model*, extensively used in database systems to store semantic information. Relationships are stored as ordered sets of (usually symbolic) entities, which can be manipulated using set algebra. A well-known graphical representation of this model is the *entity-relationship* model (Chen, 1976). The basic form of this model consists of three elements: *Entities*, *Relationships*, and *Attributes*.

Entities represent classes of real-world objects.

Relationships between entities can be expressed. These may be n-ary relationships, or *rings* which connect an entity to itself. Examples of these are shown in figures 3.3 and 3.4 respectively.

Attributes may also be associated with entities.

The model has been extended to include other features such as composite attributes and generalisation hierarchies (Codd, 1979). While the relational model is not equivalent to predicate calculus, the knowledge in a relational model can be represented in predicate calculus. Thus, it can be seen that knowledge held in any of these representation schemes can also be represented as predicate calculus.

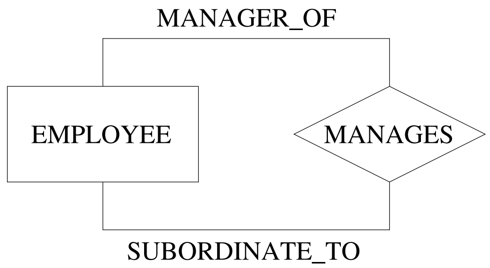


Figure 3.4: Example of a Ring Relationship

Knowledge representation is extensively used in the field of software agents. The knowledge used by software agents are referred to as an *ontology*. Huhns *et al.* define an ontology as "a computational model of some portion of the world" (Huhns & Singh, 1997). However, the term 'ontology' does not imply a particular form of knowledge representation; for example an agent could hold knowledge in the form of predicate calculus or a semantic net.

3.4 Classification

At both the construction and querying phases in a thesaurus' life cycle, and in many other situations, it is necessary to work out where a query term, object or document best fits.

- During the indexing stage of constructing a thesaurus, documents need to be automatically indexed on the most appropriate terms. A variety of methods exist for this in the text domain (Salton, 1989).
- During a query operation, each query term or phrase must be located in the thesaurus in order to find indexed documents or synonyms.
- Other applications require that objects are automatically categorised somehow, since performing that categorisation manually is prohibitively time-consuming.

These tasks are rather simpler for text than for multimedia, since it is easy to work out where a text term appears (if at all) and what terms appear in a text document. Many techniques for this 'pigeonholing' exist and have been extended to the multimedia domain. These techniques are known as *classification* techniques.

Li *et al.* review four of the main classifier types (Li & Jain, 1998):

Naïve Bayes classifiers estimate the probability that a document $\mathcal{D}$ belongs to a class $\mathcal{C}$ using Bayesian estimation, as described in section 2.2.1. Such classifiers assume that features are independent (Mitchell, 1997). The main disadvantage of Bayesian classifiers is that a parametric expression for the denominator of equation 2.2 must be specificied *a priori* or estimated. In other words, an equation reflecting the distribution of documents with respect to classes is required.

Nearest neighbour classifiers assign a document $\mathcal{D}$ to the closest class in the feature space (or vector space, as defined in section 2.2.1). Either Euclidean distance or a weighted cosine similarity measure is used. This technique is non-parametric; that is, no expression is required *a priori*.

Decision tree classifiers classify objects hierarchically (Gordon, 1987; Safavian & Landgrebe, 1991). Starting at the root of the tree, a decision is made at each node (typically based on a single feature) as to which subclass the object belongs to, and this continues until either a leaf is reached or a best class found.

Subspace classifiers divide the feature space into spaces of lower dimensionality, and classify documents based on their compressed representation in each of these subspaces (Oja, 1983; Li & Jain, 1998).

It is also noted that combining evidence from different classifiers also produces more accurate results (Larkey & Croft, 1996). There are a number of methods for performing this, the most common being a simple voting system. This problem of combining evidence is a highly active area of research.

To summarise, there are a wealth of techniques available, all of which have the basic aim of assigning a document (or object) $\mathcal{D}$ to a single class $\mathcal{C}$ from a set of classes. A system requiring this procedure could beneficially leave that functionality open to allow the application of varied and new techniques.

3.5 Information System Work Using Semantics

Multimedia retrieval or navigation systems that involve the explicit use of semantics of the data are rather thin on the ground. The few that exist are described here.

3.5.1 Nubila's Concept-based Indexing

Nubila *et al.* have produced a method for indexing multimedia information by concept (Nubila *et al.*, 1994). The system works by analysing narratives of radiological slides and producing semantic representations. The 'concepts' are words (or collections of words) in the narrative. This allows some complex queries to be specified an evaluated. However, it does rely on there being text associated with the multimedia objects, since it is from this text that the semantics are derived — as with so many "multimedia" techniques, the non-text media is treated as a set of opaque data objects.

3.5.2 *MACS*

Brink *et al.* have mathematically defined a *media abstraction* (Brink *et al.*, 1995). This is a mathematical way of representing the properties of any given object, regardless of medium. The Multimedia Abstraction Creation System (MACS) system implements algorithms for creating and querying these abstractions. This has been embedded in Hermes, a Heterogenous Reasoning and Mediator System developed at the University of Maryland. Together, they provide a powerful mechanism for querying multimedia data. However, it requires a large amount of human effort in initially applying semantics to media objects.

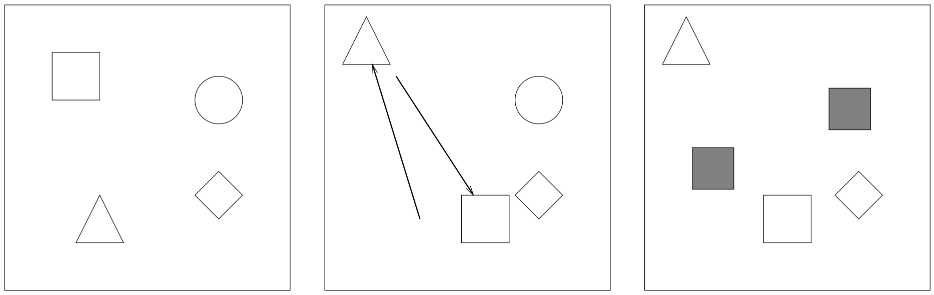
3.5.3 *El Niño*

A multimedia database system called El Niño developed by Santini *et al.* (Santini & Jain, 1999; Santini, 1998) provides an exploratory interface to a multimedia database. Rather than being based around a query-response style dialog, the system is more concerned with information *exploration*.

Instead of relying purely on feature similarity, the interface of the system invites the user via a direct manipulation interface to give some indication of their own perceptions about the similarity of images. In this way, the system is said to be operating on *emergent semantics*.

The system presents an arrangement representing the system's idea of similarity between images (figure 3.5(a)). The user can move around the images themselves such that the display is closer to their own perception of the similarity between the images (figure 3.5(b)). The system then works out a new similarity measure that matches the user's, and the display is updated according to the results of this (figure 3.5(c)). Some previously hidden images may appear, such as the grey rectangles in figure 3.5(c), and some images may disappear from the display, such as the circle in figure 3.5(a) and 3.5(b).

Since it is impractical to display all (appropriate) images in the database onscreen at the same time, a group of similar images (in the context of the current application) can be grouped together *by the user* to form a *visual concept*. This is effectively treated as a single image in further manipulations.



(a) (b) (c)

Figure 3.5: El Niño's Direct Manipulation Interface

The database uses geometric transformations to map the user's perceptions of similarity to the feature space. The technique is novel and interesting, but there are still unresolved issues:

- How the initial arrangement of images (e.g. figure 3.5(a)) should be arrived at is undecided.
- There appears to be no way that one user can 'imprint' their own notions of the semantics of images in the system in such a way that other users can see or make use of them.
- The system (currently) only works with images, and is still geared towards the idea of finding images similar to a particular example or examples. The idea of finding information *relating to* a particular image is absent.

3.5.4 The Himotoki System/COIR

Research with possibly the closest goals to those of this research has been undertaken by Hirata *et al.* at the C & C Research Laboratories, NEC USA Inc. An early paper describes work on *media-based navigation*, which allowed the use of content-based retrieval to navigate through images, though this work did not have a semantic component (Hirata *et al.*, 1993).

Later work describes a system based around the idea of 'object-based navigation' (Hirata *et al.*, 1996; Hirata *et al.*, 1997). The idea behind this is to navigate between the objects represented in media items, rather than the media items themselves. Their system apparently has the capacity for purely media-based navigation as well.

Each object for which there exists a piece of media has a corresponding *conceptual representation*, which is essentially a label for the object. This conceptual representation is bidirectionally linked to each piece of associated media. The user can navigate either using purely media, or may choose to navigate via the conceptual representations.

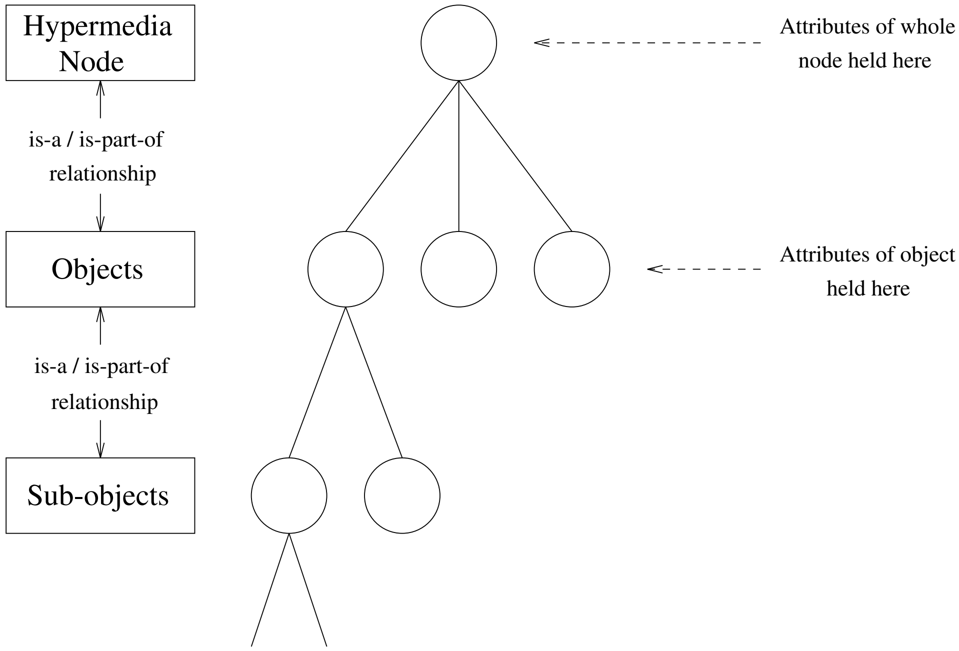


Figure 3.6: Hierarchical Attribute Structure

Visual characteristics (effectively image features) are used to extract conceptual representations from media objects, and attributes assigned automatically. Users can also assign semantic information to the conceptual representations.

A later paper of Hirata's (Hirata *et al.*, 1997) describes an extension to the content-oriented integration architecture described in the previous paper called *object-based navigation*. However, it is re-presented as a generic set of hypermedia tools called the *Content-Oriented Information Retrieval* tools (*COIR*).

Multimedia data in the system is broken down into *objects*. The user can form a query specifying multiple objects and the relationship between them. To allow this, the attribute data of each piece of media is hierarchically structured. This structure is shown in figure 3.6.

Each node (media object) is split up into objects. Each of these objects can be further sub-divided. This decomposition is stored as a tree in which each node corresponds to one object or sub-object. The root of the tree represents the whole node.

At each node of the tree (including the leaves) are held the attributes of that object or sub-object. These include media-based attributes, and a possible association with a conceptual representation. In addition, the physical relationships between the direct children (sub-objects) are held, such as positional information from an

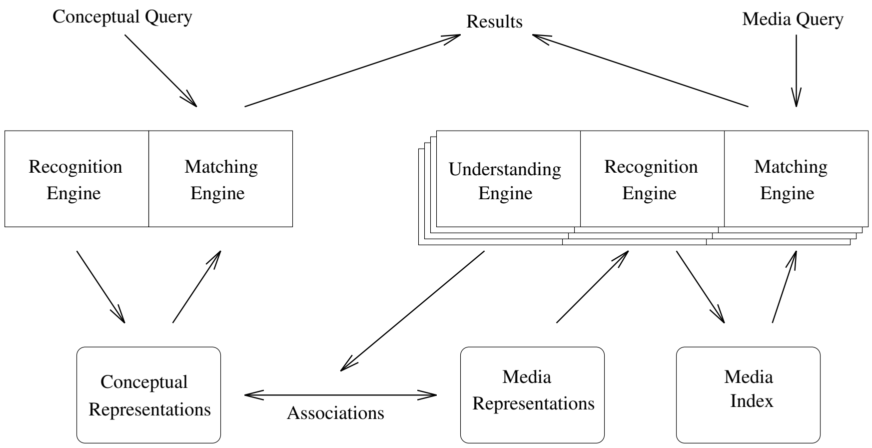


Figure 3.7: Hirata's Content-Oriented Integration Architecture

image. The logical relationships between each object/sub-object are held with the linking branches. These are bidirectional is-a/is-part-of relationships.

Structural constraint (specifying relevant nodes in the tree) and attribute constraint (specifying relevant attributes at each node) can be applied when matching to ensure the relevant qualities of the media/object are matched against.

During a query, users can specify a number of objects and relationships between them. Since this can be a complex operation involving the choosing of relevant media-based attributes, default settings can be used allowing navigation to be achieved using just mouse clicks. The query handler extracts the relevant attributes from the objects.

Technical Details

The system has a fairly generic architecture, with modular parts enabling for example new media processing and understanding technology to be supported. An overview of the architecture is shown in figure 3.7.

Media and conceptual representations are dealt with by the media and conceptual *augmenters* respectively. Each augmenter has a *recognition engine* and a *matching engine*. In addition media augmenters have an *understanding engine* which translates a media representation into a conceptual one. Since conceptual representations are already 'understood' and associated with media representation(s) the conceptual augmenter does not require this engine.

The features of each of these engines are as follows:

Recognition engine. The media recognition engine extracts media attributes and stores them in a media index. This may consist of many different modules for different recognition algorithms. The conceptual recognition engine indexes the conceptual representations.

Understanding engine. This engine uses the media attributes extracted by the recognition engine to attempt to find a corresponding conceptual representation for the media representation.

Matching engine. This object, when given a query, finds the matching media or conceptual representation. The media matching engine also employs *media-based clustering* techniques. If the query is far from a cluster then that cluster is eliminated from the search.

The matching phase is divided into four steps. Unfortunately this process is not clearly described in detail.

1. The initial phase described in the paper as "Correspond the objects with each other." It is assumed that this step involves finding objects from the media index that match those of the query.
2. "Evaluates the similarity between each of the correlated objects." This is taken to mean that the strength of each match is calculated using media and conceptual attributes. These matches may be weighted by the user.
3. "Evaluates the similarity of the relationships." Where strong matches are found, the similarities of both physical and logical relationships between them are evaluated.
4. "Evaluates the similarity between the query and the media index." This can only be assumed to collate the results of steps 1–3 above and return the results.

The system promises much, such as the ability to navigate around the 'real-world' objects. However there are gaps, and reasons to doubt the practicality of some aspects of the approach.

- No mention is given of how the conceptual schema is initially constructed, nor how relationships between concepts are created, nor of how media representations are associated with conceptual representations. To perform these tasks manually would require an enormous amount of time and effort for any sizable application.
- How, given a piece of media, the system extracts 'real-world objects' from it is not explained. It is suggested that the system can accomplish this from "a single mouse click". That the system can automatically delineate objects in a scene in an image, and extract relevant features from it, seems highly unlikely

considering the wealth of literature suggesting that this is impossible given current image processing research (Sonka *et al.*, 1993; Jain *et al.*, 1995) other than under tightly controlled sets of data. Achieving this 'by hand' during a construction phase would be far too time-consuming to apply to any medium to large sized dataset.

- The definition of semantic relationships between objects (other than spatial relationships) appears to be entirely human-authored, requiring a prohibitively large authoring effort in all but small applications.

3.5.5 Latent Semantic Analysis

Latent Semantic Analysis, or LSA, is a theoretical technique for extracting the meaning and context of words by statistically analysing large bodies of text (Landauer *et al.*, 1998). It as been found to simulate a range of human cognitive phenomena; these include discourse comprehension and even judgements of essay quality.

One of its main features is that it induces correlations between words and topics without those correlations having to be explicitly specified. This has obvious useful applications in information retrieval, since it tackles two of the principle problems in the field — the differing terminology problem, and the context problem.

LSA obtains its 'knowledge' by first processing a large corpus of text. The words in it, and sets of these words, are represented as points in a high dimensional "semantic space".

Initially, a matrix is constructed, in which each row corresponds to a word, and each column to a passage of text. The values in the matrix are set to a weighted indication of the frequency of the word in each passage.

Next, a process called singular value decomposition (SVD) is applied to the matrix, which decomposes it into three other matrices of much smaller dimension. These three matrices, when multiplied together, produce an approximation of the original matrix. It is when this dimensionality is decreased that the semantic relationship between words is induced; hence the name "Latent Semantic".

These three matrices can be used to derive vectors for each word or passage; the vectors can then be used to judge the similarity between topics. The cosine between the vectors is usually used; the higher the cosine (the closer to 1) the greater the similarity.

One disadvantage, of course, is that the initial matrix must be constructed using large volumes of text in order for it to be of any use. However this has already been

Artificial_Intelligence	Psychology_French
Biology_HS_betatest	Psychology_GleitmanCog
General_Reading_up_to_03rd_Grade	Psychology_Myers_4th_ed
General_Reading_up_to_06th_Grade	Psychology_Myers_5th_ed
General_Reading_up_to_09th_Grade	Zsink-override
General_Reading_up_to_12th_Grade	encyclopedia
General_Reading_up_to_1st_year_college	energy
Latvian_Dainas	heart
Mesoamerican	kftest
Mesoamerican2	smallheart

Table 3.1: Subjects of Existing LSA Semantic Spaces

performed at Colorado University; matrices have been created from a variety of sets of text, covering a variety of subject areas. These are listed in table 3.1.

The LSA technique has already been applied to information retrieval, in the form of Latent Semantic Indexing (LSI) (Dumais *et al.*, 1988; Letsche & Berry, 1997). In these cases, retrieval relies on there being a large collection of texts in the first instance to construct the initial matrix.

The technique has also been applied to the multimedia domain to an extent by Sclaroff *et al.* (Sclaroff *et al.*, 1999). The LSA technique is not applied to multimedia information directly, but to text and HTML tags surrounding each image. The points in the semantic space and the points in the visual feature space are combined to make a single feature space, and this used as the basis for multimedia retrieval. This is a useful technique, but it still relies to some extent on text being associated with images.

3.5.6 van der Heijden's Domain Concept to Feature Mapping

Gerie van der Heijden *et al.* have pointed out that the most intuitive way of retrieving images is by domain concept (van der Heijden & Worring, 1996). They indicate that a vital step in achieving this is by mapping image features to domain concepts. Figure 3.8 shows how this is achieved.

The mapping is achieved using standard statistical image classification techniques, demonstrated with a plant variety testing application.

In their work, van der Heijden *et al.* do not use the mapping for anything other than classification and retrieval of images in the same class as the query image. However, the work does put image classification in a slightly different and more useful perspective.

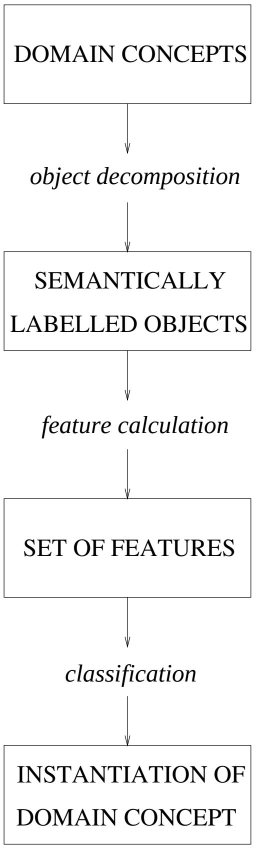


Figure 3.8: Domain Concept to Feature Mapping

3.5.7 Semantic Hypermedia Architecture (SHA)

In an architecture proposed by Cunliffe *et al.* information items in the media base are indexed as relationships which can be queried (Cunliffe *et al.*, 1997; Tudhope & Taylor, 1997). These are expressed as binary relationships in the form:

`< OBJECT RELATIONSHIP SUBJECT >`

RELATIONSHIP is one of a finite set of relationships allowed by the system such as *A-Kind-Of*. These include hiearchical specialisation/generalisation relationships, so these relationships can be used to form a graph-style network of the knowledge base with a hierarchical core.

The database of relationships can be queried based on the *Standard Associative Form* which allows queries such as:

`< ? A-Kind-Of Community-Life >`

These relationships are also weighted according to the strength of the relationship, and the specialisation of the subject and object. The reason for the latter is that two very specialised concepts close together in the network are likely to be more similar than two more general concepts.

Each of the concepts is associated with retrievable media items. The relevance of each media item can be assessed by the weightings attached to relationships, and the number of relationships traversed between concepts.

Hypermedia functionality is achieved through passing a query from an application (which may be a user's selection or a highlighted button) to the database of relationships. The results of the query can then be used by the application to offer other documents in which related terms appear, or as means to navigate around the relationships (or the 'index space') itself.

The system has no capacity for understanding multimedia content, and the initial effort involved in authoring the semantic database may be very large.

3.5.8 TourisT

Another system based on a similar idea to Cunliffe's work is *TourisT* (Bullock & Goble, 1998). A 'description logic' is used to compose a conceptual model of the subject matter, tourist information. This model is used to construct a conceptual index of the information held in the system.

Each document in the system must be given concept descriptors, that can easily be mapped to concepts in the conceptual model. Once this is achieved, the richness of the conceptual model allows a variety of styles of navigation:

Hypermedia links can be generated by searching for documents associated with related concepts in the conceptual model.

A Hypercatalogue Browser allows the user to navigate around the conceptual model, which contains multiple hierarchies.

Query by Example Document by finding documents with similar concept descriptors to the query document's.

Although there are some automatic elements in the initial construction of the conceptual model, a considerable amount of human effort is still required. Additionally, the system cannot handle non-textual media without associated metadata. However, importantly, the work does demonstrate the effectiveness of holding semantic information separately from media itself.

3.5.9 The OKAPI Projects

OKAPI is an experimental text retrieval system that includes a thesaurus element (Robertson, 1997). In a paper detailing experiments with user interfaces (Beaulieu, 1997), two different methods of thesaurus use are mentioned.

The first involves explicit use of the thesaurus. Given the terms of a query, synonyms and closely related terms from the thesaurus are presented to the user, which the user may add to the query. These terms may also be used to enter and navigate the thesaurus, from which additional terms may be added to the query. No details concerning the thesaurus navigation process are given in the paper.

Implicit use of the thesaurus takes the form of matching terms in the thesaurus and using synonyms (and maybe connected terms) to expand the query, *without* interaction with the user. However the paper does not detail the relative benefits or shortcomings of this approach as opposed to the method described above. The paper describes both methods as being useful, though the user's selection of relevant terms from the thesaurus was noted as being especially valuable.

3.5.10 The VISAR System

The *VISAR* system presented by Clitherow *et al.* (Clitherow *et al.*, 1989) attempts to improve retrieval by finding networks of concepts from a semantic net similar to that expressed in a query as opposed to media items. The user can then reach documents indexed by these concepts.

The system has available a potentially huge collection of documents. A semantic network is formed by using natural language processing (the "Lucy" system) — basically it is used to deduce "who is doing what to whom".The network is hierarchical in nature.

At any given time the user has a "focus of attention" — this is a small (and simplified) portion of the network that the user is interested in, with weightings assigned to each of the entities, which number no more than seven. (This is of course George Miller's "magic number" (Miller, 1956) — the capacity of the human short term memory.) This network is known as the user's default *perinrep* — *personal information representation*, the initial state of which is decided using a number of factors including the user's occupation and social background.

Natural language processing is used to parse a query into conceptual units which must exist in the overall semantic network. Relationships between these units are inferred and a small network of concepts results. This is combined with the user's default perinrep to produce the request perinrep. A technique called *highest common concept* is used to achieve this.

The request perinrep can then become the user's default perinrep (focus of attention). The user is free to expand/collapse concepts and navigate through the overall network, or ask for some relevant documents.

To retrieve documents, the concepts in the query and the request perinrep are used to perform keyword searches of the documents held in the system.

The prototype VISAR system only holds titles but the whole document could be held and reached from these. The results yielded by this process have a high chance of being relevant to the user even if the wording is significantly different to that of the query. The system only recognises text, and uses only the title of documents and not the content. In addition the querying process relies on the quality of the natural language processing technology used.

3.5.11 SemQuery

Work overlapping with this research in two areas is the work on the *SemQuery* image retrieval system by Sheikholeslami *et al.* (Sheikholeslami *et al.*, 1998). SemQuery retrieves images using a semantic clustering technique using multiple centroids for each cluster. These clusters are called *semantic* clusters because they represent some real-world concept, for example 'wood'.

The system also makes use of multiple feature types, which may span texture, colour, shape and other features.

A hierarchy of real-world concepts is stored in a database, and the clusters represent concepts at the lowest level of this hierarchy only. The system builds clusters using a training set, and uses neural network techniques to weight each feature type.

An additional feature of note is the ability to specify a subtree of the hierarchy to narrow the scope of the retrieval, improving retrieval effectiveness.

3.5.12 RECI

The RECI content-based image retrieval system developed at the IRIN Computer Science Research Institute at the University of Nantes, France, features facilities for expressing semantics when indexing and retrieving images, and has some knowledge discovery capabilities (Bouet & Djeraba, 1998).

The system first clusters a set of images, based on both visual features and associated text. The system then induces a number of rules for identifying images of each class. These take the form of implications, for example:

$(text, text2) \Rightarrow (colour, colour2)$

This means that if an image has the associated text *text2*, then it is likely that if the image is in this class it will have the colour property *colour2*. Each of these rules has an associated probability.

When the user issues a query, the concept is found whose rules most are respected by the image. This is effectively a classification operation. The set of images associated with that concept can then be retrieved, or more common methods of image retrieval used to retrieve specific images from that set.

The system offers high performance and effectiveness. However it does have some drawbacks:

- Images are assumed to have associated text.
- It is assumed that during the initial construction of the image database, the media processing techniques will successfully extract regions of images corresponding to real world objects. This will only work in limited application domains.
- The clusters of images found during the initial construction are assumed to each represent a discrete and useful real-world concept. This might not necessarily be the case.

3.5.13 Active Library on Corrosion

The Active Library on Corrosion is an hypermedia application developed by Arents *et al.* (Arents & Bogaerts, 1993). It uses semantic information to provide two novel methods of navigating around information. Only text information is 'understood'; other media are opaque to the system.

The first navigation method involves extracting index terms from nodes in the hypermedia network. These are related using three thesauri related to corrosion. Each hypermedia node has at least one associated index term in each thesaurus. The indexing is performed manually, required a large amount of time and effort. The thesauri are *semantically coupled*, that is, the combinations which are invalid or inappropriate are known and automatically excluded from the display.

The index space is then visualised as a *cube of contents*. Each dimension of the cube corresponds to a thesaurus. The user can 'zoom in' and 'zoom out' in each dimension by choosing broader and narrower terms in the relevant thesaurus. When the user selects a term on the cube, a *contents plane* is displayed. Documents pertaining to the term appear on the plane as white dots. The user can then select terms in other dimensions, creating more planes, and the documents shown as white dots are constrained to those lying on the intersection of the planes. Thus the number of displayed documents can be quickly narrowed down using this visualisation of the index space. At any stage, the documents shown as white dots can be listed and accessed by the user.

An obvious restriction of this approach is that it requires that every document is indexed using three thesauri.

The second navigation method involves constructing a *semantic hyperindex*, using the same thesauri. The index space consists 'descriptors' that describe particular domain concepts (terms) and how they relate. Examples of index entries are:

```
(DEFINITION dry.corrosion)
(EFFECT (dry.corrosion (lifetime valve.part)))
```

The user can navigate by choosing a template, which describes the particular relationship being explored. For example there are DEFINITION and EFFECT templates corresponding to the two examples given above. The user can then expand and refine each concept in the entry, and the documents appropriate for that entry are offered to the user. Again, the thesauri are semantically coupled, so invalid combinations will not be offered.

Both methods have been found useful by users during trials. Arents *et al.* acknowledge that the main limiting factor of both styles is that they require complete and accurate indexing of the hypermedia information. Of course, they also rely on the availability of relevant thesauri and textual information.

3.5.14 COSMOS

The Content Oriented Semantic Modelling Overlay Scheme (Agius & Angelides, 1999) is a content modelling scheme for multimedia information systems. Multimedia objects may be described in a high degree of detail. It allows the description of several aspects of semantic information:

Explicit media structures, for example, a particular clip of audio;

Presence of real-world objects in the media;

Spatial relationships between objects;

Events and actions involving objects;

Temporal relationships between events and actions;

Integration of syntactic and semantic information, which includes the ability to identify what semantic object a piece of media corresponds to.

The system provides a description language and a query language for describing and querying the information held in the system, and allows very flexible access to it. However, all information must be manually inserted. This is likely to require a huge amount of time and effort, even with a relatively small collection.

3.5.15 Colombo's Semantic Visual Information Retrieval

Colombo *et al.* have described how they automatically extracted two levels of semantic information automatically from a set of art images, and a set of video advertisements (Colombo *et al.*, 1999). The two semantic levels are:

Expressive. This level holds the structure of the images, or how the various low-level features relate, according to a set of rules. For example, there are various rules for deciding whether or not two colours are *harmonious*.

Emotional. This is a higher, more abstract level that the expressive level. Rules about how (for example) line slopes cause different emotions are used to determine the 'emotional content' of the images. For instance, a horizontal line communicates calmness and relaxation, while an oblique slope communicates dynamism.

These two levels can be used to allow the user to give a very expressive query. The approach also boasts the apparent advantage of indexing and determining semantics automatically. Colombo *et al.* acknowledge that the way in which semantics are determined are very domain-specific. Additionally, the construction of algorithms for determining these semantics is a very complex and involved task. Unless there is a huge set of images or videos in a collection, or there are readily available algorithms for determining semantics, the advantages of this approach as opposed to manual assigning of semantic information may be lost.

3.6 Summary

The use of semantic knowledge in information systems is not new. Thesauri have existed for over a century that have mapped knowledge of a subject area, and been used to aid access to that information. More recently, the artificial intelligence arena of research has provided richer methods of storing semantic information in computers.

Only a handful of systems use semantic information to any significant degree in the multimedia information system world. The majority of those that do can only work with textual information. Those that work with multimedia information are mostly limited to specific domains and media. The main exception, Hirata's Himotoki/COIR system, makes unreasonable assumptions about the quality of media processing technology.

The next chapter describes an approach to alleviating these drawbacks called the *Semantic Layer*.

Chapter 4

A Semantic Layer

4.1 Problem Summary

It can be seen from the previous chapters that the problem of multimedia information access has been addressed in recent research, but that many problems still exist.

1. Largely, semantic information has to be assigned to multimedia information by hand in the form of keywords. Thus a large multimedia collection with no associated keywords is difficult to access at a higher level than low-level media features. Assigning keywords involves a huge, often prohibitive amount of effort.
2. There are some situations that current media processing techniques cannot cope with. For example, an image showing a side view of a horse and another showing a front view will be significantly different in terms of low-level features. Thus, a content based retrieval query will not retrieve all relevant results.
3. Generally, open hypermedia systems involving multimedia content can only link between non-text media in a specific, point-to-point way. One exception to this is of course the first MAVIS project. Even in this case, a generic link can only be followed from a similar instance of the object; the link cannot be followed from a different view of the same object, since their equivalence is not picked up using only low-level features. Another example is Hirata's work at NEC (Hirata *et al.*, 1996; Hirata *et al.*, 1997). This work seems to expect rather too much of current media processing technology.

4. In most multimedia and hypermedia systems, there are no mechanisms for expressing semantic relationships between objects. Those relationships may be extremely useful for the user.

In text retrieval systems, one of the main problems encountered is that of *terminology*, or *vocabulary*; that is, the terminology an author has used to describe a subject may be different from that a searcher uses to formulate a query. Thesaurus systems address this by grouping together synonymous terms or phrases describing a particular subject or *concept*. The matching power of this set of terms is greater than that of the terms alone.

This problem of terminology is rather similar to problem 2 described above. The two pictures of the horse are two different ways of expressing the concept 'horse'. There is a disparity between the visual terminology used to represent the concept.

The roots of this problem lie with the distinction Smoliar *et al.* identify between the symbolic object used to represent a concept (in the *plane of expression*) and the concept being represented (in the *plane of content*) (Smoliar *et al.*, 1996; Smoliar & Wilcox, 1997). de Saussure termed these symbolic objects *signifiers* (de Saussure, 1986). Gupta *et al.* also identify this distinction, terming the two components the *appearance value* and *semantic value* respectively (Gupta *et al.*, 1997).

A system attempting to address the problems listed above should account for this distinction, commonly known as the 'semantic gap'. However, as can be seen by the last two chapters, few systems do, especially outside of the domain of text. The focus of this work is how this gap may be bridged. The remainder of this chapter introduces a technique for addressing these problems called the *semantic layer*.

Sections 4.2 and 4.3 describe the overall framework of a system with a semantic layer, and the additional possibilities for navigation and retrieval that a semantic layer offer. Details about how this can be achieved in practical terms are given in the remaining sections.

4.2 Adding a Semantic Layer

Section 2.2.3 introduced the idea that multimedia objects can be considered to have two parts, one in the plane of expression, and one in the plane of content. Accordingly, it seems sensible that a multimedia information system also hold these two parts.

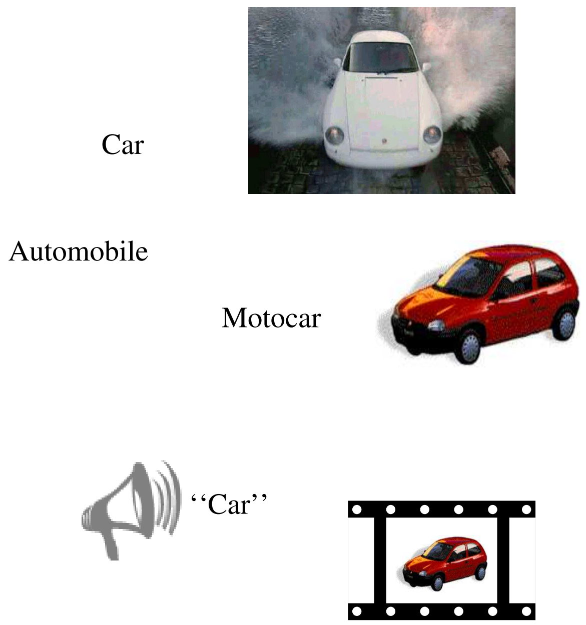


Figure 4.1: Representations of a Single Real-world Object in Different Media

A single concept can be represented in the plane of expression in many different ways. It may have several textual representations, such as ‘automobile’ and ‘car’, and visual representations, such as photographs of cars or technical drawings, audio representations, such as the sound of a car engine, or an audio clip of someone saying the word ‘car’, or video representations, such as a film of a car driving along a road. This is illustrated in figure 4.1.

Everything in figure 4.1 represents a single real-world object in the plane of content. This can be represented as an abstract entity called a *conceptual representation* inside a computer, and connected to each of the media representations, as in figure 4.2.

If there is one of these conceptual representations in the plane of content for each concept in a subject domain, and each of the media representations are connected to the relevant conceptual representation, then the power of a multimedia information is significantly improved. The following becomes possible:

- The ‘what is this?’ query
- Viewing alternative representations
- Expanding queries with alternative representations
- Document indexing
- Link augmentation

These techniques are described in the following sections.

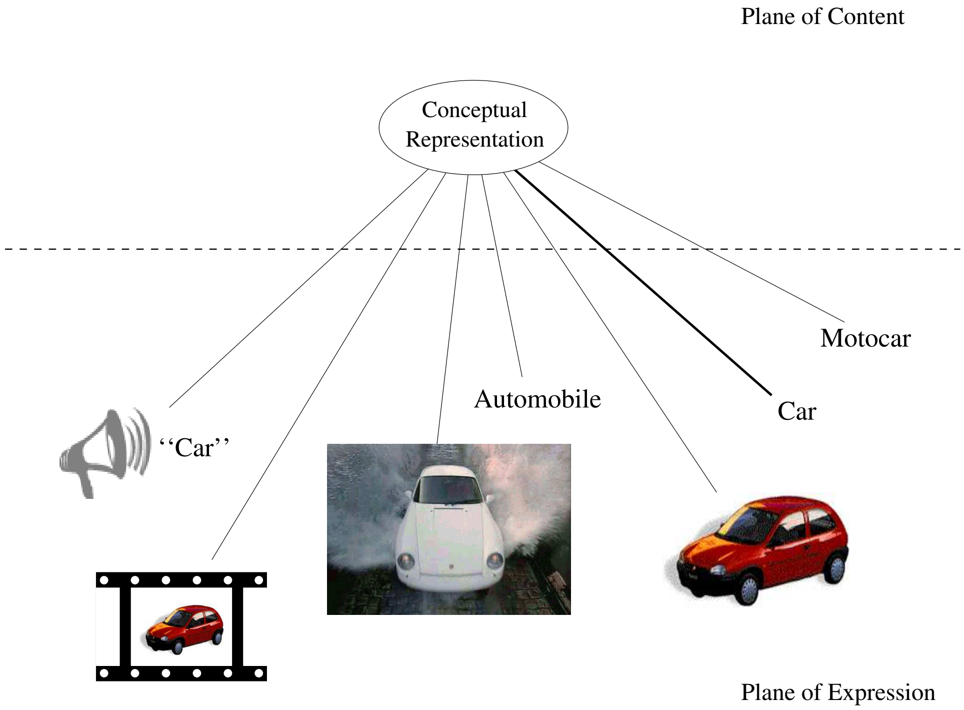


Figure 4.2: Media Representations in the Plane of Expression Connected to an object in the Plane of Content

4.2.1 What is This?

A user may at some point arrive at a media object, be it textual, graphical or another medium, that they are unable to identify. At this point they can ask a system with a semantic layer, "what is this?" The system will identify which concept or concepts that media object pertains to. For example, the user may select a portion of an audio clip with the call of an animal in it. The system identifies the concept 'Osprey' since it has a similar audio clip of an osprey's call connected to it.

4.2.2 Viewing Alternative Representations

When viewing a multimedia object, a user can ask the system what other representations of that object it has. For example, if the user comes across a word they are unfamiliar with, such as *asphodelus*, they may ask for a pictorial representation of the word. The user would then be presented with a photograph of a daffodil, and understand what the word *asphodelus* means.

This mechanism allows semantically based navigation across media types. The user is in the same 'semantic location', but in a different medium.

4.2.3 Query Expansion

A multimedia information retrieval system could use alternative representations of a query object to expand a user's query. If the user intiates content based retrieval

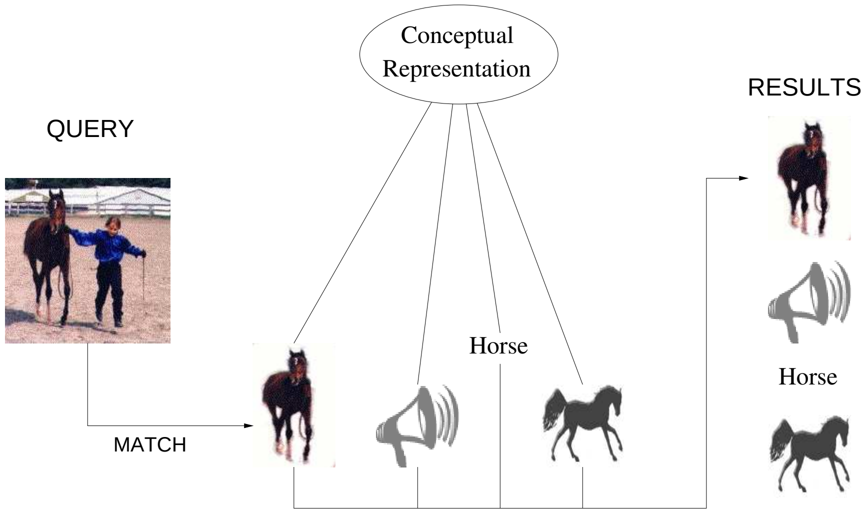


Figure 4.3: Simple Query Expansion

using a front-facing view of a horse as a query, another view of the concept 'horse' depicting a side view may be retrieved, and used to retrieve other side views of horses.

In a similar manner, representations of the concept horse in other media may be retrieved if they are also connected to the concept.

There are two levels at which this query expansion can operate. Which is most appropriate depends to a large extent on how comprehensively a multimedia collection is connected to concepts.

Simple expansion retrieves alternative representations of a concept that have been explicity connected to that concept. This is illustrated in figure 4.3.
This is likely to be effective when a multimedia collection has been comprehensively connected to concepts, since every object pertaining to that concept would be retrieved.

Further expansion uses alternative representations to perform regular content based retrieval queries in parallel. Figure 4.4 illustrates this. This is likely to be appropriate in situations where comprehensively connecting objects is impractical, for example the World-Wide Web.
An additional consideration in this case is how to present results. In certain cases it may be uncertain why a particular media object has been retrieved. If the number of representations of a concept is large, the number of objects retrieved in parallel queries will also be large. It is important that the user is not overwhelmed by the amount of information presented, and that they know *why* that information has been presented. One way this can be achieved

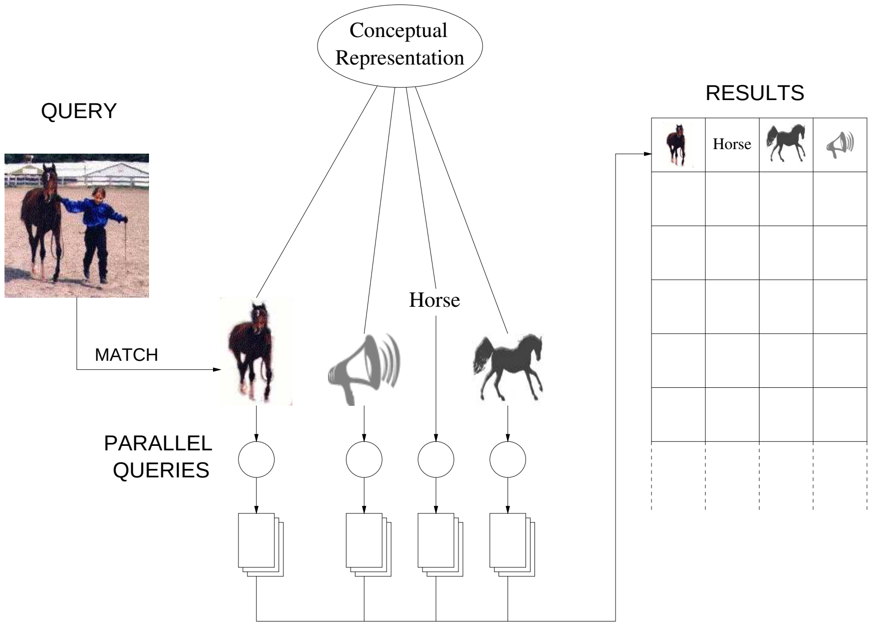


Figure 4.4: Further Query Expansion

is to show results of parallel queries separately. The user could click on a representation on the screen to see the results of a particular query.

4.2.4 Document Indexing

In a similar manner to the preferred terms in a thesaurus, the concepts in a semantic layer can be used to index multimedia objects or documents. The user can issue a query in any media, and the relevant concept or concepts can be found. Documents indexed on those concepts can then be offered to the user.

Depending on the availability of relevant media processing techniques, this indexing could occur automatically. For example, as images are introduced to the system, deformable template matching (Jain *et al.*, 1998) could be used to find out if any region of the image matches strongly with a currently held representation of a concept in the semantic layer. The image could then be indexed by that concept. However, this process may only be practical in restricted application areas, since media processing methods are not always robust enough.

4.2.5 Link Augmentation

An hypermedia system's functionality can also be enhanced with the introduction of a semantic layer. Source anchors of hypermedia links can be connected to relevant concepts. Links can be authored in one of two ways.

- Concepts themselves can be considered as end points of links. Such links can then be followed from any of the representations of that concept.

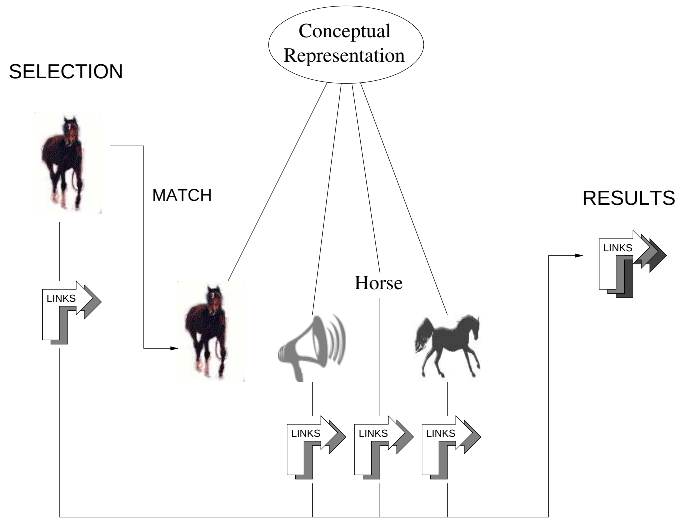


Figure 4.5: Augmented Link Computation

- The semantic layer can be used to augment existing generic hypermedia links, and other hypermedia links authored by conventional means. If a user wishes to view any links available on a selected media object, the concept in the semantic layer corresponding to that selection is located. Any links authored on other representations of that concept can be offered to the user as well as any authored on the selection itself. This is illustrated in figure 4.5.

4.3 Further Knowledge

Adding a layer of concepts to any multimedia information greatly increases the flexibility of that system. While these concepts are useful aids, there is no *knowledge* about those concepts held in the system, other than "these media items represent the same real-world concept".

These concepts are abstract entities, and can be manipulated in any way necessary. Knowledge representation techniques can be employed to represent real-world knowledge about the relationships between these concepts, and this can be used to further enhance the ways in which information can be accessed by a user.

Section 3.3 described a number of ways in which knowledge can be represented in a computer system. The semantic layer is equivalent to a semantic network, and thus related techniques can be directly applied to the layer of concepts. Any set of relations can be used to describe relationships between concepts in the semantic layer. This knowledge is implicitly transferred to the multimedia representations of those concepts.

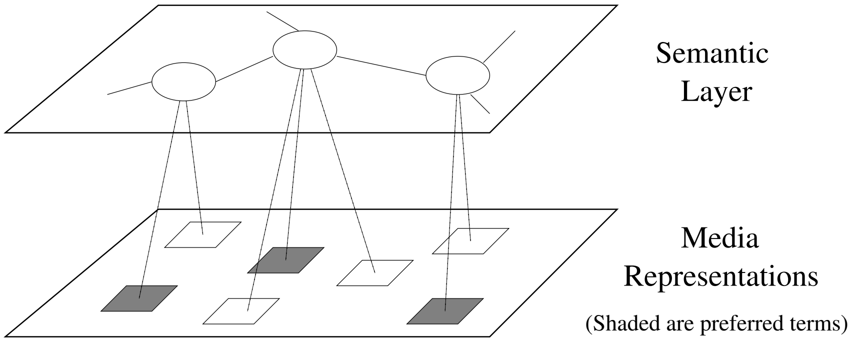


Figure 4.6: A Semantic Layer over a Multimedia Collection

Thus, given an existing multimedia collection, that may be hyperlinked, it is possible to add a *semantic layer* of connected concepts to it. This is illustrated in figure 4.6.

All media associated with a concept should be treated as equal; text terms, images and sounds may all be used to reach a concept and to represent it. When presenting a concept to a user, one particular representation should be chosen as the preferred form for displaying on-screen. This should be a text term. One good reason for this is that text is simple to display, and does not take up much screen space. Another reason is that it is usually the easiest medium with which to ensure that the concept is represented unambiguously. With a sound or image it may be unclear exactly what aspect or part of the medium is representative of the concept. However, since this preferred form may also be used to reach the concept during queries, and is representative of it, it should reside in the media layer and not the concept layer. Thus, in figure 4.6, one media representation of each concept has been highlighted as the *preferred representation.*

The knowledge in a connected network of concepts provides further possiblities for information access. These are described in the following sections.

4.3.1 Extended Query Expansion

The functionality of the query expansion method described in section 4.2.3 above may be extended by using representations of semantically related concepts in the semantic layer. As well as retrieving different representations of the relevant concepts, representations of related concepts can also be retrieved, or used to initiate further queries. The extent of this 'fan-out' will greatly affect the number of documents returned to the user, therefore some way of filtering or limiting this must be found. Possible methods include:

- Setting a threshold on the match strength or number of objects retrieved in parallel queries;
- Limiting the number of 'jumps' traversed;
- Restricting the type of relations traversed. In particular, using the representations of specialisations ('narrower terms') of the initial concept may provide good results since the scope of the initial concept encompasses these specialisations.

4.3.2 Concept Navigation

The techniques described so far have been concerned with navigating around the media representations. The semantic relationships between concepts can be exploited by the user directly by traversing them. Rather than navigating around media, the user navigates at a higher level between the concepts to which the media pertain.

This is likely to be useful since the semantic layer is effectively a semantic map of the multimedia information in an application. McDonald *et al.* have shown that a map of a hypertext improved users' ability to use the hypertext to answer questions, to such an extent that the performance of users not knowledgeable in the subject field matched the performance of those that were (MacDonald & Stevenson, 1998).

The user starts at a media object, and the concept pertaining to that is assertained by the system. The user can then 'beam up' to the conceptual layer, and navigate around it by following the relationships between concepts. Once they have found a concept of interest, they can then 'beam down' to the media level to view representations of that concept, and any documents indexed on it. This is similar to browsing the subject index before viewing documents.

How the network itself should be presented and made navigable by the user is a key area of this research. Hypermedia and thesaurus navigation tools typically allow the user to view a single term or 'focus of attention', and to move this focus of attention by selecting a new term (or document) somehow related to the original. In the semantic layer, concepts will be represented by their preferred (text) representation, as described in section 4.3.

The hierarchical structure of many existing classification systems and thesauri lends itself to a particular style of navigation. The user can specialise or generalise by moving up or down the *broader/narrower* term tree, or move 'across' by moving to a term related in some other way. These techniques can be directly applied to the semantic layer if it has an hierarchical structure.

4.3.3 Narrowing the Scope of a Query

If a semantic layer has both a comprehensive set of representations and indexes all necessary relevant documents to a task, it can be effectively used to narrow the scope of a search, reducing the amount of irrelevant media retrieved and speeding up the search process.

The user is allowed to select some portion of the semantic layer. Once this portion is identified, only representations and relevant documents associated with the selected concepts are used in search and navigation operations. For example, if the user is only interested in cars, they can specify that only the concept 'car' and narrower (more specialised) concepts are considered. The user can then submit a query consisting of just an abstract property such as the colour red, only pictures of red cars will be retrieved. To retrieve red cars using another method, the user would need to include a representation of a car in the query, and even then only similarly-shaped views of cars would be retrieved in a single step.

A key issue here is how the user selects a portion of the semantic layer and how quickly they can do it; if it takes a long time there may be no point in narrowing the scope at all. Two possible methods are:

- Select a concept using a concept navigation tool. That concept and all of its narrower concepts (and all of their narrower concepts) constitute the selected portion.
- Select a concept and its surroundings (defined by a semantic distance measure such as number of relations traversed.)

Additionally, this technique relies on a multimedia collection being comprehensively connected to concepts. If only a fraction of actual media representations are connected, then only this portion will be retrieved, and the unconnected representations will be rejected.

The techniques described above may be combined for further effectiveness. For example, the scope narrowing technique may be used to enhance the accuracy of a 'what is this?' query.

4.4 Interactivity

It can be seen from the descriptions of these techniques that some may require additional interaction, while some may occur transparently or "behind the scenes". In many cases, interaction may be appropriate; however, this takes time, and one of the main aims of a multimedia information system is to minimise the amount of time and cognitive effort required to find a piece of information. Possible methods for determining the best balance between automation and accuracy are discussed below.

4.4.1 Query Expansion

For query expansion to be effective, the expansion must start at an appropriate concept. If there is uncertainty about this, the system should ask the user, as an incorrect decision will result in a lot of wasted computation and further interaction. Additionally, the extent of the expansion could be altered by the user. If necessary, sensible defaults for the particular multimedia collection could be overridden by the user on an options dialogue or menu, only requiring additional interaction when the user encounters some circumstance that requires further or narrower expansion.

Should further extension be necessary, the user could use a "slider" indicating how many "jumps" to make. Additionally, the number of representations this will retrieve (or use to start further queries) could be displayed at relatively little computational expense, and this would give the user a good idea of when they have expanded the query too far.

4.4.2 Link Augmentation

Link augmentation is actually very similar to query expansion, applying exactly the same techniques to the source anchors of links. A default, which could be specified by the author, can be overriden with a menu item or option dialogue if necessary. Dragging a slider to alter the extent of 'fan-out' would again be useful, and the number of links returned by that extent can be updated synchronously.

4.4.3 Scope Narrowing

The basic idea of scope narrowing is to increase the chance that returned media objects are relevant to a query. The chances of irrelevant media objects being returned is reduced because many irrelevant media objects are not considered. Representations of a large number of concepts can be excluded.

Whether or not this allows a user to find the information they require more quickly will depend on how long it takes them to identify the relevant portion of the semantic layer. Anything very complex, such as selecting concepts individually, is likely to take too long.

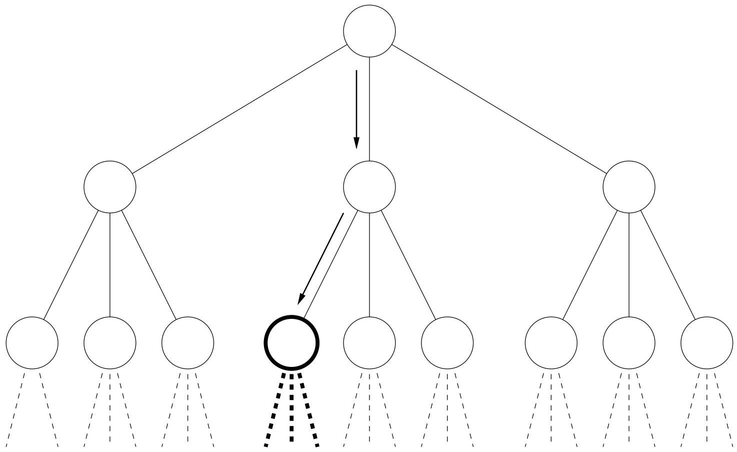


Figure 4.7: Narrowing the Scope of a Query With a Small Number of Navigational Steps

Many classification systems and thesauri feature specialisation/generalisation relationships. The backbone of such a system is hierarchical. Starting at the root, a small number of steps can effectively exclude a large percentage of such a concept network. This is illustrated in figure 4.7. Two specialisation connections have been traversed, indicated with arrows. Only media representations connected to the highlighted concept and its specialisations will be considered in retrieval (or link following) operations. Assuming the hierarchy forms a balanced tree, around 89% of the concepts in the network, and associated media objects, have been excluded in two navigation steps.

If the user knows that the information they require is likely to be a specialisation of a certain concept (such as ‘transport’) these two steps may greatly increase the effectiveness of a low-level query.

An hierarchical display on screen can be used to indicate the user’s area of interest. This could be kept on-screen alongside views of media objects, and the precision of retrieval operations is increased greatly with relatively little cognitive effort.

How effectively the technique would perform transparently would depend on the quality of automatic media object classification as described in section 4.6.2.

4.5 A Multimedia Thesaurus

As was noted in section 3.3, a thesaurus can be visualised and represented as a semantic network. In this way, thesauri can be viewed as useful, comprehensive, existing semantic networks.

Despite the relatively small number of relationships, the information contained in a thesaurus has been proven to be useful during a searching task (Beaulieu, 1997). Additionally, there are a large number of existing text thesauri covering a wide range of subjects, that have been researched and refined for decades, and contain comprehensive information about the textual terminology and relationships between concepts in those domains.

An existing thesaurus, then, may be a useful starting point in the construction of a semantic network layer to supplement a multimedia collection.

The existing sets of equivalent terms (including the preferred forms) in such a thesaurus can be separated from the underlying concepts they represent. The thesaurus is separated into two parts:

1. A layer of concepts connected by the relationships "broader/narrower" and "related", corresponding to the "broader term," "narrower term" and "related term" relationships described in section 3.2.
2. A set of terms connected to these concepts. These are the "equivalent terms" of each preferred term in the thesaurus (see section 3.2).

Thus the thesaurus can be fitted to the semantic layer architecture described in the previous section.

Equivalent terms in a controlled language text thesauri are called 'lead-ins' since they lead in to the preferred form of the term. This preferred form is used to specify relationships between concepts, and to index documents. In the multimedia thesaurus, the abstract concept in the concept layer can take over this role, and the preferred form becomes another lead-in.

Multimedia representations of concepts can be added as lead-in 'terms' in the media layer. The result of this can be called a 'multimedia thesaurus'; an existing text thesaurus has been supplemented with multimedia data, and can be used to augment the information and navigational capabilities available in an information system. An initial embryonic idea for a multimedia thesaurus was proposed by Lewis *et al.* (Lewis *et al.*, 1996b).

4.6 Practicalities

The previous sections have described the idea of a semantic layer, and the ways in which it could be used to aid information access. However, there are practical details that need attention:

- Given a piece of media, how does the system work out what the corresponding concept or concepts are?
- Given a multimedia collection, how can the semantic layer be connected to that collection without a huge authoring effort?

The first of these is rather simple if a specific media to concept association already exists. However, if the system is used in a wider context (for example on the World-Wide Web), or is to be exposed to any information that has not been comprehensively connected to the semantic layer, it is a complex problem. This is the well-known problem of classification. Ways in which this can addressed are described in section 4.6.2.

The second problem presents a rather more open question: How can the semantic layer be constructed, and how can media representations be connected to concepts without an author connecting each manually? Approaches to this are described in section 4.6.3.

It is first necessary to describe the information system environment and media processing that would be required for a semantic layer to be realised. The characteristics of such a system are given in the next section, and some assumptions listed.

4.6.1 The Surrounding System

A semantic layer in itself is not of much practical use without suitable media processing techniques. A semantic layer or multimedia thesaurus is most gainfully employed *assisting* a multimedia information retrieval and/or navigation system. In order to describe how a semantic layer will work, it is necessary to make some assumptions about that system.

From the investigation of existing information retrieval and media processing techniques in chapter 2, we can make the following generalisation:

> Any content-based retrieval operation can, given a query media object, produce a ranked set of results. These results will be in the same medium as the query object.

Thus, we can treat any content-based retrieval operation, regardless of the technique employed and medium, as a black box. This is illustrated in figure 4.8.

In addition, it is necessary to make some assumptions about how these operations can be performed in order for a semantic layer to work:

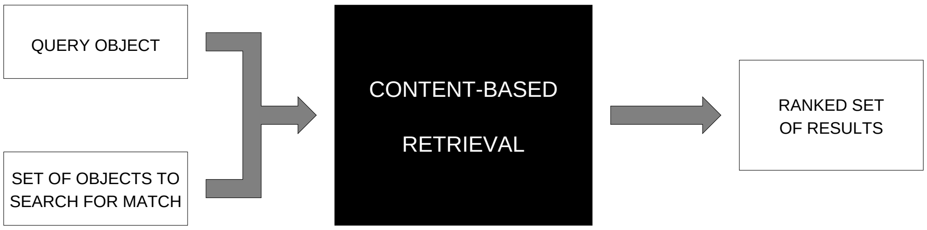


Figure 4.8: The Content-Based Retrieval Black Box

1. Different media features (such as colour histograms or textures for images, or fourier transforms for sound) may be used independently for queries.
2. Specific portions of a media object may be indexed and retrieved. This is required since real-world objects may constitute only a part of a media object, and it is only appropriate to connect that part to a concept.
3. Some techniques use an extracted intermediate representation that is not always reproducible automatically. For example, the edge map depicting the shape of an object may have been produced interactively, and thus cannot be reproduced from the original image alone. In this case, it must be possible to store this extracted representation somewhere, and to refer to it in the media layer. The reason for this is that the edge map could describe the shape of an object with a corresponding concept in the semantic layer. This edgemap will be required during a query expansion operation for instance.
4. The retrieval process can, if required, operate on only a specified subset of media objects in the system.
5. A way of combining the evidence (ranked results) of more than one retrieval operation exists, to provide an overall ranking across an arbitrary number of media features.
6. Queries can be started within the system, and the results 'swallowed' and utilised by the system, without the user being explicitly involved.

4.6.2 Classification

A process central to operations involving a semantic layer is that of finding the concept or concepts pertinent to a media object. This is no simple task, but there is a wealth of research that can be drawn upon to assist in this.

At present, the only clues a computer has as to what an object *is*, that is, what a piece of media *represents*, is that of the low-level features extracted from the digital data constituting that piece of media. Chapter 2 illustrated to some extent the number of available methods for extracting features.

So, as noted by van der Heijden *et al.* (van der Heijden & Worring, 1996), a way of mapping low-level features of media objects to the high-level concepts is required. This is not simple. Firstly, given a non-textual media object, be it an audio clip or a piece of text, it is unclear what aspect of that object represents the object. Oka is among many to note that much of the raw data is uncertainty and noise (Oka, 1998). It is likely that there is a particular aspect or aspects that can best be used to identify the meaning of the object, for example shape or texture.

It is likely that one or more of the low-level features that can be extracted from a particular medium may be better at conveying this meaning and for identifying further media objects pertaining to the same concept. Accordingly, for all media representations of a concept, which features these are should be known by the system. When these features are not known, all available features should be used, since to guess might exclude an important feature. Additionally some learning-based technique may be able to determine which features are most suitable.

Some of these features may require (or work best with) interactively generated intermediate representations, such as edgemaps. These should also be held by the system.

Figure 4.9 shows this mapping from high-level concepts to low-level features. These low-level features associated with concepts are termed *lead-in features*. This is slightly different to van der Heijden's mapping, shown in figure 3.8. However, it is more indicative of the route to which the concept is actually reached. During a query, it is the low-level features that are directly used for calculations, and it is by following the trail shown in figure 4.9 that a relevant concept is found.

Now, a query can be broken down in a similar way and compared to these features. There are a number of ways in which, given some low-level feature values extracted from a query, the most appropriate concept can be determined. These are described in the following three sections.

Closest Match

The closest-matching lead-in feature is located. The concept corresponding to this feature is considered the concept relevant to the query. This is computationally inexpensive.

Clustering

The lead-in features for an individual concept are treated as a *cluster* (see section 2.2.2). This way, each concept has one presence in the 'feature space'. Given a

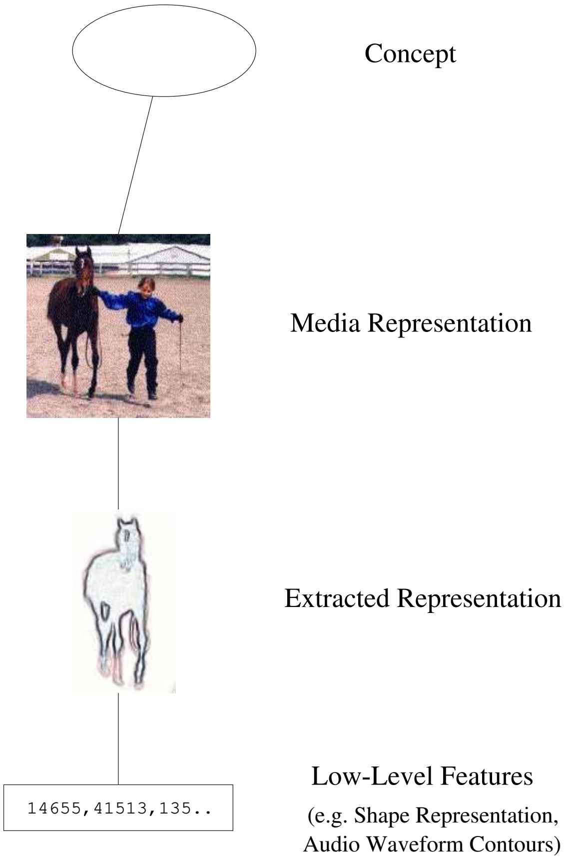


Figure 4.9: Route from Concepts to Low-Level Features

query, the closest cluster can be found, and the concept close to this cluster is considered relevant to the query. This technique makes two assumptions:

1. Media representations of a concept will appear close together in the 'feature space'. That is, for a given feature, all representations of a concept will have similar values. This often is not the case. In the horse example, the shape of a horse differs significantly depending on the viewing angle. There may be another concept 'dog'. The side view of a horse is likely to be closer to the side view of a dog in the feature space than it is to the front view of a horse. Thus, the assumption does not hold; the feature vectors of each concept are unevenly distributed throughout the feature space.
2. There is a similar, significant number of examples of media representations of each concept. In order for a clustering technique to be effective, there must be sufficient numbers of feature vectors for each concept to be able to determine boundaries between concepts. This might not always be the case, though as a system is introduced to more media representations, there may be enough

features to decide whether clustering features together is appropriate. Some monitoring agent process could perform this.

The clustering technique may not therefore be appropriate in all cases. However, it is a technique that has proven effective in many areas and should be used if the two assumptions are met. There are related techniques, for example neural networks, that could train on clustered and classified data, and that trained network may then be effective at identifying instances of concepts.

User Interaction

The low-level features of the query can be compared with lead-in features to produce a ranked list. The concepts associated with the best-matching lead-in features (for example, the top five) can then be presented to the user. The user can then select the most relevant concept. If appropriate, the best-matching concepts can be determined using the clustering technique described above.

This technique makes use of both the low-level features and the user's knowledge. Given that the user verifies the system's estimation of the best concept, this technique is likely to produce the best results. It is possible that in some circumstances this additional interaction is too costly time-wise; however this decision can be made by the software at the time.

4.6.3 Construction

Although the problem of identifying concepts relevant to a query has been addressed, there is still the problem of how media representations are connected to the semantic layer when no examples already exist. The process of constructing a semantic layer system can be divided into three steps.

Construction of the concept network

Before indexing documents or connecting the semantic layer to media, a concept network must be in place. Updating of the concept network throughout the life of the semantic layer is possible, however care must be taken: Lead-ins and documents may require re-classification. The concept(s) to which they are most appropriately associated may change. Thus, the initial classification should be prepared carefully (Aitchison & Gilchrist, 1987).

It may be the case that a thesaurus or classification system for a particular application already exists, for example the *Arts and Architecture Thesaurus* (The J. Paul

Getty Trust, 2000) classification. This can be used as the basis for a multimedia thesaurus. Any text terms can be connected to concepts as media representations, and thus the system already has the searching power of that thesaurus or classification system.

Some rather more generalised classifications already exist. For example the *Dewey Decimal Classification* system has a very wide scope. It is likely that a selected subsection of such a classification may be useful, since the scope of a multimedia system may be smaller due to the specificity of media matching and processing techniques. This does rely on there being enough concepts to adequately represent the domain covered by the multimedia application.

In the absence of a readily-available classification system suitable for adapting as a semantic layer, the network of concepts can be designed by domain experts. There is no restriction on the number or types of relationship between concepts, though these should include a specialisation/generalisation relationship as are required to make use of some of the techniques described in sections 4.2 and 4.3.

Initial connecting of media representations

How to connect representations (and hence low-level features) to concepts is a more complex problem. Associating media representations with concepts is effectively introducing semantic knowledge to the system, a traditionally difficult and time-consuming procedure. How effective a system with a semantic layer is in practical terms is likely to depend to a large extent on the effort required for this step.

There are two main approaches to this:

1. In some cases, it may be possible for a human to allocate classes for an exemplar portion of the multimedia collection. A tool with a suitable interface would facilitate this. Once these are in place, the system can classify documents or objects automatically using the techniques described in section 4.6.2. The system can fall back to user interaction in those cases where there is a large degree of uncertainty as to which is the most appropriate concept. For example, an audio clip may be significantly different from any other clips in the system, so the system cannot confidently assign that clip to any particular concept.
2. Each concept is already likely to have a textual representation from the initial construction of the concept network. Some set of other media items may have

associated text. For example, there may be keywords associated with images, or captions associated with a video sequence. This text can be searched for text terms associated with concepts using the wealth of techniques available for this purpose (see section 2.2.2), and thus connected to appropriate concepts.

A drawback to this technique is that it connects media representations to concepts without specifying which low-level features are representative of the concept. This can be addressed if, for a set of media objects, there is a common feature that is appropriate for this purpose. For example, when adding medical images to the system, a technique that locates artefacts and extracts shape information will be the most appropriate for a large portion of images.

It may be the case that some percentage of the collection to be introduced into the system has associated text metadata. This technique can be used to connect media with metadata, and the remainder classified semi-automatically, as described in point 1 above.

Indexing Documents

Optionally, multimedia documents can be indexed with concepts. Text documents can be indexed using one of the well-established indexing techniques. Some of these are described in section 2.2.2.

For multimedia data, it is likely that an automated or semi-automated process will provide good results, provided each concept has adequate representations. The collection can be searched for each representation of concepts in the semantic layer, and each document in which that representation appears (or is confidently considered to appear) is indexed on the appropriate concept.

At any point during concept navigation or querying, the documents indexed on a concept of interest may be offered to the user.

4.7 Summary

In this chapter, the technique of applying a semantic layer over multimedia data has been introduced. Some functionality required of the surrounding system has been described, and the following techniques explained:

- Viewing synonymous representations
- The 'what is this?' query

- Query expansion, with varying degrees of expansion
- Indexing documents
- Link augmentation
- Concept navigation
- Narrowing the scope of queries

Some of these can be implemented with varying degrees of interaction.

Concepts can be mapped to low-level features via the media representations. In different cases, different low-level features are best for specifying what it is about a media object that is representative of the connected concept, and this is also held by the system.

Given a query, the most relevant concept to that query can be found in one of a number of ways:

- Simply by using the concept connected to the closest matching low-level features
- Treating low-level features connected to concepts as clusters, and using one of the many available methods for finding the most appropriate cluster
- Presenting a list of 'best guesses' to the user, allowing them the final decision.

Initial construction of a semantic layer, and connection of this layer to a multimedia collection, can be broken down into three steps:

1. Construction of the concept network
2. Connection of media representations to concepts (which may be fully or partially automated)
3. Indexing of the multimedia collection.

In order to evaluate the techniques introduced in this chapter, it must be experimentally implemented on a computer system. The next chapter describes MAVIS 2, an experimental content-based retrieval and navigation system incorporating a semantic layer facility.

Chapter 5

The MAVIS 2 System

5.1 Introduction

In order to find out whether the semantic layer technique is viable, a system for implementing and testing one is needed. The work in this thesis was carried out as part of the MAVIS 2 project (Hall *et al.*, 1996; Dobie *et al.*, 1999a; Dobie *et al.*, 1999b). The project involved implementing a multimedia navigation and retrieval system with a semantic layer component. This chapter describes that system, and how the semantic layer component interacts with the rest of the system. This work was carried out as a group effort by MAVIS 2 team members.

The development platform used was Sun's Java Development Kit, developed on Linux and Sun Solaris machines. The system also runs and has been tested on Windows NT machines. Processes running on different machines and platforms may all interoperate to satisfy a single query.

5.2 Overview

MAVIS 2 is broadly termed a *Multimedia Information System*. It encapsulates both the information retrieval and hypermedia information discovery paradigms. Additionally, it features a semantic layer component called a *Multimedia Thesaurus* or *MMT*.

MAVIS 2 allows content-based retrieval in the form of 'query-by-example'. A query may be in any medium for which appropriate plug-in modules are available. This also includes free-text queries.

Content-based hypermedia navigation is also possible using Microcosm-style specific and generic links (Fountain *et al.*, 1990; Heath, 1992; Hall, 1994). These generic links can be authored on arbitrary media types in a similar manner to MAVIS 1

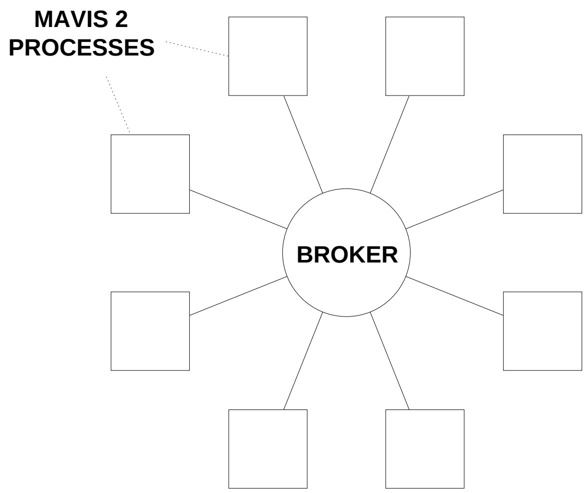


Figure 5.1: MAVIS 2 Topology

(Lewis *et al.*, 1996a; Lewis *et al.*, 1997b), as described in section 2.3.5. The linking information is held separately from the media itself, so the media does not need to be held locally, and write access is not required.

The multimedia thesaurus component allows for the higher-level, conceptual form of navigation and aids navigation and retrieval in the manners described in chapter 4.

The system is implemented as a set of interacting processes that communicate using a messaging system. These processes are implemented in Java, and can run on separate machines. Java was chosen as it promotes rapid development, and code runs on a large variety of platforms with little or no modification.

Communication between processes is via the HTTP protocol. The HTTP protocol is well-established, and is in widespread use in the World-Wide Web. Using this protocol, MAVIS 2 processes can communicate over the internet.

The messages themselves are in a text-based XML format. The use of two open standards for the messaging allows established and future developments to be employed. Further processes can be developed and run in any language and on any platform. The job of integrating existing tools with the system is relatively simple.

When they initialise, processes register their services with a central broker. When a query or other operation is initiated, messages are sent to the broker and forwarded to relevant processes. More than one process may respond to a message; for example, several processes may respond to a content-based retrieval query. The topology of MAVIS 2 processes is shown in figure 5.1.

Messages may be synchronous, in which case processes responding to a message send a reply back to the process issuing the message. Such messages normally require that a process evaluate something and send back the result. For example, a process might request a low-level feature (called a *signature* in MAVIS 2). Once a receiving process calculates this, it returns the result back to the issuing process.

Other messages are asynchronous messages; once they are sent the process will not wait for a reply but carry on as normal. For example, a *view* message is of this type. A process can send such a message requesting that a particular media object be displayed to the user, and will not wait for a reply.

The following sections describe the MAVIS 2 system in detail. The data architecture is given first, followed by descriptions of currently implemented processes.

5.3 MAVIS 2 Data Types

Any number of storage processes (called MAVIS *stores*) may be connected to a single broker. These contain the semantic concepts, the associations between concepts and media objects, and hypermedia links. Processes may use stores exclusively for their own purposes. For example, signature modules may use a store to store extracted signatures. One thing the stores are *not* used for is to store raw media data, such as bitmap images. The actual raw media data is left outside of the system, for example on a World-Wide Web server.

There is a core set of data structures that all processes must be aware of and will exist in every store. These structures are called MAVIS 2 objects. MAVIS 2 objects can be 'serialised' and transmitted as XML. A recipient process can parse the XML and reconstruct the object.

The core data structures describe actual media objects using *references.* Usually this is a Universal Resource Identifier, or *URI.* In practical terms, this allows MAVIS 2 to hold information about any media on the World-Wide Web or via FTP, and to take advantage of any further protocols that the URI mechanism will offer in the future.

The data structures also hold the semantic layer (or multimedia thesaurus), connections between concepts in the semantic layer and the media objects. The MAVIS 2 objects can be divided into four layers, as shown in figure 5.2. This model is quite close to Agosti's three-layer hypermedia/IR architecture (Agosti *et al.*, 1995; Agosti *et al.*, 1996).

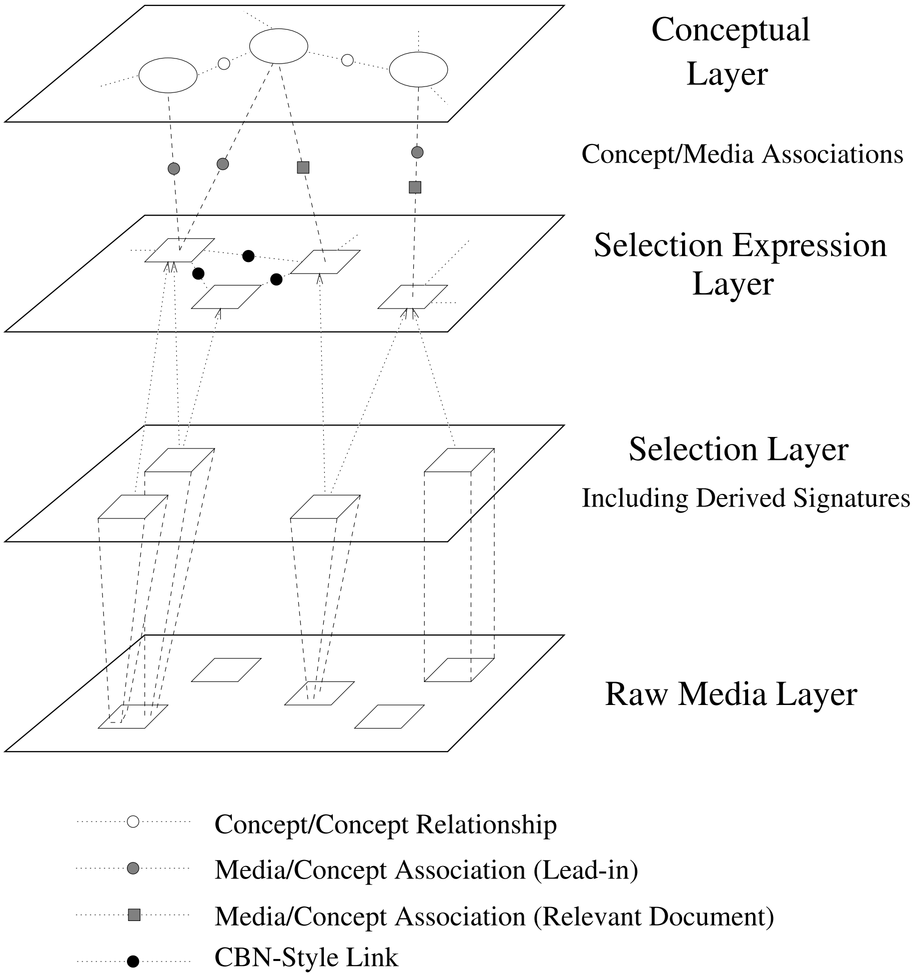


Figure 5.2: The MAVIS 2 Four Layer Data Model

The following sections describe each of these objects, and what data they hold, starting with the bottom layer of the diagram.

5.3.1 Raw Media Layer

The raw media layer contains references to the raw media objects themselves. These are called *RawMedia* objects. Usually this will consist of a URI, with an associated media type. The raw media data itself is not actually stored in the MAVIS 2 store, but conceptually this layer can be thought of as comprising the raw media data known to the MAVIS 2 system.

5.3.2 Selection Layer

The selection layer in composed of *Selection* objects. These consist of a reference to a piece of media from the raw media layer, together with the specification of a relevant part of it. This specification is called a 'shape', though it might be something as simple as *start of selection* and *end of selection* offsets, to a complex shape representation of an image segment. Thus, the selection may be a text selection from a text document, a region in an image, or a clip of an audio sample. Of course, a selection may specify a raw media object in its entirety.

Also specified are signatures (low-level features, as introduced in section 2.3.5) which should be used during retrieval or navigation operations. This is known as 'qualification'; selections with particular signatures specified as relevant are known as 'qualified' selections.

Usually, no actual media is held here. The exception to this case is the 'anonymous' selection. This is a selection that has not been derived from raw media. This may be a small text term that does not originate from a text document; in this way, text representations of concepts can be introduced without having to find an example in a text document. The selection may also be the specification of some abstract property, such as the colour red.

Any signatures that have been calculated may be held in this layer. Also, any extracted representation that may have required interaction to produce is held in this layer. For example, if the extraction of a shape from an image required that a user sets some parameters, then that extracted shape (and/or the parameters set by the user) are held here, so that it can be used to calculate further signatures.

5.3.3 Selection Expression Layer

In order to increase the flexibility of queries, the idea of a selection expression was introduced into MAVIS 2. Eventually, a *SelectionExpression* will consist of one or more selections, which may be 'like' (similar to) or 'not like' (not similar to) selections. This is primarily designed to be of use for formulating queries.

During matching, the system will treat a strong match with a 'like' selection as an indication of relevance, and a strong match with a 'not like' selection as an indication of non-relevance. This will allow a form of relevance feedback. If the results of a query do not suit the user, they can add additional selections to the query selection expression in the 'like' and 'not-like' columns, and re-run the query.

This mechanism has not yet been fully implemented. However, it was decided that this mechanism may be useful elsewhere in the system, for example to produce lead-ins to concepts that more accurately match representations of those concepts, so selection expressions were added as an extra layer in the MAVIS 2 data model. Everything in the system is specified as a selection expression, though currently they all consist of a single 'like' selection. Whenever a link is authored or a concept connected to a media representation, the endpoints are always selection expressions in the selection expression layer. Selection expressions are always used to manipulate and match media objects.

Hypermedia links are held in this layer. The endpoints of links are always selection expressions. Links may be specific or generic.

5.3.4 Semantic Layer

This is the implementation in MAVIS 2 of the semantic layer described in the previous chapter. It consists of *Concepts*, abstract entities representing real world objects in the system. These hold no information in themselves; representations of them are associated via the selection expression layer.

These *Associations* between concepts and selection expressions may indicate that the selection expression is a 'lead-in', that is, that it should be used for matching or otherwise finding concepts relevant to a query. The association may also indicate that the selection expression is a 'relevant document'; that is, the media specified by the selection expression pertains to the concept but is not representative of it and should not be used for matching. For example, a text document describing the operation of a car is relevant to the concept *car*, but would contain many "noise terms" that would confuse text matching signature modules. Thus, the document is not considered representative of the concept, and is not a 'lead-in'.

One of the 'lead-in' connections is flagged as the 'preferred lead-in'. Each concept must have one of these associations in order to be displayable to the user. The selection expression at the other end of this association should specify a small text word or phrase that unambiguously describes the concept.

Concepts are connected to each other by inter-concept *Connections*. These connections may be arbitrarily typed and weighted. It is up to the semantic layer process to enforce any constraints on this (such as structural constraints, or limiting the possible connection types.) The Multimedia Thesaurus implementation, for example, ensures that only *broader/narrower* and *related* connections are allowed.

The concepts combined with these connections form the semantic network that contains semantic knowledge about the real-world objects represented in the multimedia collection.

It is an interesting philosophical point that the concepts themselves hold no information, and are only given meaning by their connection to media and other

Address	Action	Input	Output
`http://eric:9120/Mavis`	MatchSignatures	L(image/*)	L(Distance)
`http://ernie:9240/Mavis`	MatchSignatures	L(audio/*)	L(Distance)
`http://ernie:9240/Mavis`	Classify	Selection	L(Result)
`http://client.com:900/Mavis`	View	text/html	-

Table 5.1: Part of a MAVIS 2 Broker Registry

concepts. This is known as the 'existence criteria' — for a concept to have a meaningful existence in the MMT, it must have media representations and/or related concepts.

5.4 MAVIS 2 Processes

In order to show the route of messages throughout a query process, the topology of the MAVIS 2 system is depicted differently. Firstly, since all messages go through the central broker, this is left out; messages are depicted as going straight from one process to one or more others. Secondly, *Store* processes are also omitted. Store processes contain information about media objects, thesaurus concepts, hypermedia links and so forth (although they do not contain the media objects themselves). Every other process in the system has to access the store, so to show this on the process diagram would complicate it unnecessarily.

Figure 5.3 shows the MAVIS 2 processes and how they interact using messages. It is possible to follow the arrows around the system from the query source (usually a viewer) to the *Results Viewer*. This is not offered as the only possible architecture for a system with a semantic layer component.

5.4.1 The MAVIS 2 Broker

Receives messages: Register, Deregister, Lookup. *Forwards others*

Sends messages: *Forwards messages*

All messages in MAVIS 2 are sent to a central brokering process. This broker farms the message out to appropriate processes.

The broker maintains a registry of running MAVIS 2 processes. When a MAVIS 2 process starts, it registers with the broker which message types it can receive. It can also specify input and output types. When a message is received, the broker then knows which processes are suitable for dealing with that message.

A section of a broker registry is shown in table 5.1.

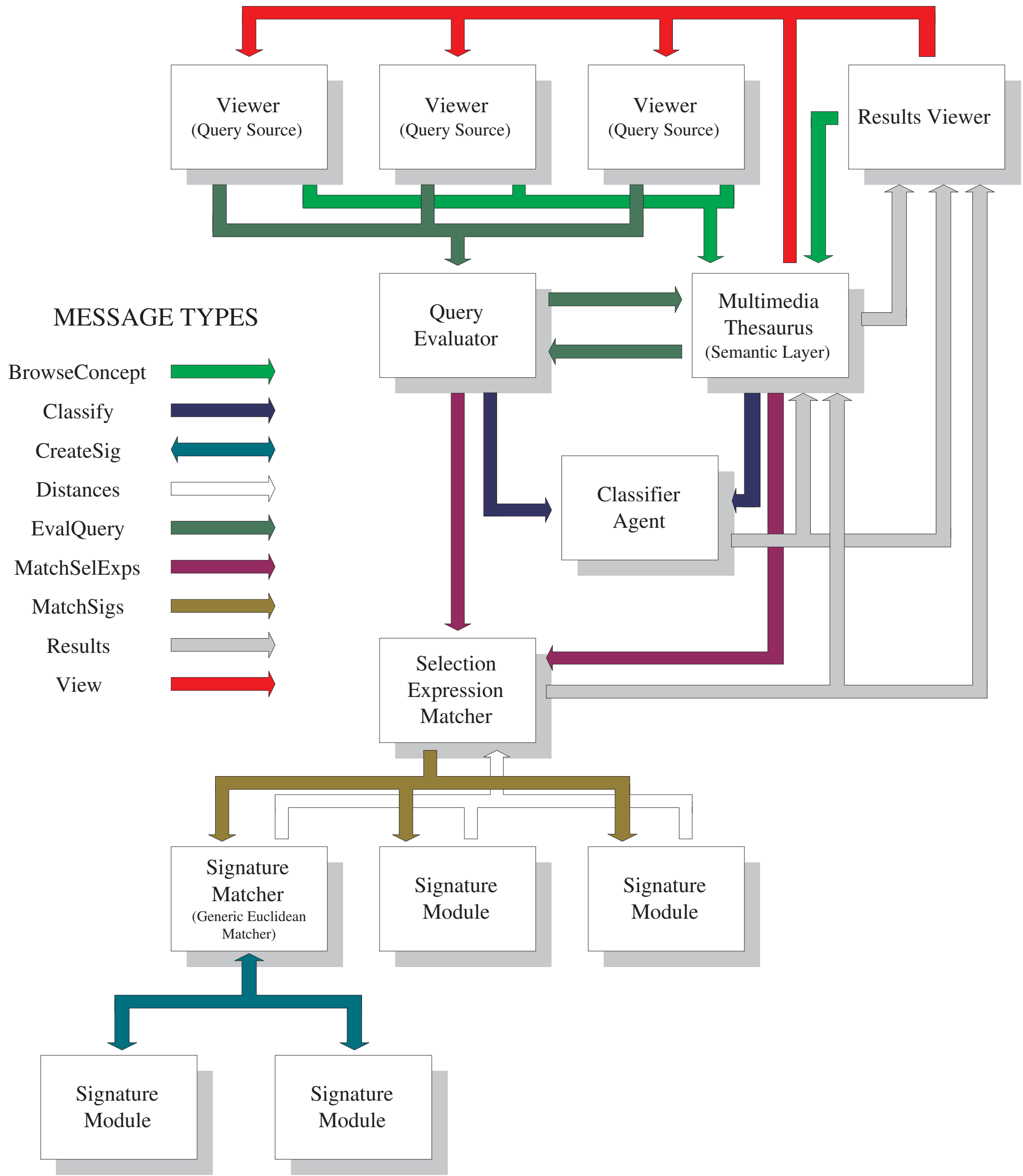


Figure 5.3: MAVIS 2 Processes

Action	Input	Output
MatchSignatures	L(image/jpeg)	*

Table 5.2: Example MAVIS 2 Message Specification

Since messages are conveyed using HTTP, MAVIS 2 processes are effectively mini Web-servers, and as such the address can be given as a URL. To send a message, the broker connects to the server and POSTs an XML encoded form of the message.

It can also be seen from the table that not all messages will prompt a response, for example the *View* message. It can also be seen that a message may be received by more than one process, and processes may register to receive more than one type of message. The input and output types may also affect which processes receive it. These input and output types may be wildcards.

When a message sent by a process is received by the broker, it also has associated with it *Action*, *Input* and *Output* fields. The message is forwarded to any process with a registry entry matching it. For example the message shown in table 5.2 would be directed to the the address `http://ernie:9120/Mavis`.

By using this dynamic mechanism, processes can register and "deregister" their services as required, transparently to the rest of the system. Processes are also able to look up details of available services from this register using *Lookup* messages. This can be used to find out which signature modules are available for instance.

The broker is central to MAVIS functionality, and needs to be responsive and fast. Accordingly it should be run on a high performance machine, preferably with a high priority.

5.4.2 Store

Receives messages: DeleteObject, GetObject, GetObjects, ModifyObject
Sends messages: *None*

The MAVIS 2 data described in section 5.3 is held by one or more *Store* processes. These processes are not depicted in figure 5.3 since most processes use the Stores, and representing this in the diagram would overly complicate it.

Objects stored by the Store are generally transmitted in XML. Each object stored has a unique identifier (ID). This identifier is of the form:

`StoreName_OBJECTTYPE_UniqueIDNumber`

This identifier is used by objects to refer to each other, and by processes to retrieve and modify the objects in the Store. For example, a Selection object will contain the ID of the RawMedia object of which it is a selection. The inclusion of the Store name in the identifier removes any ambiguity if there are multiple Stores running.

A description of the object manipulation messages follows. In the future, this message may have to include authentication information if security is a requirement of the system.

DeleteObject messages remove the object with a given ID from the store.

GetObject messages retrieve a single object with a given ID.

GetObjects messages retrieve a set of objects given some criteria. These include:

- Optionally, an ID or URI that retrieved objects must refer to. For example, this is useful for finding which links are available from a particular selection expression.
- The type of retrieved objects.
- Optionally, an XML tag with content to match. One example of where this is useful is when collating lead-ins — the 'AssocationType' tag must equal 'LEADIN'.

ModifyObject messages overwrite an object with a supplied replacement.

Stores are an integral part of the MAVIS 2 system. Any operating MAVIS 2 system requires at least one Store.

5.4.3 Viewer/Query Source

Receives messages: View

Sends messages: BrowseConcept, EvalQuery

As with most open hypermedia systems, information about links (and the semantic layer) is held separately from the media itself, and the multimedia data is held in its native format. Rather than having one monolithic viewer that must cater for all (possibly proprietary) media types, separate *viewers* handle different media types. In a similar manner to Microcosm (described in section 2.3.4), viewers may be fully or partially MAVIS aware, or unaware.

Thus, there are a number of viewer processes depicted in figure 5.3. To be fully MAVIS 2 aware, a process must be able to accept *View* messages, find link information from any Stores, and send *EvalQuery* messages. Less aware viewers may be started by a MAVIS 2 wrapper process, that may also start queries using the clipboard or via a proxy.

A viewer must give a query a unique query ID. This is so that when corresponding results messages are broadcast, it is known where the results should be sent.

5.4.4 Query Evaluator

Receives messages: EvalQuery

Sends messages: MatchSelExps, Classify

It is the Query Evaluator that decides how best to deal with a query. Upon receiving a query, the Query Evaluator examines the scope of the query, held within the *EvalQuery* message. According to this scope, messages may be sent to the brute force *Selection Expression Matcher*, to the *Multimedia Thesaurus*, or the *Classifier Agent*.

The Query Evaluator will also work out what to match the query against, and pass this on to the relevant process. For example, the query may be a content-based navigation query. The Query Evaluator will then know that only the source anchors of links should be retrieved. In some cases, the Query Evaluator will not send messages directly to the Classifier or Selection Expression Matcher, but will relinquish control of the query to the Multimedia Thesaurus. An example of this would be when the user wishes to expand a query. In this case, the MMT will find the concept relevant to the query before starting any further queries.

5.4.5 Selection Expression Matcher

Receives messages: MatchSelExps, Distances

Sends messages: MatchSigs, Results

The *Selection Expression Matcher* performs the 'brute force' matching of media objects using low-level features. The process is started when a *MatchSelExps* message is received. This contains information about the query selection expression, and which selection expressions it should be matched with. The matcher works out which signature modules should be used to match query selections. Some selections may be 'qualified'; that is, a particular set of signatures should be used to match it since they are representative of the concept or link they are connected with. The matcher then sends out appropriate *MatchSigs* messages to the modules.

These messages are sent without waiting for a reply, since different modules may take different amounts of time to return results. Results from the faster signature modules should be made available to the users as quickly as possible.

The Selection Expression Matcher receives *Distances* messages from the signature modules, containing the 'distance' between a query selection and selections matched against. These are sent to appropriate processes (via the broker) as a *Results* message, which contains the query ID.

5.4.6 Signature Modules

Receives messages: MatchSigs, CreateSig

Sends messages: Distances

The signature modules are responsible for extracting and using low-level features to compare media objects. Modules may be self-contained, or use the default Euclidean distance matcher provided by the MAVIS 2 core.

A signature module that wishes to manage its own index of features must register to receive the *MatchSigs* message. Upon receiving the message, such a module uses its index to find the best matches with the query selection, and sends these back as selection-distance pairs in a *Distances* message.

The *MatchSigs* message may contain information about which subset of a selection's features should be matched against, so the module needs to maintain a map from features to selections. Thus, the signature module does not need to decide which features to match against (e.g. only link source anchors) — this is decided by the *Selection Expression Matcher.*

A simpler way of writing a signature module is to let the default MAVIS 2 Euclidean distance matcher handle the indexing and matching. In this case, the module should respond to the *CreateSig* messages. When it receives such a message, it should extract the relevant feature or features, and return them to the MAVIS 2 signature matcher. The features must be specified as n-dimensional vectors of real numbers where n is constant for a particular signature type.

These two methods allow signature modules to be integrated with the MAVIS 2 system with very little effort, or to have full control over the indexing and matching if required. An additional advantage of using the simpler approach is that an agent subsystem, such as the *Classifier Agent* described below, can utilise the signature without modification.

5.4.7 Classifier Agent

Receives messages: Classify

Sends messages: Results

The *Classifier Agent* process is the result of work by Dan Joyce, another member of the MAVIS 2 team (Joyce *et al.*, 2000). The classifier does not just respond to messages; it also works proactively in the background, discovering novel associations between concepts and feature vectors, and hence between concepts and media objects.

The agent process actually encompasses a multi-agent population. These agents are trained to possess knowledge of specific concepts, and specific signature (feature vector) types.

Upon receiving a *Classify* message, an agent wrapper distributes the features of the query to appropriate agents. These agents return results in the form of concept-confidence pairs. A weighting sum is used to fuse these results, and these results are sent to the *Results Viewer*.

The Results Viewer can send back information about the user's satisfaction with the agents' classifications. This information is used when weighting the confidences returned by agents in the fusing process.

Further discussion of the agent subsystem is outside the scope of this thesis. From the point of view of this work, the agent subsystem can be treated as a 'black box' into which a query object is dropped, and lists of confidences and concepts returned.

5.4.8 Results Viewer

Receives messages: Results

Sends messages: View, BrowseConcept

Any navigation or retrieval query initiated by the user will conclude with the presentation of the results of that query to the user. It is the *Results Viewer* process that handles this.

The Results Viewer may receive many messages concerning a single query. The *Selection Expression Matcher* sends at least one *Results* message for each signature type used in the query. In addition the *Classifier Agent* and *Multimedia Thesaurus* may send additional results. Each of those messages has an associated query ID; this is used to determine which results pertain to which query.

Upon receiving a *Results* message, the results viewer first has to decide whether it is appropriate to display those results. In some cases the results should be consumed by the *Multimedia Thesaurus* and not displayed to the user. If this is not the case, the results viewer opens a new results window if the query ID has not been previously encountered, otherwise the results are sent to the appropriate existing window. When a results window opens, the results are displayed and their ordering may be altered as more results come in.

Depending on the type of the original query, the results may be primarily media objects, links, or concepts. The first two are actually very similar, and results are processed and sorted in the same way, so these are considered identical; it is only what is displayed that differs.

The user may click on retrieved media (selection expressions), link destinations or concepts in the results window. A *View* message is sent if a link or selection expression is clicked on. The appropriate viewer is started up if necessary and the retrieved media or destination of a link is displayed. If the user clicks on a concept, a *BrowseConcept* message is sent to the *Multimedia Thesaurus*, which then opens a concept browser at the appropriate location.

5.4.9 The Multimedia Thesaurus

Receives messages: BrowseConcept, EvalQuery, Results

Sends messages: EvalQuery, MatchSelExps, View

The *Multimedia Thesaurus* or MMT is an implementation of a semantic layer in MAVIS 2. It controls the concept layer, connections between concepts, and associations with selection expressions. It assists the user in the ways described in chapter 4.

Whether or not a query is to involve the MMT is decided in the Query Evaluator process. This is decided based on user settings, though in the future some more intelligent reasoning could make the decision. If the query is to involve the MMT, the EvalQuery message is forwarded to the MMT process; the MMT has effectively intercepted the query.

Sometimes, the MMT will issue further queries using EvalQuery or Classify messages. It may 'intercept' the produced Results messages in order to make a decision about how to proceed further. On occasion the MMT will need finer-grained control of the scope of queries; in these cases, a MatchSelExps message will be sent directly to the Selection Expression Matcher.

Figure 5.4: The MAVIS 2 'Control Room'

Further details on how the MMT operates in MAVIS 2 are given in section 5.5.2.

5.5 The MAVIS 2 System in Operation

The MAVIS 2 system consists of a number of different processes. To start up a MAVIS 2 system, each of these processes must be started up separately. To facilitate this, MAVIS 2 includes a 'Control Room' program, that allows easy switching in and out of processes. This is shown in figure 5.4. The Control Room reads in process information from a configuration file and starts a MAVIS 2 broker. It may then automatically start processes according to options set in the configuration file. Usually Stores are started first.

MAVIS 2 processes can also be started as 'daemons' that sit on server machines. Unfortunately, at the present time, MAVIS 2 does not support multiple users. However the system has been designed with this in mind, so the alterations to enable multi-user functionality are minimal. In the enhanced system clients could connect their Viewer and Results Viewer to a running broker, without having to start up a whole new MAVIS 2 system.

Figure 5.5: The MAVIS 2 URL Viewer

5.5.1 Navigation and Retrieval

Use of a MAVIS 2 system typically starts in a Viewer. If the Viewer is MAVIS 2 aware, it can use the Stores to find out if there are any links from the currently viewed media object, or concepts directly associated with it. Any links or concepts found will be displayed. Figure 5.5 shows the default MAVIS 2 viewer, the URL viewer, that can display HTML text, and images in GIF or JPEG format.

Also on the right of the URL viewer is the *qualification panel*. This panel displays the signature types that can be used with the currently displayed selection. A specific set of these can be selected by the user, or the user can simply opt to use all available signature types.

At this point, the user has a number of retrieval and navigation options open to them.

Following a specific link directly from the Viewer. In this case the Viewer sends an appropriate *View* message to be handled by the appropriate Viewer.

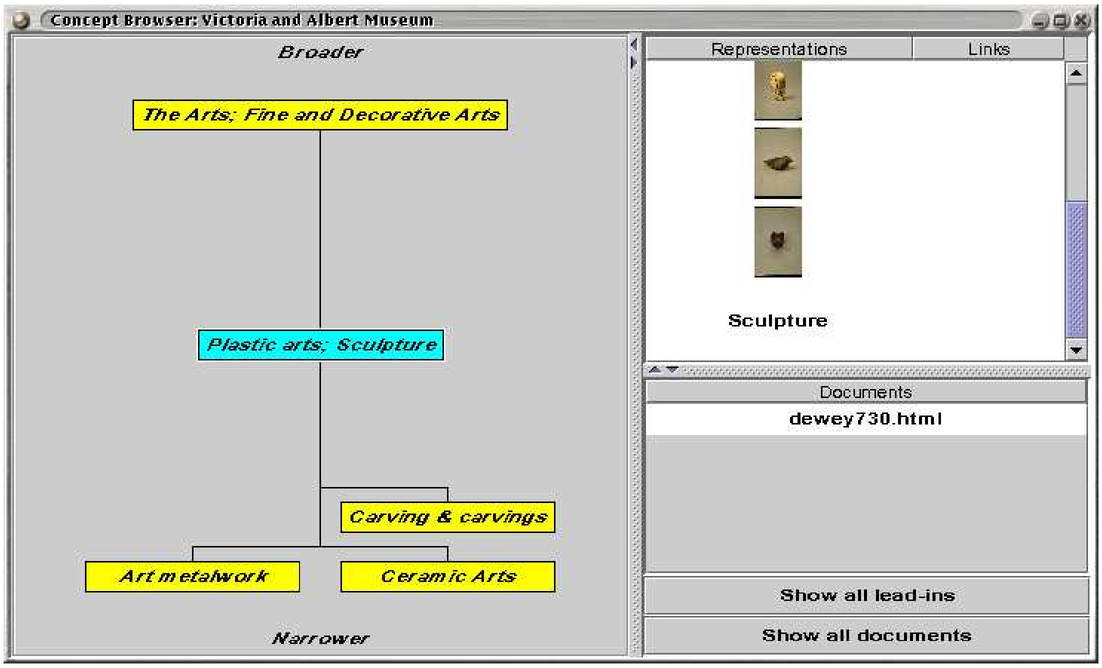


Figure 5.6: The Concept Browser

Authoring a link within the Viewer. The user selects the *Start Link* option from the "Action" pull-down menu, and the currently viewed selection is made into a selection expression and becomes the source anchor of the link. The signature types selected in the qualification panel will be used when matching queries with that anchor. Then the destination of the link is reached by some means (often it may be in another window), and the *Create Specific Link* or *Create Generic Link* item selected. This creates the link and stores it in one of the MAVIS 2 Stores.

Browse concepts by selecting a concept connected to the displayed object in the Viewer. The Viewer dispatches a BrowseConcept message that causes the Multimedia Thesaurus to open a *concept browser* (shown in figure 5.6 at the appropriate concept. The user can then browse around the network of concepts in the multimedia thesaurus.

Associating the selection with a concept. If the user wishes to associate a selection with a concept, the Viewer makes the selection into a selection expression, and sends that in a 'CreateAssociation' message to the Multimedia Thesaurus process. The MMT then opens an *Associator* tool that allows the user to choose a concept. The MMT may perform a classification operation to make an initial guess.

Performing a query is how other navigational operations are performed. The

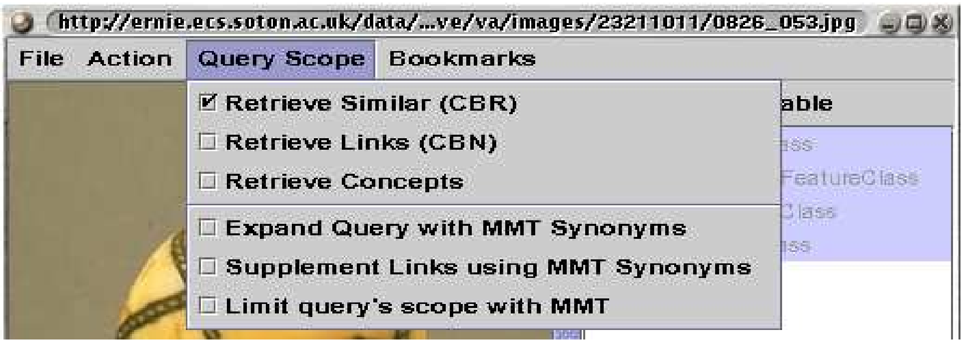


Figure 5.7: Query Scope Options

user sets the parameters of the query, and then starts the query evaluation process. How the query is processed by the rest of the system depends on user settings. The menu of available settings is shown in figure 5.7. Each of the options are described below.

- *Retrieve Similar (CBR).* If this option is selected, the system will use the signature modules (via the Selection Expression Matcher) to find selection expressions (and hence, media objects) similar to the query selection. This is content-based retrieval (CBR).
- *Retrieve Links (CBN).* If this option is selected, the system will use the signature modules to try and find links whose source anchors are similar to the selection. This is exactly the same process as for content-based retrieval above, except that the scope of the search is restricted to the source anchors of links, and it is the destinations of these links that are primarily of use to the user. This is content-based navigation (CBN). The Results Viewer will display the source anchors of retrieved links when presenting the results of a CBN query anyway, since this may assist the user in determining whether or not following the link is useful.
- *Retrieve Concepts.* This option instructs the system to find concepts relevant to the query selection. A Classify message may be sent to the Classifier Agent. Additionally, the Selection Expression Matcher is used to match the query selection with selection expressions that are lead-ins to concepts. More details of this are given in sections 5.5.2 and 5.5.3.
- *Expand Query with MMT Synonyms.* This item turns on further query expansion, as was described in section 4.2.3. In this case the MMT will intercept the query, and proceed as was described in section 5.5.2.

- *Supplement Links using MMT Synonyms.* Selecting this option turns on link augmentation using the MMT, as described in section 4.2.5.
- *Limit Query's Scope with MMT.* This causes the MMT to limit the scope of the query as was described in section 4.3.3.

After setting these options, the user selects the *Perform Query* menu item to start the query. A results window may then appear. This may contain the final results of the query, or it may ask the user to confirm a classification so that the MMT may carry on expanding a query.

5.5.2 The Multimedia Thesaurus

Section 4.6.1 listed some requirements of an outside system. MAVIS 2 fulfills these requirements:

1. Selections may be qualified; i.e. they specify which features are relevant for matching, as described in section 5.3.2. Thus, media features can be queried independently.
2. Selections may also include a shape specification, so the system can operate on specific portions of media.
3. Signatures and intermediate representations may also be stored.
4. Through use of the *MatchSelExps* message, the MMT can specify exactly what should and should not be in the scope of a retrieval operation.
5. The results viewer includes code that can combine results from different signature modules, as described in section 5.5.3.
6. MAVIS 2 can determine whether the results of a query should be displayed to the user or 'consumed' by the MMT. If it starts such an 'internal' query, the Query ID is prefixed 'MMT_'; the results viewer knows not to display the results of such a query.

Content-based retrieval can be treated as a black box, since an *EvalQuery* or *MatchSelExps* message can be sent, and the corresponding *Results* messages listened for.

The MMT is involved in each of the seven ways described in section 4.2:

What is this? In this case, the query is intercepted, and the query classified by the classifier agent and the brute force matcher. The results of these are then sent to the Results Viewer, since that process contains the code necessary to sort and present the results.

Alternative representations can be presented in the concept browser, in a Results Viewer window or an MMT-aware viewer. Such viewers can obtain the information about alternative representations straight from the Store, since all the necessary information is held in the data model held in the Store.

Clicking on a representation in a view will result in an appropriate view message being sent.

Query expansion operates on two levels as has been described. In simple expansion, the Results Viewer can gather alternative representations of retrieved concepts without explicitly requiring intervention by the MMT process. Further expansion requires that the MMT process intercept the query as described above. The MMT then classifies the query in the same manner as during the 'What is this?' query. Representations of the resulting concept are used to launch further queries via EvalQuery messages.

Indexing documents. Documents indexed on a concept are displayed in the concept browser. Additionally, the Results Viewer may display these with the corresponding concepts.

Link augmentation is also handled by the Results Viewer. When the source anchor of a retrieved link is connected to a concept, links from other representations of that concept are also presented. The Results Viewer communicates with the Store independently to achieve this.

Concept navigation is performed by the user in a Concept Browser started by the MMT process. A screenshot of the browser is shown in figure 5.6. On the left is a portion of the concept network arranged hierarchically. The central concept is the 'focus' or point of interest. Representations of the concept are shown in the top right list, and indexed documents in the lower right list. The "Show All" buttons gather all of the representations of a concept and its specialisations. This allows the user to see all examples of sculpture in the system, for example.

A Results Viewer or a MAVIS 2 aware viewer may know that what it is displaying is associated with a concept in the MMT. If this is the case it can send a *BrowseConcept* message to the Multimedia Thesaurus, which starts up a Concept Browser.

Scope narrowing requires that a query is intercepted and the most relevant concept found, as for query expansion. The portion of the MMT to be used must have been selected by the user using the scope specifying tool, shown in

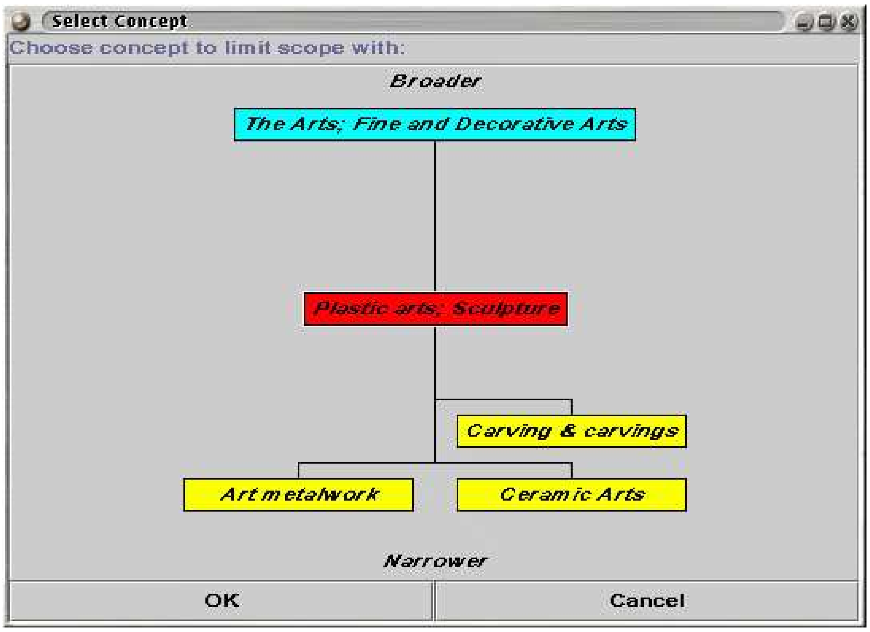


Figure 5.8: Scope Limiting Selection Tool

> figure 5.8, in advance. The MMT process collects everything associated with the concepts in that portion, and starts another query by sending a MatchSelExps message. The message is sent to the Selection Expression Matcher directly, circumventing the Query Evaluator, since a greater degree of control over the scope is required than can be specified in an EvalQuery message.

The following section describes how during a query, results from separate signature modules are combined to form a single ranked list of results.

5.5.3 Result Sorting

During the course of a query, a variety of signature modules may be used to retrieve results. Since results for different signature types arrive separately, the results viewer must 'reassemble' or fuse these results as they arrive.

This is no trivial task, since the results may consist of selections with various signatures tagged as relevant ('qualified' selections). Because of this, not every retrieved media object (or concept) may have the same match information available.

Additionally, no assumptions can be made about the distances between selections returned by the signature modules. Some features have fixed ranges of distance values that could be normalised to a range of 0–1 for example. However, others are unbounded, and it is not possible to know in advance the range of the distance values between selections. Accordingly, the only assumption we can make reliably about the distances returned by signature modules in general is that they will be real numbers greater than or equal to zero, with zero being a perfect match, and the larger the number, the more dissimilar the two compared selections.

This problem is similar to the collection fusion problem described by Vorhees *et al.* (Vorhees *et al.*, 1995), although their proposed solution was relatively complex, requiring the storage of previous results of queries. A simpler sorting algorithm has been developed to cope with this situation. The algorithm is used in slightly different ways, depending on whether the query is required to retrieve media and/or links, or a list of concepts (a classification).

The sorting algorithm works only on selection expressions, and assumes each selection expression will contain a single, 'like this' selection. For simplicity, the description below only refers to selections, and not the selection expressions in which they appear.

Sorting for Media and Links

Content-based navigation and retrieval operations involve comparing and retrieving media objects. Thus, the aim of the sorting algorithm in this case is to produce a rank ordered set of retrieved media objects. If links are to be retrieved, the same algorithm can be applied to the source anchors of those links, which are also media objects.

The sorting algorithm is described formally below.

1. Each query is initiated on a query selection, referred to as S_q. Some number of signature types will be relevant for that query selection. These may be specified as part of the selection (a 'qualified' selection, as described in section 5.3.2). Otherwise, all signature types that can be used to match the query selection with other selections are used. This set of relevant signature types is defined as $SigTypes$.
2. Each query has a 'scope', that is, the set of selections that can be retrieved (source anchors in the case of a content-based navigation operation, or any selections of a similar media type in the case of a content-based retrieval operation). This set of selections is defined as $MatchSelections$.
3. For each such selection, we may have a distance value returned by a signature module, if that signature module is relevant to use for both query and retrieved selections. We define this as the distance function $dist$:

$$\forall S_m \in MatchSelections \mid dist(S_m, sig) \geq 0, sig \in SigTypes \qquad (5.1)$$

where S_m is a retrieved selection, and sig is a signature type.

4. For each signature type in $SigTypes$, we have a set of distances:

$$Distances(sig) = d_1, d_2, \dots d_i \tag{5.2}$$

such that

$$\forall d \in Distances \mid d = dist(S_m, sig), S_m \in MatchSelections \tag{5.3}$$

Each such set is then ordered, and each match given a normalised ranking value between 1 (for the best-matching) to 0 (for the worst-matching). The ordering is a *weak ordering*, so if two matched selections have the same distance, they have the same rank. In practice this is somewhat unlikely if the distances are real numbers.

The distance function $dist$ can now be replaced by a rank value function $rank$:

$$\forall S_m \in MatchedSelections \mid 0 \leq rank(S_m, sig) \leq 1, sig \in SigTypes \tag{5.4}$$

The value of this function is zero in those cases where a signature type relevant for the *query selection* is not relevant for the *matched selection* S_m. For example, if colour and shape are relevant for the query selection, but only colour is relevant for a retrieved selection, the rank for the shape signature is zero.

Each distance between the query selection and a retrieved selection is now normalised as a similarity value between 0 and 1. Since the duration of this normalisation is for a single query, no assumption has had to be made about the absolute maximum distance for a particular signature type.

5. Each signature in $SigTypes$ has an associated weight, $W(sig)$, that may be supplied by the user or system. These weights are real numbers between 0 and 1 and sum to unity:

$$\sum_{sig \in SigTypes} W(sig) = 1 \tag{5.5}$$

For each matched selection, these weights can be used to create an overall rank value:

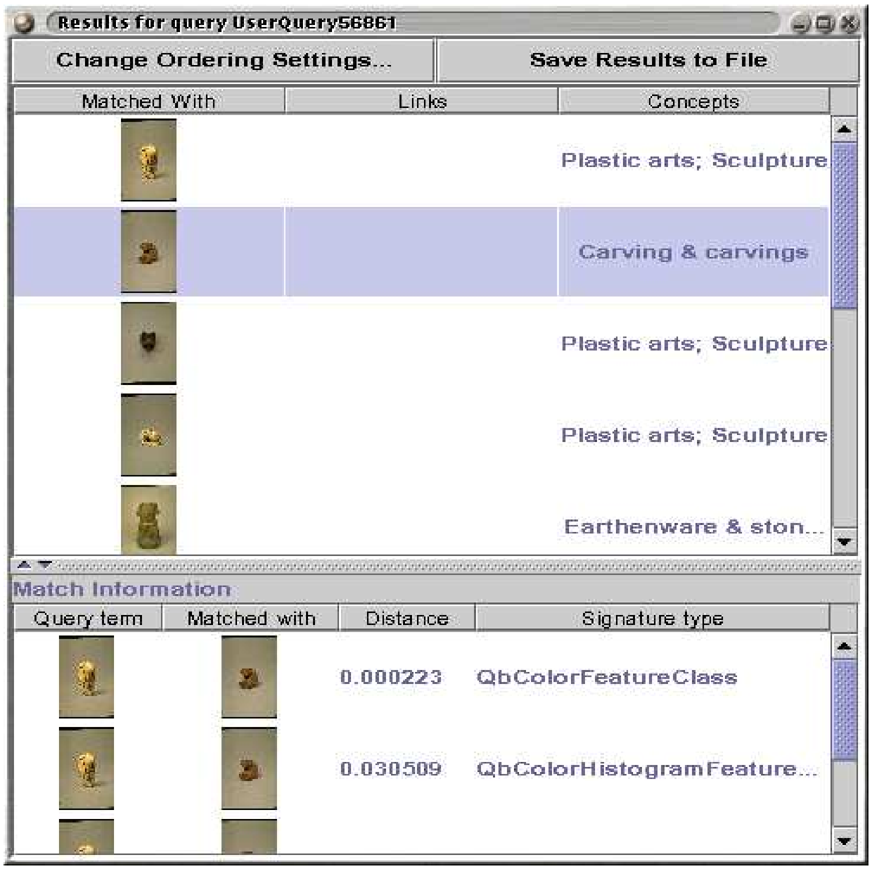


Figure 5.9: The MAVIS 2 Results Viewer

$$overallrank(S_m) = \sum_{sig \in SigTypes} rank(S_m, sig)W(sig) \tag{5.6}$$

This gives selection S_m an overall rank value between 0 and 1. These overall ranks can then be used to sort the final list of selections. Selections with higher values for *overallrank* appear higher in the list.

The list of results produced by this algorithm is displayed in a results window, together with links if the matched selections are source anchors of links. Information about each match between two selections with each signature module are displayed in the window so, if they wish, the user can see exactly why the system retrieved a particular object. This is shown in figure 5.9.

If the user clicks on any thumbnail of a media object, be it a retrieved object or a link destination, an appropriate View message is dispatched. Clicking on a concept sends a *BrowseConcept* message.

Sorting for Classification

During queries involving classification, concepts are the effective result of the query. Evidence for each concept may arrive from the Classifier Agent or from the signature matching. How results are sorted in this context is slightly different, since the results should be presented as a list of concepts, with the concept most confidently judged as relevant at the top.

Results from the Classifier Agent are treated as results for a single signature type. The classifier agent becomes another signature module, i.e. another member of the set $SigTypes$. The confidences provided by the agents are mapped to distances, i.e. higher confidences are mapped to lower values. This allows confidence results from the classifier agent to be integrated into the results of a query using the algorithm described in section 5.5.3 above.

Steps 1–5 from section 5.5.3 can then be used to give each representation of a concept an overall rank value ($overallrank$ in equation 5.6).

It is likely that we may have a number of representations for each concept, that have different overall rank values. These values must be used to produce a single rank value for the associated concept in order to produce a ranked list.

The methods of classification described in section 4.6.2 make the assumption that clustering (and all inductive learning techniques) make: For a given feature, all representations of a concept will have similar values, or at least are more similar to each other than to objects in other classes. It was argued that this may not hold for all multimedia data.

Given that different representations of a single concept may differ significantly, the overall rank values of representations of a concept will differ correspondingly. So, to take into account all of these rank values is not likely to yield good results, since to produce a high confidence, a query must match well with all of the very different representations.

Consider the query shown in figure 5.10. The side view representation of a horse matches best with the query image. The side view of a dog also matches well. However, the front-facing view of a horse does not match well with the query. If the front-facing view is considered evidence against the proposition that the query image depicts a horse, the overall evidence that the query image depicts a horse is reduced. In this case, the query image is then incorrectly classified as a dog. It is

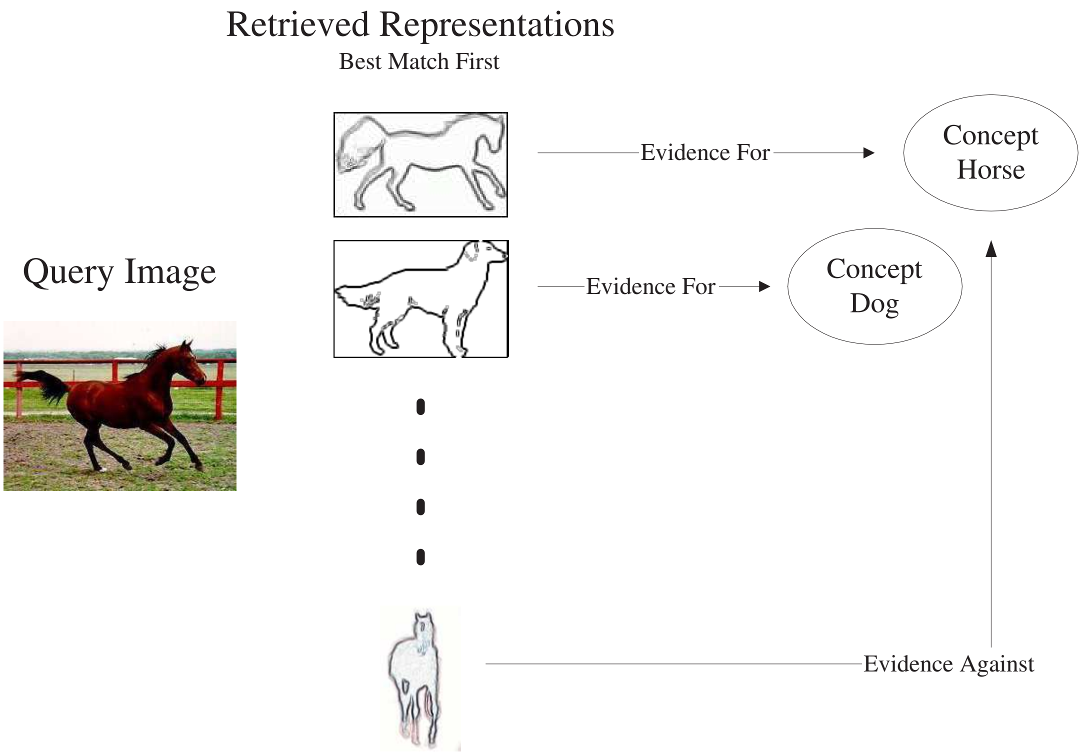


Figure 5.10: Use of Best Match Only during Classification

clear that in this case, and in other cases where a concept has a number of visually different representations, that only strength of the best match is appropriate to use.

Therefore, the highest overall rank value of any representation of each concept is used as the overall rank value for that concept. The list of concepts can then be sorted and presented to the user according to these values.

5.6 Signature Modules

This section describes the signature modules that have been implemented as part of the MAVIS 2 project. These cover text and image media. The signature modules implemented so far are:

- HSV and RGB colour histograms, written by David Dupplaw
- Spatial HSV and RGB colour histograms, written by David Dupplaw
- A signature module allowing the use of IBM's Query By Image Content (QBIC) features
- A word-matching text signature module by Mark Weal.

5.6.1 HSV and RGB Colour Histograms

The RGB and HSV colour histograms are variants of the same signature module, and use most of the same code. They compare images using colour.

An RGB (red-green-blue) colour histogram is generated from each image by counting the number of pixels whose colour falls in a certain range. In this implementation, there are three 'bins' in each dimension R, G and B, making up $3^3 = 27$ bins in total.

The HSV (hue-saturation-value) colour model is closer to the human perception of colour, so these histograms are often of more use. An HSV histogram is generated in much the same way as the RGB one, but the colours are translated into the HSV space, and the histogram has six bins in the hue dimension, three in the saturation dimension, and three in the value dimension, resulting in $6 \times 3 \times 3 = 54$ bins in total.

The histograms of two images are compared using a quadratic distance measure, that takes into account the similarity of other bins as well as of corresponding bins in each image. The result is a single measure of dissimilarity between the images. The dissimilarity between each image and the query image is dispatched as a *Distances* message once this is complete.

5.6.2 Spatial HSV and RGB Colour Histograms

This module uses histograms in the same way as the previous signature module, but first divides each image into a grid of subimages. By comparing corresponding subimages, the similarity measure produced takes into account the location of colour in the image (Dupplaw *et al.*, 1999).

The URL viewer features a grid selection mechanism. A displayed image is divided up into a grid, and the user can select which grid squares should be used during a query. In this way, the user can retrieve images (and hence, associated links and concepts) based on the image having similar colours in specific areas.

5.6.3 The QBIC Wrapper Signature Module

The QBIC content-based image retrieval engine, described in section 2.2.4, has been integrated into MAVIS 2 by means of a signature module wrapper. The signature module responds to MAVIS 2 messages, invokes the QBIC engine, and sends the results produced back to the MAVIS 2 system.

QBIC provides the following features:

QbColorFeatureClass. This judges the similarity of images based the average colour throughout.

QbColorHistogramFeatureClass. This generates HSV colour histograms, and uses these as the basis of comparison.

QbDrawFeatureClass. This divides the image into a grid, and uses the dominant colour in each subpart to compare images.

QbTextureFeatureClass. This calculates the directionality, coarseness and contrast of the texture of the image.

QbTextFeatureClass. This retrieves images based on text keywords assigned to them. This feature is not used by MAVIS 2.

The preceding features all operate on entire images. QBIC also has versions of some that can operate over a 'subpart' of the image:

- QbColorSubPartFeatureClass
- QbColorHistogramSubPartFeatureClass
- QbTextureFeatureClass

Unfortunately these are not fully implemented in the current version of QBIC. To allow the QBIC signature module to operate on subparts of images, the signature module must extract the relevant subpart and present it to QBIC as an entire image.

The QBIC system works by maintaining catalogues within databases. A MAVIS 2 catalogue must first be created. Any images that are to be retrieved must then be imported into the QBIC, which then indexes them. The query image must also be imported.

The QBIC system does not have an understanding of MAVIS 2 selections. The signature module wrapper must present images to the QBIC system, and map the results back to MAVIS 2 selections. When images are imported into QBIC, their ID is the filename they had when they were imported.

The following is the process the QBIC signature module uses to introduce images, map them to MAVIS 2 selections and perform queries.

1. When the signature module is initialised, it first asks QBIC which images it knows about. If necessary, a new QBIC database and/or catalogue are created. When images are imported into QBIC, they are given a filename of the form:
   `MAVIS2_Selection_ID.gif`
   Where '.gif' is a file extension given according to the storage format of the raw media the selection is from. QBIC requires this extension to determine the format.
2. When a 'MatchSigs' message arrives, the signature module first ensures that the QBIC catalogue contains all the images concerned with the query. Any

images not in the catalogue are first downloaded (from the URI) to a local location, and given the required name as described in step 1. They are then imported into the QBIC system. After the import is complete, the local copy of the image can be removed.

3. Once all necessary images (including the query image) are imported, QBIC is queried specifying the relevant feature class. If the number of images in the catalogue is small, distances for the whole set of images is returned. It is usually necessary to ask for more than the top n if the top n is required, since not all of the top n images may be in the scope of the query.
4. Results are mapped back to the MAVIS selection ID's by removing the file-name extension. Those selection IDs and distances relevant to the query are then 'packaged up' in a *Distances* message, and dispatched back to the Selection Expression Matcher.

In this way QBIC is transparently integrated into the MAVIS 2 framework. It is likely that other systems may be integrated in a similar way.

5.6.4 Word Matching

Text is treated as any other medium in MAVIS 2, and thus to support text matching, at least one text-based signature module is required. The current implementation of text matching in MAVIS 2 is the Word Match signature module written by Mark Weal.

The matching is quite simple, and has been designed to cope with various kinds of matching, including phrase to document matching, phrase to phrase matching, and document to document matching. However, it does not take into account word frequency and does not stem words before matching.

To compare two pieces of text, two sets are formed, C and D, where the members of C and D are the words in each document, minus stop words. Also it is enforced that $|C| < |D|$.

The set of words in both pieces of text is found:

$$W = \{w_1, w_2, \ldots, w_n\} \tag{5.7}$$

such that

$$\forall w \in W \mid w \in C, w \in D \tag{5.8}$$

When comparing two pieces of text, the distance value is given by:

$$d = 1 - \frac{|W|}{|C|} \tag{5.9}$$

Thus, if C is a subset of D, the distance is zero, and if C contains no words that D contains, the distance is 1.

At some later date, this module would ideally be replaced by an existing, more sophisticated technique.

5.7 Summary

The MAVIS 2 system is a content-based navigation and retrieval system, that can operate over a heterogenous distributed environment. Processes register their services with a central broker. Processes communicate through this broker, specifying the operation they require and any required inputs and outputs. The broker then forwards the messages to the appropriate processes. Using this mechanism, processes can transparently register and "deregister" their services at will, and the rest of the system benefits.

The MAVIS 2 system includes a multimedia thesaurus or 'MMT', an implementation of a semantic layer that supports the traditional thesaurus-style relationships, *broader/narrower* and *related*.

Metadata about media objects, MMT concepts and links are held in *Store* processes. Only references to media objects are held, in the form of URIs; thus the system can scalably cope with distributed information.

Low-level media features can be used in the query process by implementing *signature modules*. A signature module can perform the whole indexing and matching process for a particular feature by registering to receive *MatchSigs* messages, or may choose to make use of the default MAVIS 2 Euclidean distance matcher. In the latter case, the indexing and matching is managed by the MAVIS 2 matcher, and the signature module need only extract feature vectors from the media objects in response to *CreateSig* messages.

MAVIS 2 also features a Classifier Agent, developed by Dan Joyce as part of related agent-based research in the MAVIS 2 project (Joyce *et al.*, 2000). The agents use inductive learning techniques to attempt to provide better and faster classifications of query objects.

How navigation and retrieval is performed in MAVIS 2 has been described, including how results from a number of signature modules are combined and presented to the user.

Four signature modules have been implemented, allowing MAVIS 2 to retrieve and navigate based on the content of images and text. Further media may be supported by writing new signature modules; existing code (other than possibly viewers) need not be altered.

Thus, there now exists a flexible multimedia information retrieval and navigation architecture, supporting the semantic layer technique in the form of a multimedia thesaurus. This system can be used as the basis for testing and evaluating a semantic-layer assisted multimedia information system.

Chapter 6

Building An Application

6.1 Introduction

The previous chapters have described the semantic layer technique, and how this has been implemented as a multimedia thesaurus in the multimedia information system, MAVIS 2. In order to demonstrate the feasibility of the technique, and to evaluate its usefulness, a multimedia collection is required, and this must be connected to the semantic layer. How this can be achieved without the prohibitive amount of effort required to connect everything manually is the subject of this chapter.

There are a wide variety of ways to perform this task. Chapter 4 describes various techniques. This chapter describes how an exemplar multimedia collection from a museum is connected to the semantic layer.

The basis of the semantic layer developed for this application is a subset of the Dewey Decimal Classification system, a widely used classification system with a broad scope. A technique called Latent Semantic Analysis (introduced in section 3.5.5) is used to connect images with metadata to the semantic layer. Other images from the collection are classified using purely their content.

The multimedia collection is described first. The Dewey Decimal Classification system is then introduced, and the subset chosen presented and explained. Section 6.4 describes how the Latent Semantic Analysis technique can be used to classify images in order to populate the multimedia thesaurus.

This construction of a multimedia thesaurus application is itself an evaluation of the feasibility of building a semantic layer in a practical situation. This is discussed in section 6.7.

Figure 6.1: Sample Images from the Victoria and Albert Museum Collection

6.2 The Multimedia Collection

The core of the multimedia collection used in this application is a set of 1023 images of artefacts owned by the Victoria and Albert Museum. There are a wide range of artefacts depicted; they include paintings, sculptures, clothing, furniture and textiles. A small sample of images are shown in figure 6.1.

This collection of images had been compiled during a previous project concerning electronic access to images of museum artefacts, called the Electronic Library Image Service for Europe, or ELISE (Seal, 1995).

The images and metadata were provided on a CD-ROM with numbered filenames. Each image has a working image identification number; the filename of an image is a truncated version of that number. For example, the image corresponding to the record shown in table 6.1 is `0761_076.jpg`. All of the images in the collection have image ID numbers starting with 23211011.

Also on the CD-ROM is some text metadata associated with the images. This metadata had been compiled during the course of the first phase of the ELISE project, which ran for two years from 1993. This metadata is held in text files, and takes the form of field-value pairs. An example record from one of the text files is shown in table 6.1.

The metadata is not complete. There are acknowledged gaps. Only around half of the images have any associated metadata, and not all images with metadata have

```
%N Elise
%F museum_number
C.491-1919
C.491A-1919
%F object_name
Jar with Cover
%F object_category
Ceramic
%F material
stoneware, wood (plinth)
%F object_description
Known as 'Wally birds' these grotesque bird-jars loosely inspired by
gothic fantasy and Japanese art were very popular. This one takes the
form of an owl.
%F subject_depicted
Bird, Owl
%F institution
R.W. Martin & Bros.
%F institution_role
Maker, Designer
%F text_date
1899
%F early_date
1899
%F late_date
1899
%F action
Made
%F place
England
%F server
vaserver
%F working_image_id
232110110761-076
%N Elise
```

Table 6.1: A Sample ELISE Metadata Record for a Victoria and Albert Museum Image

Field	Number of Images
action	600
archive_image_id	26
archive_location	26
author	286
author_role	278
born	152
died	170
dimensions	476
early_date	597
institution	73
institution_role	73
late_date	596
material	590
museum_number	612
object_category	589
object_description	592
object_name	610
object_title	177
place	451
server	526
style	117
subject_depicted	503
subject_person	78
text_date	595

Table 6.2: ELISE Metadata Fields

the same fields. Some records in the text files refer to images not on the CD-ROM, and some images have two slightly different records. This presents problems when extracting metadata for a particular image.

A summary of all the fields used and how many images have data for each field is given in table 6.2.

One field of particular interest is the 'object_category' field. It could be supposed that this field constitutes a conceptual representation of the object depicted by the image; unfortunately, these categories form no consistent classification. The terminology is varied, and many images have a unique category. This would not make for a useful multimedia thesaurus; each concept may have only one representation, making classification and linking based on image content very difficult. Additionally the categories have no structured relationships. Thus, a suitable controlled classification is needed.

In summary, the collection of images held in this application has only incomplete and inconsistent metadata associated with it. It seems likely that this will often be the case with existing multimedia collections. Thus, the exercise of constructing a

700	The arts; Fine and decorative arts
730	Plastic arts; Sculpture
738	Ceramic Arts
738.2	Porcelain
738.3	Earthenware and Stoneware

Table 6.3: Example Portion of DDC Hierarchy

semantic layer given the available material can be used as a study of the feasibility of the semantic layer technique.

6.3 The Dewey Decimal Classification Schema

The Dewey Decimal Classification schema (DDC) is a widely used hierarchical classification system conceived in 1873 by Melvil Dewey. It was first published in 1876, and has since become the most widely used library classification system in the world. The classification is still subject to modification to reflect current trends in publishing.

Each class in the DDC has an associated number. Even though the DDC has been translated into over 30 different languages, this number remains constant.

Each class has a number with at least three digits, and possibly more digits after a decimal point. In general, the more digits a class has the more specific it is. The specialities of a particular class share the leftmost digits.

Table 6.3 gives a sample portion of the hierarchy. The significant figures for working out the hierarchy are underlined. The class "The arts; Fine and decorative arts" is a top-level class. Each line is a specialisation of the previous line, except the last two lines, which are both specialisations of the class Ceramic Arts with the decimal code 738. Further specialisations of Porcelain would be of the form $738.2x$, with further digits being added as the specificity increases. These Dewey classes are visualised hierarchically in figure 6.2.

6.3.1 Selecting a Subset for the Collection

The Dewey Decimal Classification is a necessarily large set of classes, as it is designed for libraries with potentially millions of volumes to index in mind. It is obviously pointless to include the entire set in the semantic layer, as only a very small proportion would have any media representations.

Thus, a subset of the classes suitable for the subject domain is required. One of the advantages of using the Dewey Decimal Classification is that a subset can be 'attached' to another with a wider scope. In this way two subsets (and hence

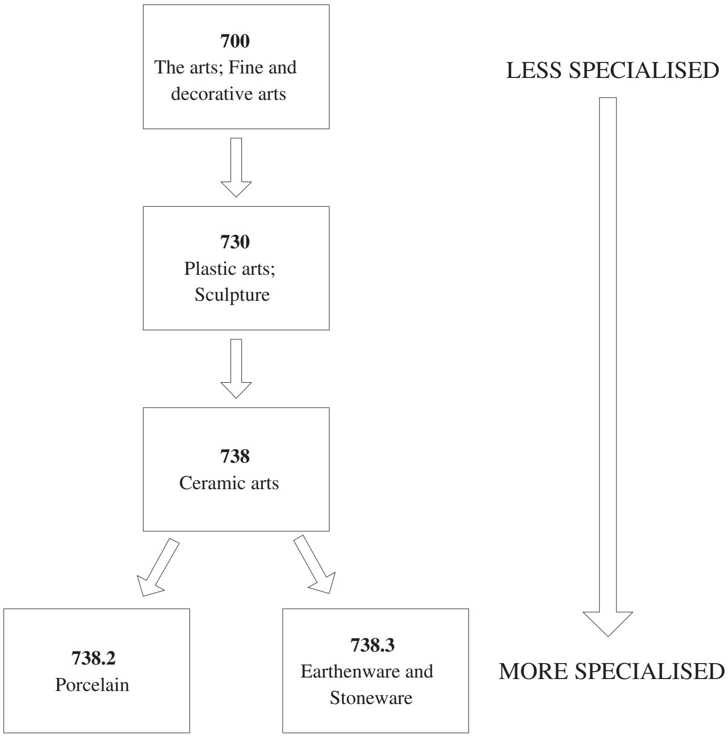


Figure 6.2: A Portion of the Dewey Decimal Classification Hierarchy

multimedia thesaurus assisted collections) can be merged however much or little the subject areas overlap.

Fortunately, related work in the previously described ELISE project is also of use here. As part of the ELISE project, a selection of DDC classes was chosen for indexing museum collections, each with a number of associated keywords. The subset the ELISE project used was designed with several museum collections in mind, and is thus still rather widely-scoped for the Victoria and Albert collection used in this application. A subset has been chosen with the scope of this collection in mind, and not so deep that it will be sparsely represented.

The chosen subset is shown in figure 6.4. The indentation shows the hierarchy. The portion is presented as a hierarchical tree in figure 6.3; this is the manner in which it may be presented to the user.

6.3.2 Transferring the Subset into MAVIS 2

The Dewey Decimal classes can easily form the basis of a semantic layer. Each Dewey Decimal class becomes a concept. The hierarchy of the DDC classes can also be easily transferred to the semantic layer. The multimedia thesaurus implementation of the semantic layer in MAVIS 2 allows for two basic relationship types;

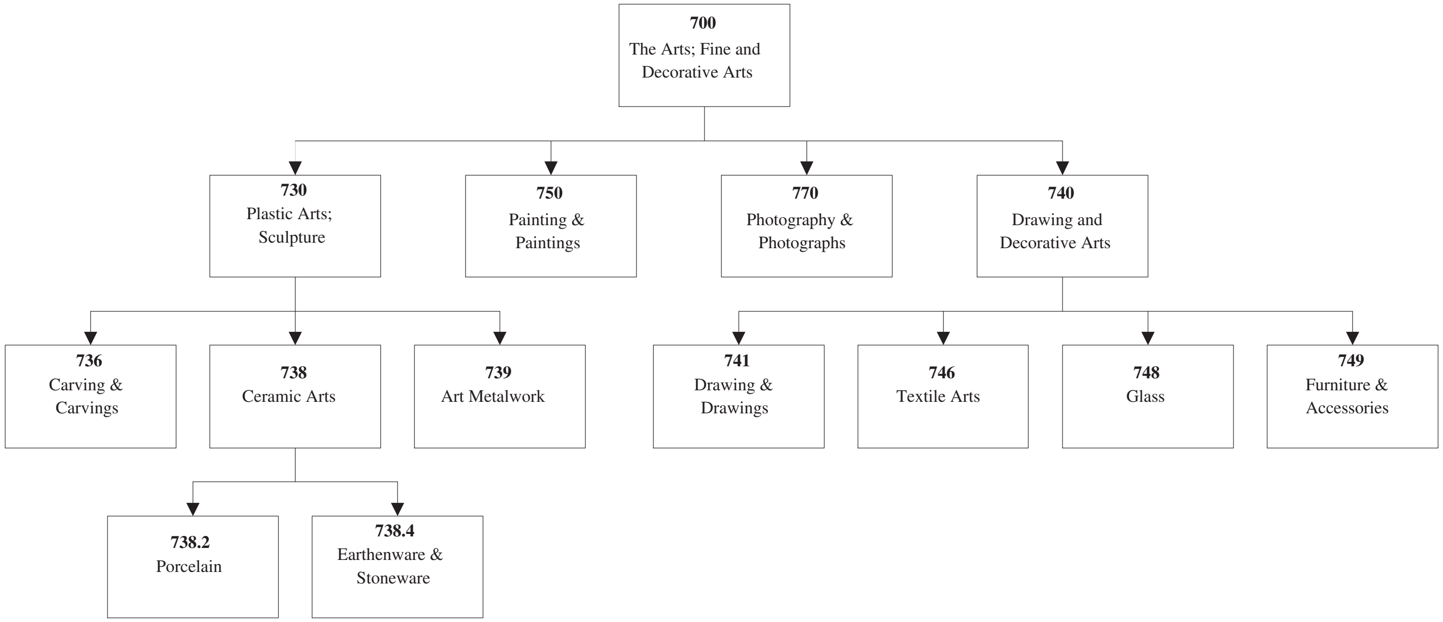


Figure 6.3: Dewey Decimal Classification Subset Hierarchy

700	The Arts; Fine and Decorative
730	Plastic Arts; Sculpture
736	Carving & Carvings
738	Ceramic Arts
738.2	Porcelain
738.4	Earthenware & Stoneware
739	Art Metalwork
740	Drawing and Decorative Arts
741	Drawing & Drawings
746	Textile Arts
748	Glass
749	Furniture & Accessories
750	Painting & Paintings
770	Photography & Photographs

Table 6.4: Dewey Decimal Classification Subset

broader/narrower and *related*. Each class becomes the *broader* concept of more specialised classes, and the *narrower* concept of more generalised classes.

To facilitate displaying to the user, each concept is given a preferred text representation, the corresponding DDC class name. The class names are imported as "anonymous selections"; that is, selections that do not specify some portion of a document, but an independent text phrase (or image feature). These selections are connected to the concepts via the intermediate selection expressions.

Now there is a multimedia collection, and a set of related concepts that covers the subject area of the multimedia collection. The next stage is perhaps the most difficult: The multimedia objects must be associated with appropriate concepts.

6.4 Connecting Images to Concepts Using Latent Semantic Analysis

This section describes a novel method for automatically assigning a set of images to a set of classes, using text metadata with greatly varying vocabulary. Each step of this process is described below, along with information about how that particular step in the process is performed. Following this, images with no text metadata are assigned to concepts by using image features to match with the pre-assigned images. Thus a technique is presented that assigns a set of images to a set of classes, when only a small number of images have text metadata containing greatly varied terminology.

Both the set of museum images and the DDC classes used in this application have associated text metadata. In the case of the images, this takes the form of key-value pairs, and in the case of the DDC classes, sets of keywords.

Unfortunately the terminology does not match up; it is not a simple matter to compare the image metadata to the class keywords to select appropriate concepts. Another technique is needed, that can detect similarity between different terminology.

Section 3.5.5 described a technique called Latent Semantic Analysis, that can identify semantic similarities between words and/or sets of words. Initially, it would seem that since the Dewey class keywords and image metadata comprise a relatively very small corpus of text, the effectiveness of the technique would be poor. However, at Colorado University, there are several "ready-trained" semantic spaces, in a variety of subject areas; these are listed in table 3.1.

Reviewing the subject areas reveals that the *encyclopedia* set holds the most relevant information about museum objects and artefacts; accordingly this is likely to produce the best results.

The LSA technique can, given two sets of text, give a value as to how closely related those two pieces of text are, even if the terminology in each piece of text differs. Thus, to work out which concept is most appropriate to an image, the metadata associated with each image can be compared with the DDC class keywords to give a measure of correlation.

The LSA World-Wide Web site offers on-line access to LSA software. Given two sets of text, and the other necessary parameters such as which document space to use, the software produces the relevant cosine values. A simple tool for producing this value was developed using Java. This tool sends the text to be compared to the LSA WWW site, and receives the result from the LSA software. This happens as a 'batch' process; the sets of text are sent and the results received and written to a file fully automatically.

Thus, the 'knowledge' in a large body of encyclopedia text can be used to correlate class and image keywords. Two problems remain: What part of the metadata should be used for the comparison, and how are the results used to assign classes?

6.4.1 Metadata Field Selection

While a simple approach would be to use all of the metadata associated with images for the comparison, there are two drawbacks:

1. Not all images have the same metadata, and those with more metadata may in general produce a higher cosine value from the LSA process than those with less metadata.
2. The aim is to associate each image with the concept that it is best suited to *represent*. Many of the fields are inappropriate to use, since they do not seem to indicate in any way which DDC class they would best represent. For example, the 'born' and 'died' fields, holding the dates of birth and death of the artist or creator, will not give any useful information about what the image depicts.

The four well-suited fields below were chosen. Each gives useful information about the object that the image concerned depicts. Additionally, referring to table 6.2, it is possible to see that they are also the most common fields that images may have.

- object_category
- object_name
- object_description
- material

The metadata used to calculate a similarity value is a concatenation of these four fields. Since the keywords should be treated as individual terms, and the concatenated fields constitute a passage of information about the object, the term to document method of comparison described in section 3.5.5 will be used.

6.4.2 Interpretation of Results

Using the metadata, keywords and the LSA tool, cosines can be calculated indicating the degree of similarity between the images and the DDC classes (and hence, the concepts in the semantic layer). The next problem that must be tackled is, how are these values used to assign images to classes? If the semantic layer technique is to significantly reduce the effort involved in construction and use of a multimedia information system, its construction must be at least semi-automatic, as pointed out in section 4.6. Thus, it is more appropriate to test the effectiveness of the

DDC Class	No. of Image Representations
The Arts; Fine and Decorative	0
Plastic Arts; Sculpture	5
Carving & Carvings	2
Ceramic Arts	0
Porcelain	19
Earthenware & Stoneware	22
Art Metalwork	17
Drawing and Decorative Arts	0
Drawing & Drawings	8
Textile Arts	11
Glass	4
Furniture & Accessories	2
Painting & Paintings	9
Photography & Photographs	7

Table 6.5: Images Assigned to DDC Classes

semantic layer technique when constructed automatically, than when constructed laboriously by hand.

The most effective automatic technique for addressing this problem is to assign images to classes for which the LSA technique gives the highest cosine values. A description of how this conclusion was arrived at is given in appendix A.

6.4.3 Performing the Classification

The classification was performed with 106 images that had associated ELISE metadata. A list of classes with the number of images assigned to each is shown in table 6.5.

The relevant images were associated with the appropriate concepts in MAVIS 2 using a batch process that sent relevant messages to import the images and create the associations. Of course, one of the disadvantages with the technique used here is that it is not possible to determine which low-level features should be used for matching. In any case, the currently implemented signature modules are rather general in nature, and in the future the Classifier Agent may be left to work out which signatures are most appropriate for classifying images.

Once the main categorisation has been done, the resulting network was browsed through using the concept browser to find and move any images that are obviously out of place. For this purpose, a script that quickly allowed the reassigning of an image to another class was developed. This process of correction took about a quarter of an hour, in which eleven images that had obviously been misclassified were correctly assigned manually.

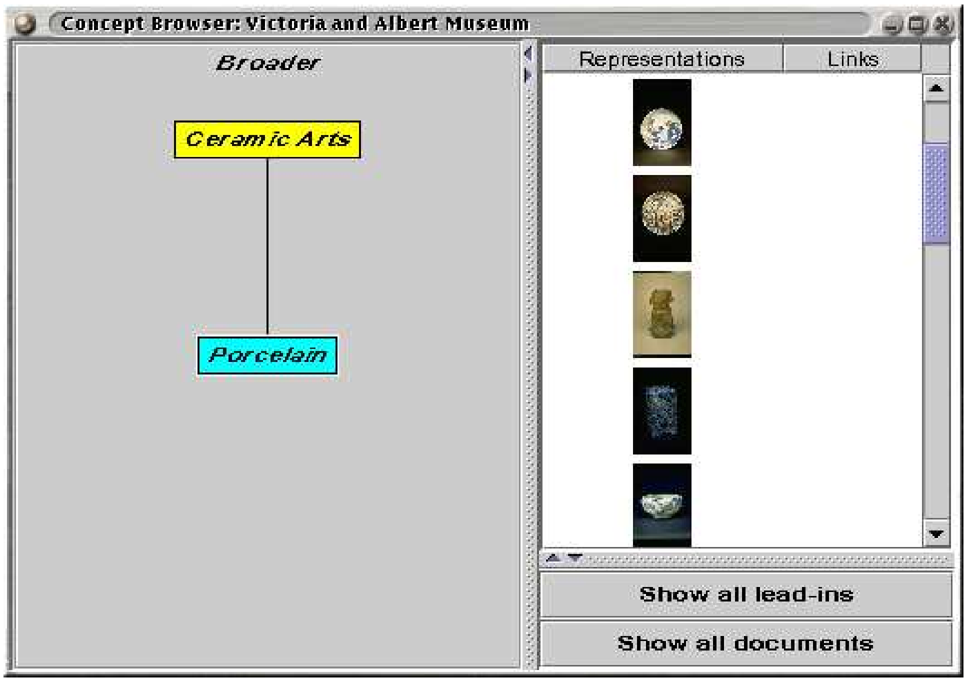


Figure 6.4: Concept Browser with Victoria and Albert Museum Images, and DDC Concepts

The overall time spent generating and verifying the results like this is still a fraction of the time that would have to be spent manually associating every single image with concepts. The improvement in effectiveness is likely to be worth this small corrective effort. The interface and design of the tool used for this correction is likely to have a significant impact on how long this takes.

Figure 6.4 shows the concept browser in MAVIS 2 open at the concept 'Porcelain'. Representations of the concept are shown on the right.

6.5 Completion of the Test Application

In this way, we have created a multimedia collection, consisting of text and images, connected to concepts in a semantic layer (MMT) in MAVIS 2. The way is now clear to test and evaluate many of the techniques that a semantic layer provides.

There are still some techniques that cannot be tested, since there are no hypermedia links or documents indexed on concepts:

- Indexing documents
- Link augmentations

No concepts in the application we have created have any associated *relevant documents*, neither are there any hypermedia links. The following sections describe how these shortcomings are addressed so that these techniques may be evaluated.

Dewey Class Information

Dewey Decimal Code
738.2
Class Name
Porcelain
Description
A hard, fine-grained, sonorous, nonporous, and usually translucent and white ceramic ware that consists essentially of kaolin, quartz, and feldspar and is fired at high temperatures
Keywords
ceramic porcelain decorate underglaze

Figure 6.5: Sample Description of a DDC Concept

6.5.1 Indexing Documents

A relevant document of a concept is something that is usefully associated with the concept, but not actually representative of the concept, or useful for matching in order to reach that concept. The process described thus far has assumed that the images in the collection are representative of the concepts with which they have been associated; in order to test the relevant document mechanism in MAVIS 2, some other media objects must be included in the application.

The information directly concerning the collection of images from the Victoria and Albert Museum and the DDC semantic layer is:

- The images
- The image metadata
- The DDC classes
- Descriptions of the DDC classes
- DDC Keywords

Of the available information, only the image metadata, DDC class information and keywords are not already in the system. The image data is associated with specific images, and features information peculiar to the image, and not generally to do with the concepts. This leaves the DDC class descriptions and keywords.

Thus, with each concept, an HTML document describing the concept is associated as a *relevant document*. An example description is shown in figure 6.5.

6.5.2 Link Augmentation

To test how useful the semantic layer is at supporting hypermedia navigation, there must be some hypermedia links in the first place. The only data available that

Source Anchor	Destination
"Furniture & Accessories"	http://ncnet.com/ncnw/furn-lib.html (Furniture Library Web Site)
	http://www.textilemuseum.com/ (Oriental Garment)
"Glass"	http;//www.encyclopedia.netnz.com/ cutglass.html (Glass Encyclopedia)
"Sculpture"	http://www.sculpture.co.uk/ (British Sculpture Site)
	http://www.vam.ac.uk/vastatic/microsites/ history/intro.html (Victoria and Albert Museum Photograph Exhibition Information Page)
"Carving"	http://www.mercury-gallery.co.uk/ blume/index.html (Carving Exhibition)
	http://www.tate.org.uk (Tate Gallery Web Site)

Table 6.6: Hypermedia Links in Museum Application

is not already included in the application is the metadata associated with the images. However, this metadata is obviously very specific to an image, so a generic hypermedia link is not appropriate. Some other links are therefore needed.

Another feature of the MAVIS 2 system is the ability to author links between objects over which the system has no control. Thus, links can be created pointing to a variety of related websites. These links are listed in table 6.6.

6.6 Classification of the Remaining Images

In order to determine the viability of the technique, the remainder of the Victoria and Albert Museum images were classified using only image features. No text metadata associated with either the images or the concepts were used.

To facilitate this, a "batch classifier" tool was developed. This tool is given a number of media objects to classify. It achieves this by submitting queries to the

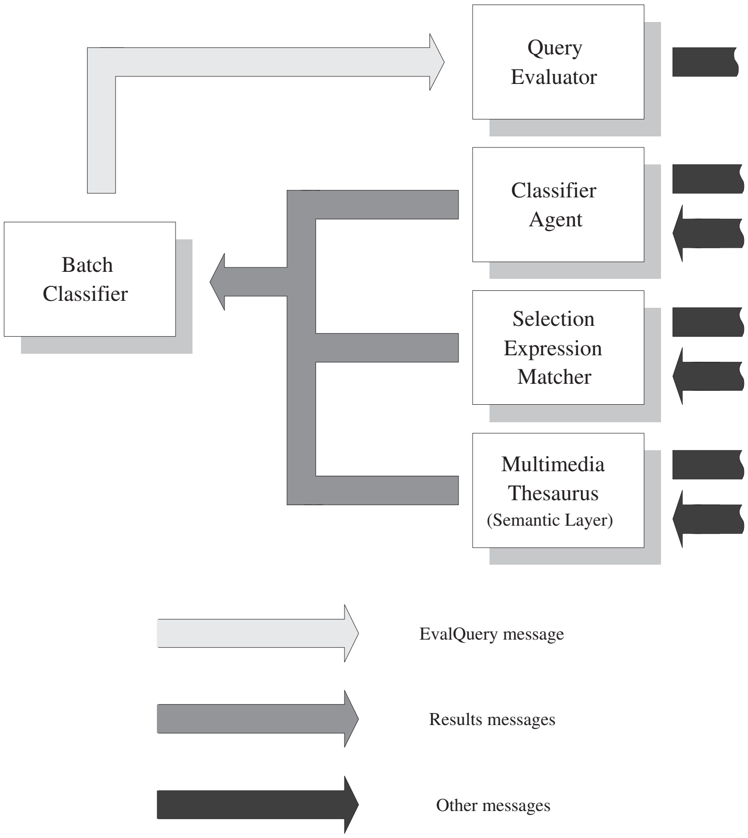


Figure 6.6: The Batch Classifier Process

MAVIS 2 system, and waiting for the results generated by those queries. From those results, it uses the same result sorting code as the *Results Viewer* process, described in section 5.5.3, to find out to which concept an object would best be associated. This process is shown in figure 6.6.

The tool, when executed, registers with a running MAVIS 2 broker to receive *Result* messages. It then starts sending relevant *EvalQuery* messages to the *Query Evaluator*. The query object is classified by the relevant components within the MAVIS 2 system, and when the corresponding *Results* messages are received, the result sorting code from the *Results Viewer* is used to determine a best-matching concept. This concept is then written to a file, which can then be used by another batch tool to create the associations.

This is in effect a *nearest neighbour* type of classification, as described in section 3.4. Those images classified using the latent semantic analysis technique are taken to be the 'ground truth' against which other images are compared.

There was a problem of overloading the MAVIS 2 system with hundreds of concurrently-running queries. How many such queries the system could handle at any given instant depended on the capacity and power of the system(s) the components were running on. Accordingly, the Batch Classifier limited the number of queries running at any one time to a sensible number. The operation took around three hours to classify nine hundred images. Since it required no user interaction, and the code is not yet fully optimised, this is an acceptable length of time.

The ease with which this tool could be integrated into the system is a testament to the flexibility of the MAVIS 2 architecture. Any number of applications could make use of the system, without necessarily requiring any of the system's user interface components.

6.6.1 Results of the Automatic Classification Using Features

In this case, the automatic process resulted in rather less than reliable results. Many of the images were not classified correctly. While we have no access to a *comprehensive* "ground truth" (which would render this process unnecessary), in most cases it is possible for a human to tell heuristically whether or not the classification an image has been given is correct, given the relatively non-specialist nature of the subject matter.

The batch classification had a high rate of success in identifying *Glassware* and *Furniture & Accessories*. Figure 6.7 shows some of the images correctly identified by the batch process as examples of glassware. Images depicting paintings, however, were particularly prone to misclassification. This is because the paintings in the collection vary greatly visually.

The problems with this instance of automatic classification based on features can be summarised thus:

- There are not enough examples in each class for robust classification to take place. For a good result, all of the images being classified must be visually more similar to at least one of the example set of paintings than to a representation of another concept. Hence, since paintings vary greatly, it follows that there must be a large number of example paintings already connected to the multimedia thesaurus for a good classification. Unfortunately this is not the case.

  Additionally, the classification used a rather simplistic 'nearest neighbour' approach; more sophisticated techniques exist that may produce better results, such as decision tree classifiers.

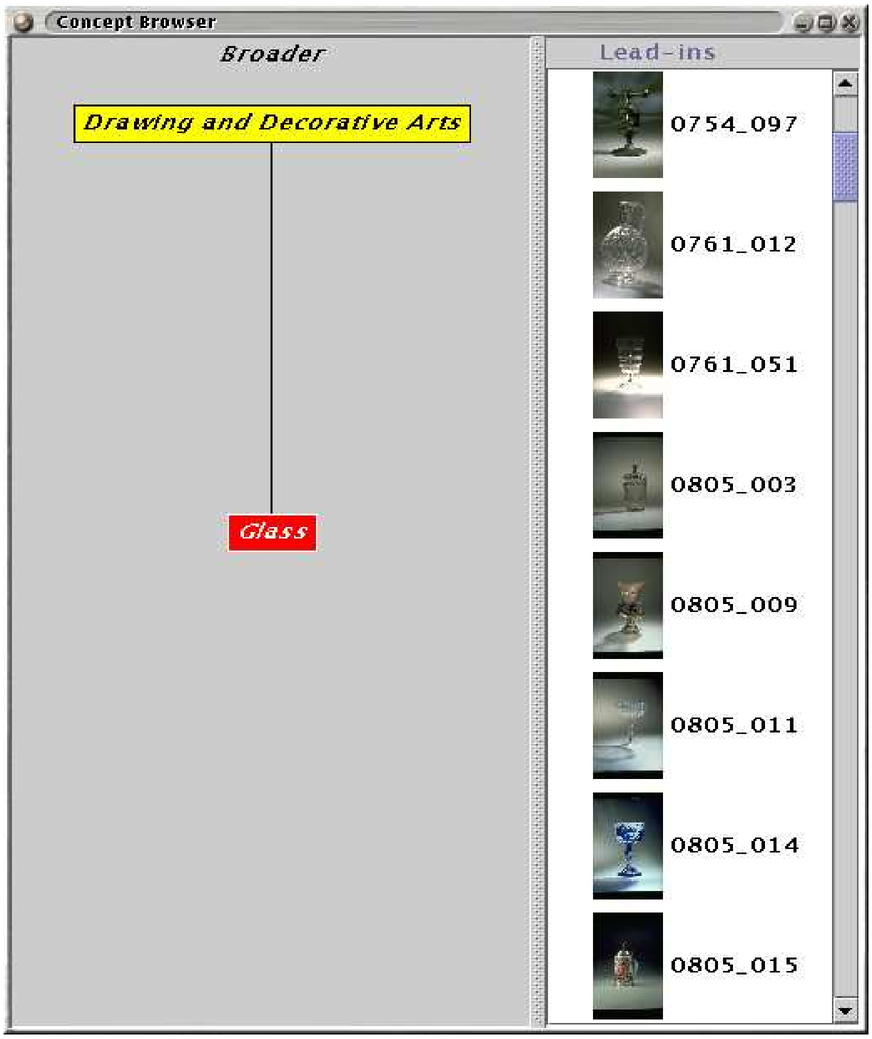


Figure 6.7: Correctly Classified Examples of Glassware

- The QBIC features used for the classification are rather generic. To classify robustly, the features must be able to discriminate between (for example) paintings and non-paintings. The features QBIC uses are colour, and some crude measures of texture. It is obvious that colour is not a good discriminating measure for identifying paintings, since paintings come in many differenct colours. While a specialised texture module may be able to pick up subtle features such as brush strokes, the generic QBIC texture feature is not sufficient for this.
- The classification used the features of each whole image in its entirety. Many of the images in the collection have black borders, for example the top left and bottom right images shown in figure 6.1. Such images tended to be found very similar in terms of QBIC features to other images with black borders, regardless of what was in the centre of the image.

An automatic segmentation technique could detect and remove those black borders before classification took place, which is likely to cut down misclassifications caused by this. More sophisticated techniques may also be able to segment the objects depicted in images whatever the colour of the background.

While this procedure was not entirely successful for the reasons cited above, it demonstrates that such a procedure is feasible within the semantic layer architecture. In future applications, more specialised domain-specific feature types will result in more robust classifications.

Even in the case where the automatic classification is not perfect, it can still be used as a starting point. With a suitable tool, a human user can verify and correct the results.

However, since in this case the performance was low, the decision was made to leave the incorrectly classified images out of the application, as their presence would reduce the performance of the system. Although there are relatively few representations of each concept, there are enough to be able to demonstrate the semantic layer technique.

6.7 Evaluation

The methodology described in this chapter constitutes an evaluation of one aspect of the semantic layer technique — the construction process. The feasibility of the technique has been demonstrated.

Every step of the construction has been automated, with the following exceptions:

The selection of the DDC subset. While this required some thought, it required only selecting part of an existing classification system. The scope could easily be extended, and so the approach used is scalable to larger systems.

The selection of appropriate image metadata fields. Even if the number of images had been many times larger than the set used in this application, the number of fields for each image would remain the same. The size of the dataset does not affect the amount of time this step of the construction process takes.

The selection of the LSA semantic space to use for comparison. This requires only a review of the subject areas of the semantic spaces, and is again unaffected by the size of the multimedia collection in the application.

The classification of images based on image features. Initially, this may be performed wholly automatically. Unless the features used for the classification are sufficiently robust, it is likely that some user interaction will be required to correct any errors, but this effort is still small relative to the effort required to classify the images from scratch by hand.

All other stages have been fully automatic. While they may take a significant amount of computation time, the computation can be left unattended and requires little human effort. Thus it can be seen that the approach to multimedia thesaurus construction here is scalable to larger sets of multimedia data.

MAVIS 2 is an open system, so the techniques described here are likely to be a small subset of a plethora of available methods for constructing a multimedia application with a semantic layer component. The steps described here demonstrate that it is possible to create such an application without a prohibitively large human effort.

6.8 Summary

This chapter has described how a multimedia application suitable for testing and evaluating the semantic layer technique has been constructed.

The multimedia application holds a selection of images depicting objects owned by the Victoria and Albert Museum. Many of these (approximately half) have associated metadata, produced as part of the ELISE project, in the form of records. The fields in each record do vary somewhat, but the majority have a set of four core fields, which were used for the initial association of images with concepts.

The Dewey Decimal Classification was chosen as the basis for the semantic layer, since it has been extensively researched and is widely used. This alleviates the need to create concepts and relationships from scratch. A subset of the Dewey Decimal Classification hierarchy suitable for the multimedia collection was chosen. Many of these classes have associated keywords, also produced as part of the ELISE project.

The terminology in the metadata fields and the DDC class keywords did not completely match up, so a means of correlating the terms other than straight matching was needed. The Latent Semantic Analysis technique learns associations between words from large corpora of text. An LSA system running at Colorado University has been trained using an extensive range of encyclopedia texts. This system was used to estimate similarities between DDC class associated keywords, and the image metadata, in order to work out which class each image should be placed in.

A best-fit algorithm was chosen, after it produced the best results in a small test sample. In this way, a semantic layer with associated media objects was constructed. To complete the application, some relevant documents and links were added, to enable testing of the indexing and hypermedia features of the semantic layer.

Some automatic classification of the remaining Victoria and Albert museum images was attempted based on QBIC features. Whilst in this case, due to the lack of suitably robust image features, the results were not of a high quality, the feasiblity of such an approach was demonstrated. Automatic classification of image collections is a well-researched task, and we have shown that it is possible to 'plug in' such an approach in order to establish an application.

Finally some heuristic evaluation of the construction process was given. The process described by this chapter was demonstrated to be scalable to large sets of data. This demonstrates that the construction of large applications with semantic layer components is feasible without requiring extensive human effort.

Chapter 7

The Application in Use

7.1 Introduction

So far, the semantic layer technique has been described, a system implementing a semantic layer in the form of a *multimedia thesaurus* has been introduced, and a multimedia collection has been imported into the system and connected to a small semantic layer. It was demonstrated that the techniques used to construct the application are scalable to larger sets of data and concepts.

However, the question "is it useful?" still remains. Is the effort taken to construct such a multimedia thesaurus worthwhile? This chapter attempts to answer these questions.

van Rijsbergen poses three questions concerning evaluation (van Rijsbergen, 1979):

1. Why evaluate?
2. What to evaluate?
3. How to evaluate?

The answer to this first question is, as van Rijsbergen points out, largely social and political; in this case it is to find out whether or not the semantic layer technique is a technique worth pursuing further.

In the context of this work, the answer to the second question is, put simply, "the semantic layer technique". The previous chapter dealt with evaluating the construction process; this chapter focuses on the use of the resulting application.

We are not attempting to evaluate the MAVIS 2 system itself, nor make any comparison between the MAVIS 2 system and other retrieval or navigation systems. While other systems do have similar facilities, unlike the MMT, the vast majority operate only on text. The few that do have significant *content-based* support for media require a large, unscalable amount of human effort during the construction of applications. Examples of these are *MACS* and *Himotoki/COIR*, described in sections 3.5.2 and 3.5.4 respectively. As was demonstrated in the previous chapter, a multimedia thesaurus application can be constructed semi-automatically using scalable techniques.

One system that did use image features in a thesaurus context was Ma's texture thesaurus (Ma & Manjunath, 1998), described in section 2.2.4. However, the 'terms' in the thesaurus are *visually* similar, not necessarily *semantically* similar—they are just clusters of similar images.

Thus, since there are no directly equivalent systems with which to compare performance, it is purely the benefits that the semantic layer bring that are the focus of this work.

In the case of MAVIS 2 and the multimedia thesaurus, the third of van Rijsbergen's questions is not simple to answer. Evaluation of information retrieval systems is a well-established area; there are several techniques for evaluating retrieval effectiveness. These cannot be applied directly to an evaluation of the semantic layer technique in MAVIS 2, however, because it is the retrieval effectiveness of the current *signature modules* that would result.

Evaluating hypermedia is not a simple case either. As was seen in section 2.3.6, most existing hypermedia evaluation methodologies rely on user trials. User trials are not a practical option at this stage in this research, since such trials inherently include an evaluation of the *interface* to a system. Garzotto's methodology also incorporates evaluation of a system's interface. How the existence of links are indicated, how the user can tell whether the link is worth following or not, and how following a link updates the display are amongst several factors that affect the result of trials of a hypermedia system. Since the semantic layer idea and indeed MAVIS 2 are new, the interface is still unsophisticated and experimental, and unlikely to match up to the extensively-researched interfaces available in other hypermedia systems.

Additionally, the speed of the system would skew the results of any trials. The MAVIS 2 code has not yet been optimised. It has been designed to demonstrate that techniques are possible rather than to be an efficient implementation of existing techniques. These issues of interface and speed are likely to get in the way of users' ability to concentrate on the advantages of the semantic layer itself.

Another factor is the relatively small size of the dataset in the test application. Typically, text information retrieval evaluation datasets involve thousands of documents. There are many standard queries for which the ideal results are known, and these can be used as a benchmark for evaluation (Smeaton & Harman, 1997). Such collections do not exist for the field of multimedia retrieval (Smith & Li, 1998). The museum application used for testing in MAVIS 2 involves only around a hundred images connected to a handful of concepts. The application is far too small to obtain reliable empirical measures.

Therefore, this thesis has taken an heuristic approach to the evaluation of the multimedia thesaurus technique. A variety of scenarios of the use of the museum application are described, and the way that the multimedia thesaurus can be employed is explored. In this way it is possible to show the benefits that a multimedia thesaurus, and hence a semantic layer, bring to a multimedia application.

7.2 Scenarios

In the following sections a variety of scenarios of use are described. For each scenario, the aim of the user is given. The process by which the user achieves this aim without the aid of the multimedia thesaurus is described. This is followed by a description of how the user might achieve their aim with multimedia thesaurus assistance. At the end of each section is a discussion of the differences between each mode of use, and the benefits (or otherwise) that the multimedia thesaurus has brought to the user.

7.2.1 Scenario 1

A user has found an image depicting an interesting costume, shown in figure 7.1. The user wishes to know which museums or galleries might have collections containing exhibits similar to this. Unfortunately there is no text metadata associated with the object, nor are there any hypermedia links involving this object. It is simply one of the images in the Victoria and Albert collection for which we have no associated information.

Without the MMT

Without the MMT, the number of available options is limited. The user can proceed in one of the following ways:

- The user can perform a content-based information retrieval operation. This finds images similar to the query, shown in figure 7.2. While it shows that

Figure 7.1: Costume of Interest to the User

there is a very similar image present, this does not help the user find out which museums or collections contain similar exhibits.

- The user can perform a content-based navigation operation. This searches the system for generic hypermedia links, whose source anchors are visually similar to the image. As can be seen by the result set in figure 7.3, no such link exists.
- The user can elect to disregard the image as a possible starting point, and start a free text retrieval query. However, unless they choose the correct terminology, relevant information may be missed. Searches for "oriental garments", "clothing" and even just "garments" all result in no hits. The user must continue searching trying different terminology until something useful is picked up.

With the MMT

Once the MMT is available and switched on it can help considerably. The following options become available:

- The user can perform a content-based navigation query, searching for links with source anchors similar to the query image. Before, this operation yielded no useful results, but with MMT link augmentation switched on, the set of

Figure 7.2: Results of Content-Based Retrieval Query

Figure 7.3: Results of Standard Content-Based Navigation Query

Figure 7.4: Results of Content-Based Navigation Query With MMT Assistance

results shown in figure 7.4 are returned. The first retrieved link is a link to a textile museum, shown in figure 7.5.

The link was picked up because another image connected to the concept *Textile Arts* was linked to the museum site. That image is shown in figure 7.6. Although the image depicts an object semantically close to the object depicted in the original query image, and is in fact quite obviously visually similar to the human eye, the similarity was not picked up by the automatic media matching alone. However, the extra information that the MMT provided allowed the user to find a relevant site with a single navigational step, which was not otherwise possible.

- The user can initiate a classification query. This instructs the MMT process in MAVIS 2 to attempt to find the concept most pertinent to the query image by comparing it to representations of concepts. The results of this query can be seen in figure 7.7. The image has been classified as belonging to the *Textile Arts* class, which is indeed the most appropriate class in the MMT for the image. The link to the Textile Museum has also been retrieved since it is a link from one of the representations of *Textile Arts*.

  From here, the user can choose to browse the concepts in the MMT by double-clicking on the concept in the results viewer. This brings up the *Concept Browser* with the display shown in figure 7.8. From there they can view

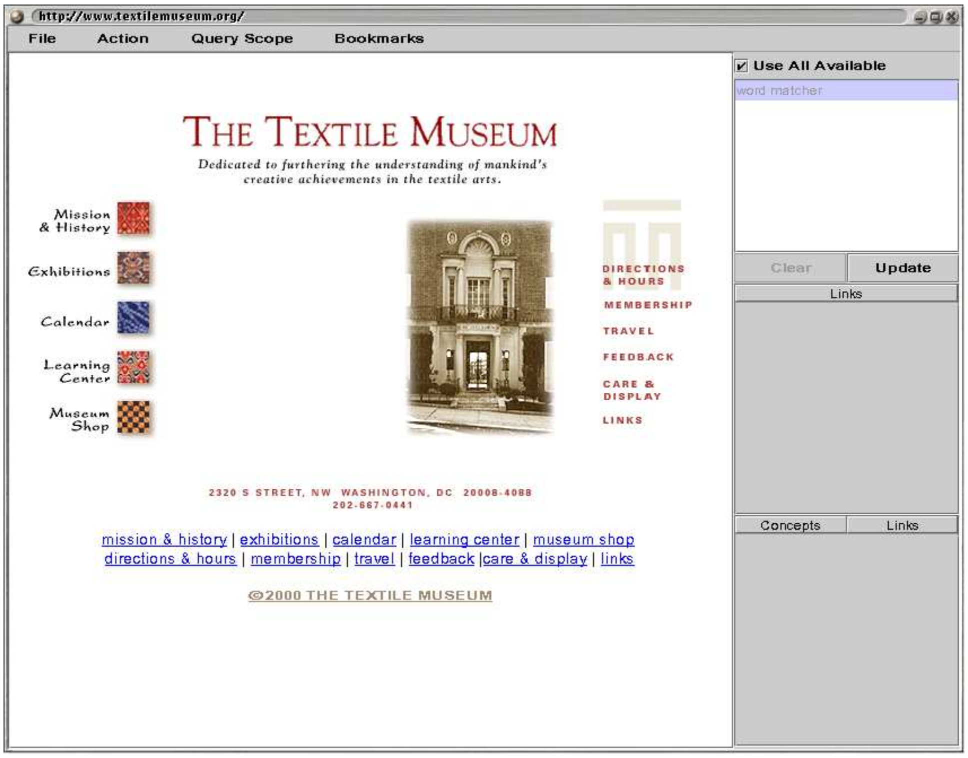


Figure 7.5: Destination of Link Found With MMT

Figure 7.6: Source Anchor of Link to Museum Site

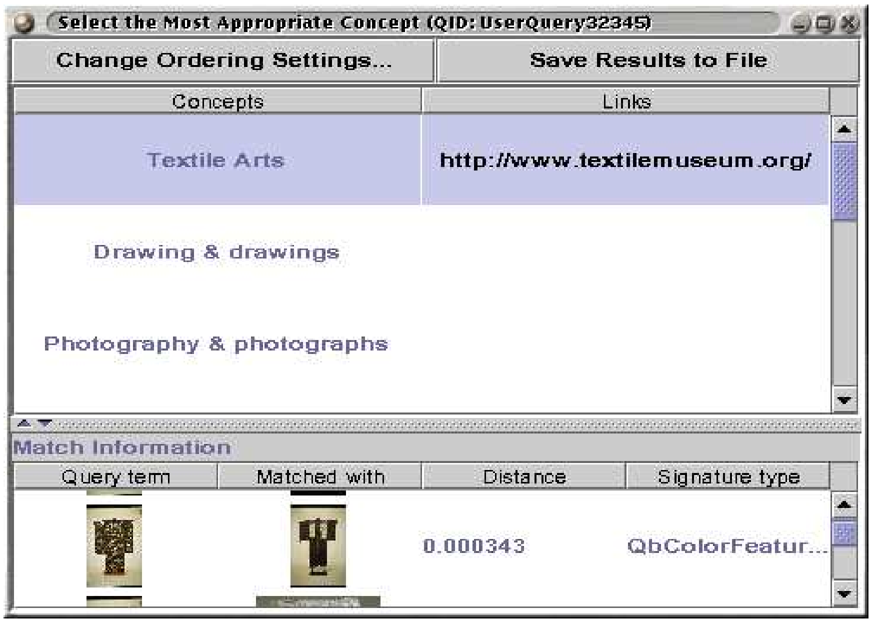


Figure 7.7: Results of Classification Operation

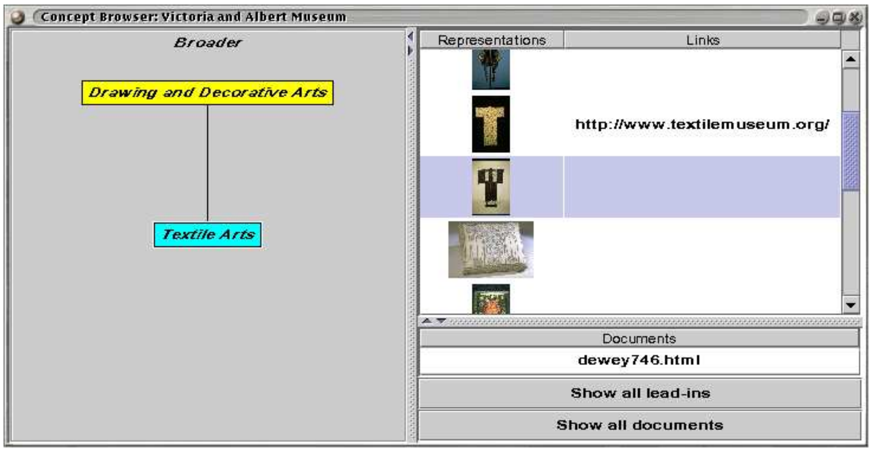


Figure 7.8: The Concept Browser at the *Textile Arts* Concept

other representations of the same concept, or browse around the concepts themselves.

- The user may for some reason choose to start with a text search anyway. If they do, the MMT still maximises their chances of finding something useful. For example if one of the representations of *Textile Arts* is 'clothing', then a search for that term will reach that concept. From there, the Textile Museum site can be picked up, or the concepts in the MMT explored as described above.

Thus, it can be seen the the Multimedia Thesaurus has provided a wealth of extra navigation possibilities to the searcher. The system has made use of the media data (in the *plane of expression*) to bridge the gap to the semantic layer (in the *plane of content*). Once in the semantic layer, the associations between concepts and the media itself can provide the searcher with useful information that was very difficult to reach by using image and text searching techniques alone.

In other words, the system was able to provide information about an isolated image that was previously unknown to the system, and that had no associated metadata or textual information. This represents a significant step forward from where previously images had to be assigned text metadata in order to be usefully searchable.

7.2.2 Scenario 2

In many situations, the user may wish to find an image depicting an example of a particular type of object. For example, a student may wish to view some different examples of wood carvings before writing a dissertation on the subject.

Consider the case in which the student has no examples of carvings to hand. They cannot use a 'query by example' style of content-based retrieval, nor use content-based navigation to search for links.

Without MMT

There are a number of ways in which the user can proceed, if no MMT is available:

- If a 'scratchpad' mechanism is available, such as the QBIC colour layout interface described in section 2.2.4, they may try to sketch out what they *think* a carving might look like. This is not a simple task. Carvings are likely to be of a largely uniform colour. The subtle changes in shading and the surface texture are impossible to represent using the rectangular shapes that the QBIC colour layout interface provide. The user's best hope is to give some estimation of the general layout of an image containing a carving, which may not coincide with what carving images in the system actually look like.

  As an example, a guess consisting of a wood-colour square on a white background was used as a query to QBIC. The results of this query are shown in figure 7.9. The query image is shown at the top of the window.

  The first retrieved image does at first glance look like it may contain carvings, but turns out to contain sculpted ceramic figurines. The images below this obviously do not depict wood carvings.

  It may be that with some trial and error, the user is able to find an example of a carving, and to use that to start further queries. This may be adequate if the user only wishes to find a single example of a carving, but the user may

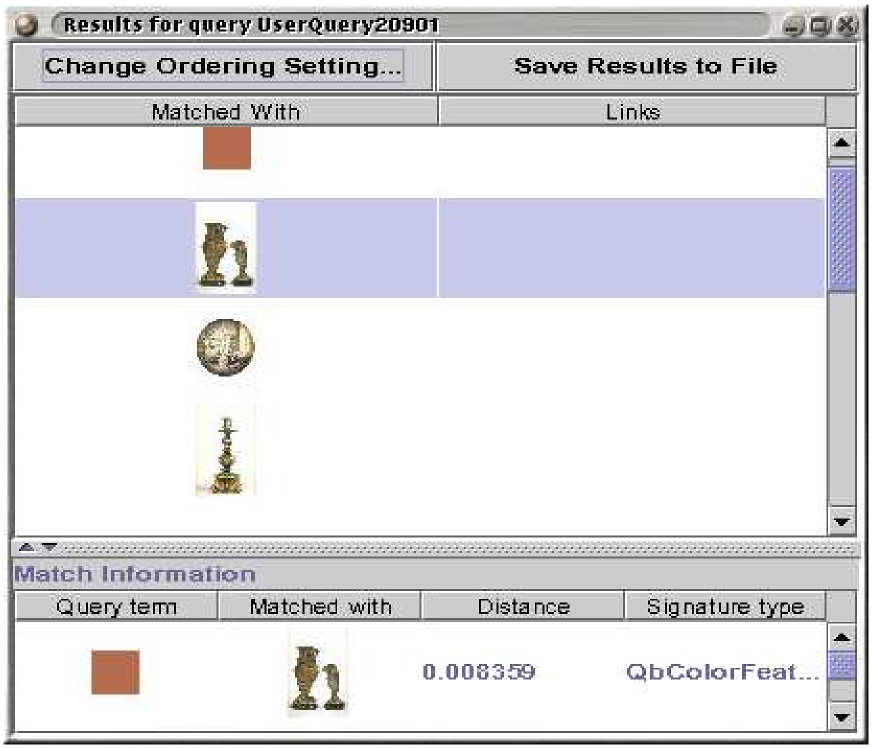


Figure 7.9: Results of Sketch Query

wish to view some different types of carving. These different types are unlikely to be similar enough visually for a simple content-based retrieval operation to be effective.

Thus, it is *possible* for the user to accomplish their aim in this way, but it is not an ideal method, and may require a degree of time and patience.

- The user may start a text query, using the word 'carvings' as a query term. This may retrieve text documents containing the word 'carvings', which may contain (links to) images depicting examples of wood carvings.

  In the museum application, there is no text associated with the wood carving images. The associated metadata does not use the term 'carving' explicitly, so a simple text search does not retrieve any useful information. Even if a document was found, there would be no guarantee that it would contain a useful image. The user may have to try a variety of queries and view a large number of documents before they find a suitable example.

- The only other alternative is to browse around using normal hypermedia links in the hope of finding an example. This could take an indefinite amount of time and effort, and is obviously an unacceptable solution.

With the MMT

Once the multimedia thesaurus is made available, the number of possible ways to find an example is significantly increased:

- A text search is much more likely to retrieve suitable *images*. The user can submit the text query 'carvings' and turn on the multimedia thesaurus *query expansion* facility. The system can use the text matching features to determine that the concept *Carving & Carvings* is the most appropriate concept pertaining to the query. It then performs parallel queries using other representations of that concept, some of which are images.

  The top results of that query are shown in figure 7.10. The query has provided a variety of wood carving images that are not all visually similar. For instance, the second image from the bottom depicts a carved wooden chest, even though it looks significantly different to the small carved figurines in other images.

  Not all of the retrieved images are directly associated with the concept *Carving & Carvings*; some images were retrieved because they were visually similar to a representation of that concept. In this way, the operation has retrieved a variety of visually different images from a text query. Not all of the images depict wood carvings, though the topmost results certainly do, since they are directly connected with the concept *Carving & Carvings*.

  Precision is not perfect but is much higher than for a single 'query by example'. The results of a simple query by example are shown in figure 7.11. Even though the retrieved images look similar, the number of actual wooden carvings retrieved is three, with four non-carvings. The precision of the MMT-enhanced query compares favourably, retrieving five carvings and only two non-carvings in the top seven ranked objects.

  In summary, the MMT has enabled the user to retrieve a variety of images depicting carvings from a single text query, even though some of the images were not explicitly associated with the concept *Carving & Carvings*.
- The user can also perform a text query as a *classification* query. It is trivial for the system to recognise that the query pertains to the concept *Carving & Carvings*. From there, the user can open a concept browser and view the representations of the concept there, as can be seen in figure 7.12. This in itself provides a number of examples of carvings. The user may also use the

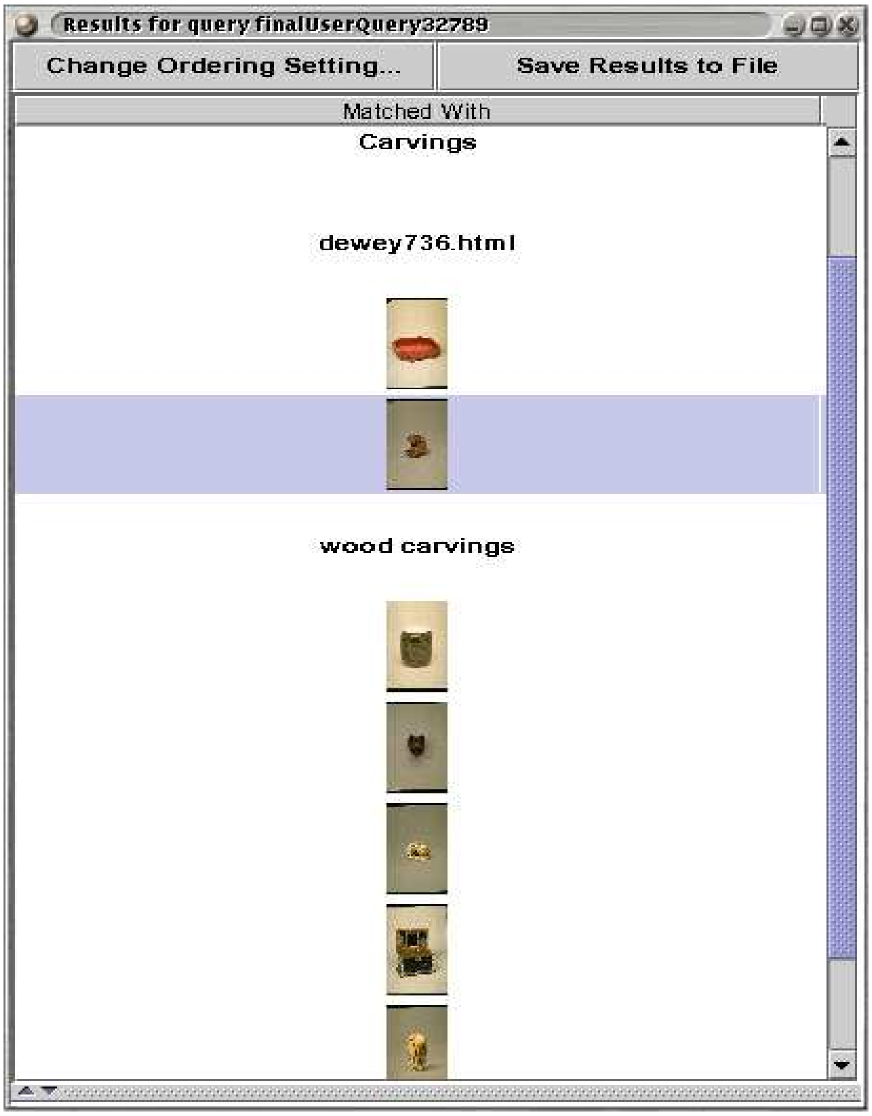


Figure 7.10: Result of Text Query With MMT Query Expansion

representations as examples with which to find visually similar carvings which are not directly associated with the concept.

- The MMT also allows the user to start searching without requiring an initial piece of text or image at all. The user can open the MMT concept browser at the top level or *root* concept, shown in figure 7.13. With only two navigational steps, the user can reach the concept *Carving & Carvings* and see images depicting examples of carving, without actually having had to specify anything to search for.

Thus, the MMT has provided a new solution to the problem facing many searchers, "Where do I start?" Provided that the concepts in the MMT are organised sensibly, so that it is obvious to a user down which path of concepts they should proceed, the MMT provides a high-level means of reaching

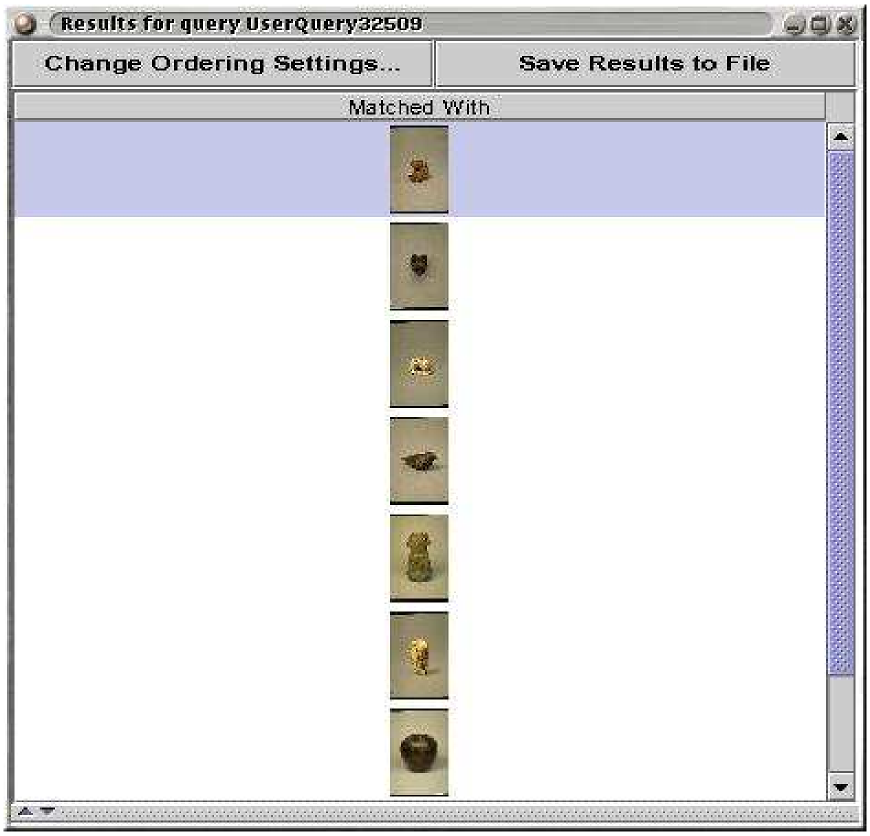


Figure 7.11: Results of Simple Query by Example

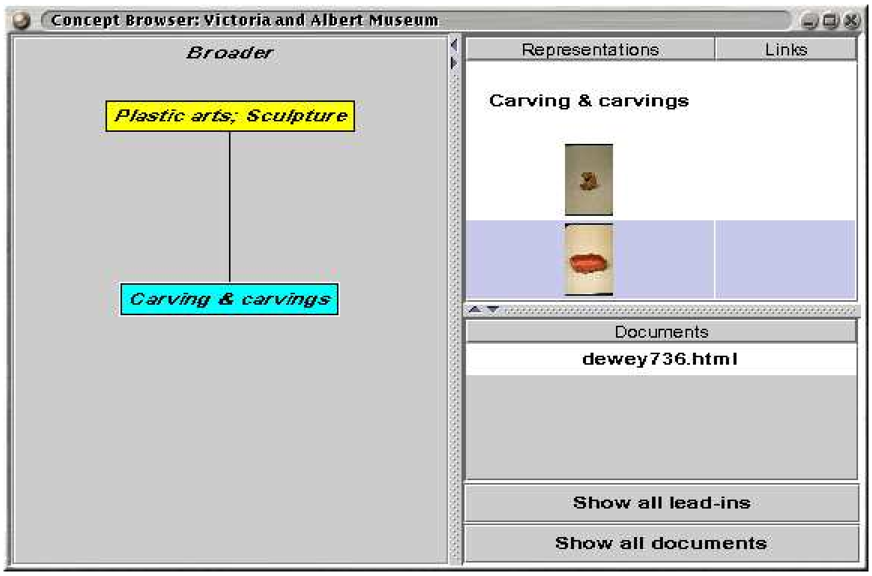


Figure 7.12: The Concept Browser at the Carving & Carvings Concept

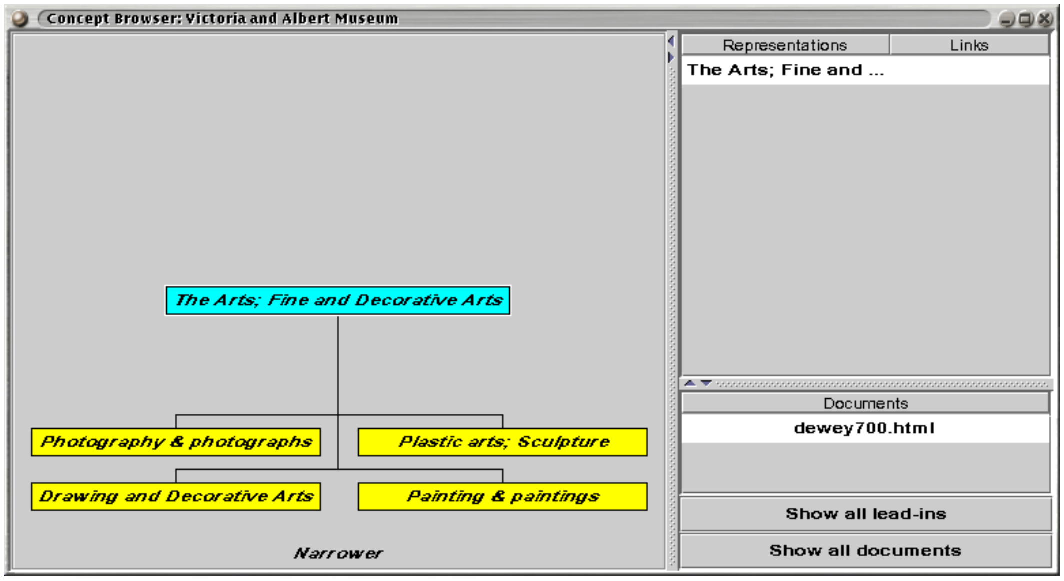


Figure 7.13: The Concept Browser at the Root Concept

Figure 7.14: Query Image

examples and information about a particular subject in different media.

From these points it is clear that the MMT enriches the possibilities for searching when one has no suitable example to hand. As well as augmenting existing retrieval techniques, the MMT provides an alternative method of starting a search, using the root concept of the semantic layer.

7.2.3 Scenario 3

The ubiquitous example of use of a content-based image retrieval system is to ask it to 'find me objects similar to this one'. In this scenario we consider how the MMT might assist in this task.

Consider a user who has found a porcelain dish, shown in figure 7.14. They wish to find similar porcelain dishes to this.

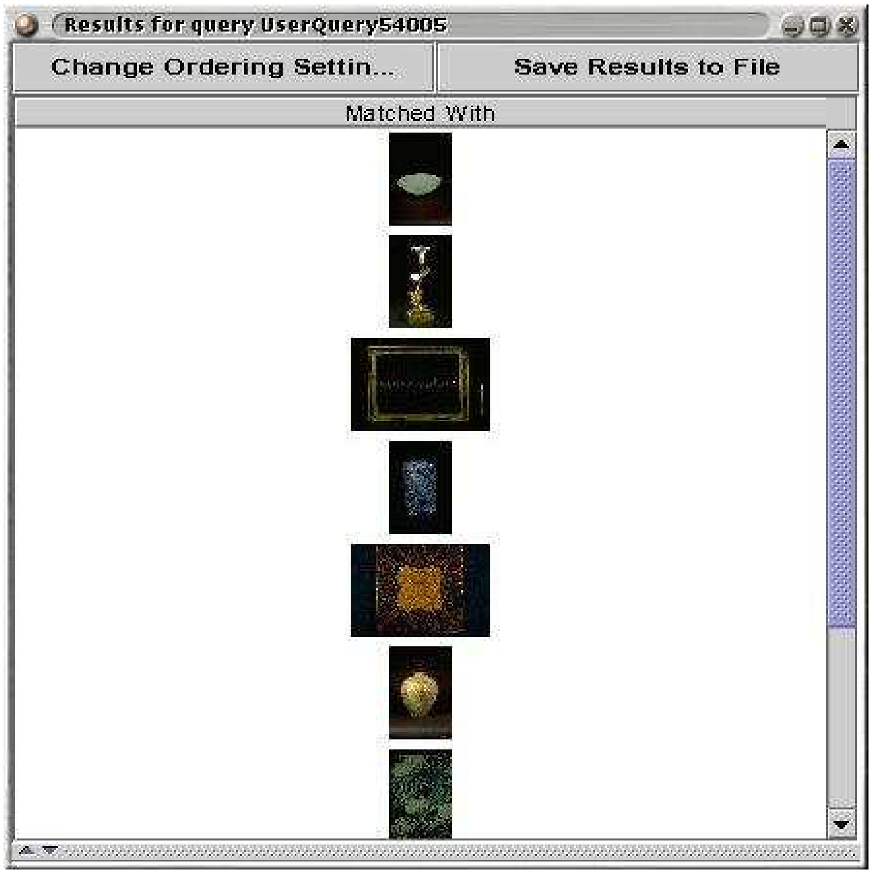


Figure 7.15: Result of Standard 'Find Similar' Query

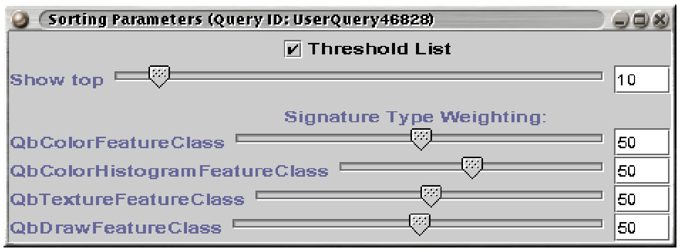


Figure 7.16: Signature Weightings Dialogue

Without the MMT

Without the MMT, the user may of course submit a 'find similar' query to the system. MAVIS 2 and QBIC return the set of images shown in figure 7.15. It can be seen that not all of the images in the result set show porcelain dishes, in fact only two of the retreived images contain any porcelain object at all. The user can slightly reorder the retrieved results by altering the weightings of each of the QBIC feature classes, using the dialogue shown in figure 7.16, but this requires time and effort and does not necessarily produce effective results.

The only other alternative is to start a text query, which may well be unsuccessful for reasons discussed in the previous scenarios.

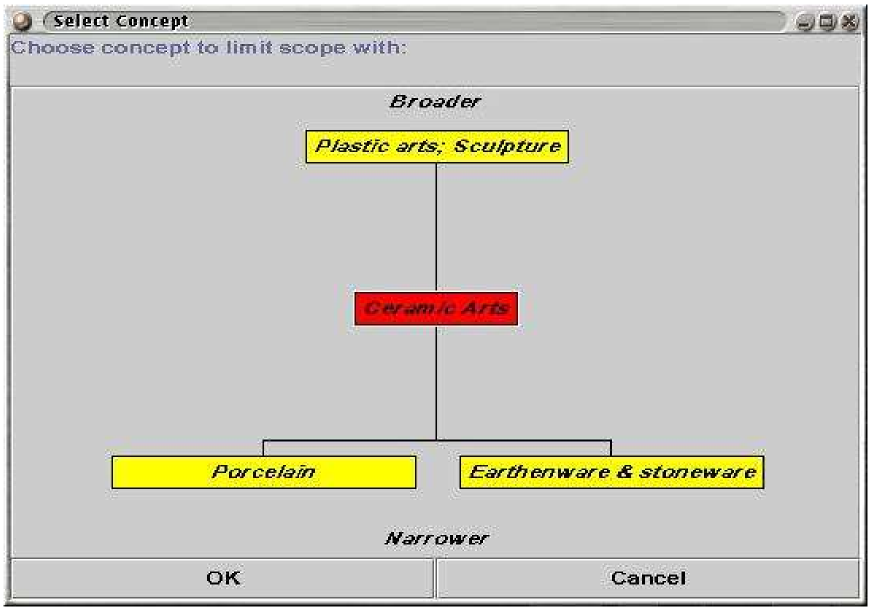


Figure 7.17: Selecting Broadest Concept for Scope Narrowing

With the MMT

The MMT offers a means of improving the effectiveness of retrieval: The user can switch on the *Limit query's scope with MMT* option, and select a branch of the concept layer to include in the search. The dialogue shown in figure 7.17 is used to select the broadest concept that is relevant to the search. During the query, only representations and documents associated with that chosen concept and narrower concepts are retrieved. The rest are excluded as described in section 4.3.3.

The user can select a suitable concept, *Ceramic Arts*, using just two mouse clicks. They then submit the "find similar" query. With the scope narrowing switched on, the query retrieves the set of results shown in figure 7.18.

The results show that there is now a very similar porcelain dish third from the top of the list, and three porcelain plates also appear in the results, which may also be of use to the user, being very closely related.

The only disadvantage of this approach is that only objects explicitly associated with the semantic layer will be retrieved. However the set of retrieved images is a much better set of results than retrieved by the straightforward query without the MMT. One very useful result is provided as well as three possiblities. The retrieval without the MMT resulted in no useful hits in the top six retrieved objects.

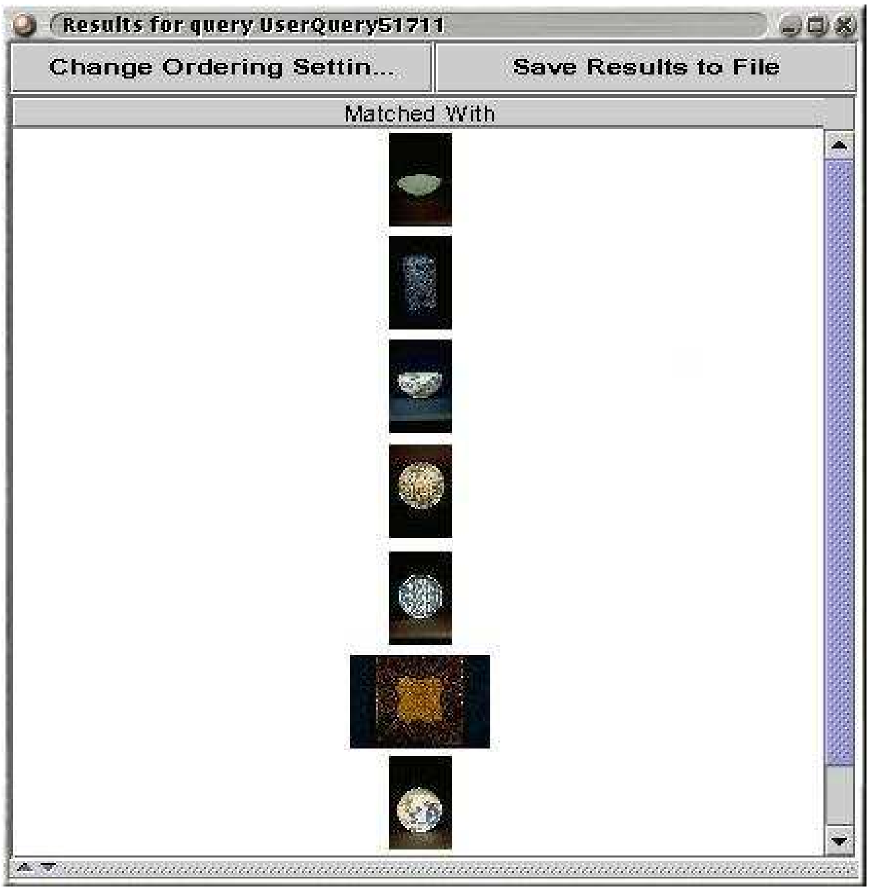


Figure 7.18: Results of Query With MMT Scope Limiting

7.3 Summary

This chapter has detailed some heuristic evaluation of the multimedia thesaurus implementation of the semantic layer technique. Three scenarios of typical multimedia information system use were given:

1. Find out about an object depicted in an image.
2. Find an example of a type of object.
3. Find images similar to a query image.

In each case, the possible strategies for completing the given task were explored, both with and without the aid of the MMT. In each case, the MMT was found to provide a richer set of possible strategies, and to improve the effectiveness of retrieval and navigation considerably.

In this way we have shown that the multimedia thesaurus technique, and hence the semantic layer technique, have the capability to enhance the effectiveness of a multimedia information system. The next chapter discusses the implications of this, and outlines some directions for future research in the area.

Chapter 8

Conclusions and Future Work

8.1 Reflection

This thesis has reviewed existing multimedia information access techniques, and advocated a new technique for addressing some of the problems with information access. The semantic layer technique has been implemented as a multimedia thesaurus in the MAVIS 2 multimedia information system. An application has been built within the architecture using scalable techniques, and its use demonstrated.

The closest related work is by Hirata *et al.* at NEC (Hirata *et al.*, 1996; Hirata *et al.*, 1997), but that work left many open questions, such as how the conceptual relationships are initially constructed, and how media representations are associated with conceptual representations. Performing those tasks manually would be prohibitively laborious. This thesis has provided an exploration of the issues involved in building and using an application with a semantic layer.

This chapter reflects upon the findings of the thesis, and suggests some future directions in which the work could be taken.

Chapters 6 and 7 included some evaluation of the feasability of the technique and how useful it is in practice. This chapter evaluates the work in a broader context. In order to achieve this, we must return to the original aims of the work, and judge the success of the work in fulfilling those aims.

Chapter 1 listed four issues with current multimedia information systems. The impact this work has had on those four issues is discussed in the following sections.

8.1.1 Manually Associating Metadata

In many cases, the task of manually assigning text metadata to images in a collection is lengthy, laborious and impractical. What constitutes impracticality for each application is likely to vary according to economic and political factors. In the case of this research, manually connecting images to concepts in the museum application was not a practical option.

Chapter 6 described how the museum application was built. Some images in the image collection already had associated metadata; however this was incomplete, and inconsistent in that different types of metadata were available for different images. However, the metadata present was enough to establish a small set of images with a semantic layer, with the aid of the Latent Semantic Analysis technique. It was necessary for a human user to go through and verify and correct the results of this procedure, however the time required for this is still a small fraction of the time it would have taken to assign all of the images to concepts from scratch.

In cases where no metadata exists at all, a number of the images would have to be connected to concepts by hand; there is no way round this, but only a subset need be connected manually.

The small set of images connected to concepts generated by the LSA process constituted a ground truth, which was then used as a basis for automatically classifying the rest of the images based on their visual features. In this instance, the results of this procedure were not wholly satisfactory, since the features used for classification, the QBIC features, were not robust enough. However, in some instances, particularly in the cases of *Glassware* and *Furniture & Accessories*, most images were classified correctly.

Successful classification, then, relies on the availability of features suitable for a particular application. Once these are found, large image collections can be automatically classified in a scalable way. In many cases, there may be a suitable existing classification technique available. In other cases, finding suitable image features with which to classify a very large collection is likely to take less time than manually assigning metadata to all of the images separately.

During actual use of the system, semantic information about images could also be determined without the existence of associated text metadata. For example, section 7.2.1 described several ways in which the user could find out about the garment depicted in figure 7.1. The system determined what the object was, and offered additional information about the garment without having any associated information, textual or otherwise. The system used only the very generic image features used by *QBIC*.

Thus, the semantic layer technique has alleviated the need to associate metadata with media manually with reasonable success, even using only very generic image features. If more effective, application-oriented image features are used, the effectiveness of the technique in this area is likely to be greatly improved.

8.1.2 Recognising Different Views

Even in cases where effective image matching techniques are available, they often cannot identify cases where the same object, or category of object, are viewed from two different angles, or under different lighting conditions, for example. The semantic layer allows the association of views differing in such ways; this relationship is implicit if both views are connected to the same concept.

An example of this was demonstrated in the scenario described in section 7.2.1. The user was able to find different views of the same object by classifying the query image, and opening a concept browser focussed on the resulting concept. The concept browser is shown in figure 7.8. The user can see different representations of the concept *Textile Arts*, many of which show very similar objects but from different angles and in a variety of lighting conditions.

The knowledge of different views can be further used by the system. Section 4.2.3 describes how queries can be expanded using these different views. Using this technique, different views of the same concept may be retrieved, even if those views are not explicitly connected with the concept; figure 4.4 demonstrates this point.

It is likely that query expansion will work well only if the concept has a small number of representations, otherwise too much may be retrieved. How the results of such a query are presented to the user is also an issue that may require attention.

The semantic layer holds implicit information about the relationship between different views of media objects. We have shown that this information can be exploited in a number of ways.

8.1.3 Following Hypermedia Links Across Media and Views

The generic link mechanisms in Microcosm and MAVIS 1 allowed the following of hypermedia links from any instance of the source anchor of a link, provided the instance was in the same medium and sufficiently similar to the source anchor. It is not possible to follow the link from a piece of media that represents the same object, but is different in terms of low-level features, or in a different medium.

The semantic layer alleviates this problem by effectively making all representations of a concept, whatever their low-level features and medium, equivalent. This

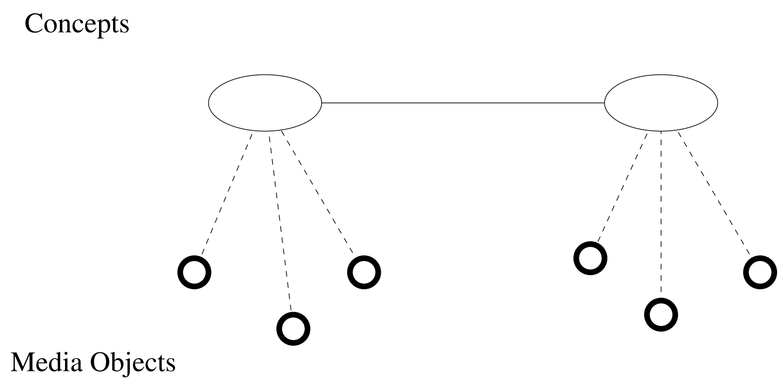


Figure 8.1: Relationships Held Between Concepts

way, if any representation is the source anchor of a link, that link can be followed from any other representation, or distinct likeness thereof.

For example, in scenario one in section 7.2.1, a hypermedia link existed between an image depicting a garment and a textile museum Web site. The link was followed from a query image that was visually dissimilar in terms of the available low-level features to the source anchor of the link. The query image was similar to another representation of the same concept, so the link was accessible. The link could have been followed from any other representation, regardless of medium (for example the text 'Textiles').

Another benefit of this approach is that cross-medium *retrieval* is also possible. For example, in scenario two, described in section 7.2.2, the user was able to retrieve images of carvings by issuing the text query 'Carvings'. The user could also follow hypermedia links from any of the representation of the concept *Carving & Carvings*.

In this way the semantic layer has addressed the problem with the generic link mechanisms of Microcosm and MAVIS 2.

8.1.4 Semantic Relationships

The semantic layer offers the facility for expressing explicit semantic relationships between objects, largely unavailable in existing multimedia information systems. Relationships between media objects are implied by the relationships between the concepts with which they are associated. In fact, the relationships are stored very efficiently, in the manner shown in figure 8.1, as opposed to holding the relationships between each media object individually, as shown in figure 8.2.

The relationships can be very useful for the user. To use the example given in the introduction in section 1.2, the user may encounter the word 'Southampton'. If they then ask the system for relevant concepts, they will be presented with the

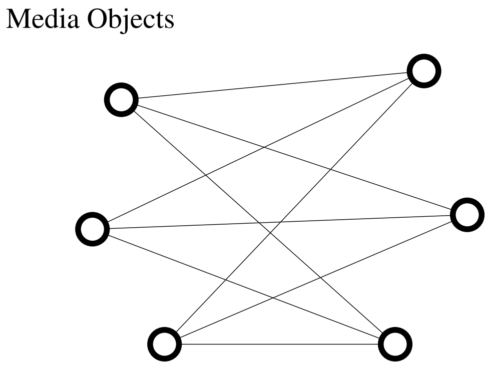


Figure 8.2: Relationships Held Between Media Objects

concept 'Southampton', and will be able to see the related concept, 'Highfield'. Any links from any representation of either concept can be followed by the user.

Additionally, the semantic relationships allow a different, high level form of navigation. The user may navigate around the concepts using the concept browser. This was demonstrated in scenario two in section 7.2.2. By starting at the root concept (shown in figure 7.13), the user was able to navigate to relevant information in a variety of media (shown in figure 7.12), without requiring a media example.

The semantic relationships may also be exploited in other ways. In scenario three, described in section 7.2.3, the relationships were used to isolate a relevant portion of the semantic layer, and this was shown to improve the precision of the retrieval. Section 4.3 described some other ways in which the relationships held in the semantic layer can be beneficially exploited.

The semantic layer therefore offers an ideal mechanism for expressing and exploiting the semantic relationships between objects.

8.2 Future Work

The following section describes how the techniques advocated in this thesis may be further improved and refined, and suggests some directions for future research in the area.

8.2.1 Richer Semantic Relationships

The multimedia thesaurus implementation, in common with most existing thesauri, supports only two types of semantic relationship other than equivalence, *broader/narrower* and *related*. The *broader/narrower* relationship is effectively a single, directed relationship between two concepts. It indicates that the scope of one concept is encompassed by the scope of another. The *related* relationship is used to indicate all other kinds of association between concepts, and is a single bidirectional connection between those concepts.

The semantic layer could support more relationship types. For example, a number of different kinds of 'related term' have been identified by Willets (Vickery, 1986), such as "an entity and its parts" and "a process and its property". If these are modelled explicitly, the user has much more information on which to base their decisions when navigating through the semantic layer.

It may be possible to integrate other forms of knowledge representation into the semantic layer, for example predicate logic (although, as discussed in section 3.3, predicate logic can be expressed in terms of a semantic net anyway). Provided that the representation contains entities that may have media representations, those entities can be associated with the appropriate entities in the same way they are associated with concepts in the semantic layer. This would allow the application of any available knowledge processing techniques.

8.2.2 Weighted Relationships and Associations

Connections in the concept layer indicate a relationship between the two concepts. The strength of the relation may vary; for example the concepts *Ceramic Arts* and *Porcelain* are strongly related; the concepts *Drawing and Decorative Arts* and *Textile Arts* might be considered less strongly related. The assigning of weights indicating the strength of a relationship will enhance in particular the deciding of the extent of 'fan-out' during the extended query expansion described in section 4.3.1.

Associations between media objects and concepts may also be weighted. For example, one image might be considered a more representative image of the concept *Glass* than another. This weighting could be taken into account during classification operations; a good match with a highly weighted representation would give more confidence that a concept is relevant than a good match with a representation weighted less.

While both of these techniques may improve the effectiveness of the semantic layer, they do introduce another level of complexity. Determining these weights is definitely a non-trivial task. Such a task would be so laborious to perform manually as to possibly outweigh the benefits of a semantic layer. It may be some time before appropriate techniques have been developed for assigning these weightings with any degree of automation.

8.2.3 A Dynamic Semantic Layer

In this thesis, the life cycle of the semantic layer (as with existing thesaurus based systems) has been divided into two distinct steps. Firstly, the application with a semantic layer component was created, and then the uses of the application so produced discussed. In practice, it may be desirable for the semantic layer to have a more dynamic life cycle.

For example, as a user encounters more media, they may decide that some piece of media is a good representation of a concept, and choose to associate it with that concept. This is possible with the current implementation of the semantic layer, the multimedia thesaurus, in MAVIS 2. In this way, the representations of concepts in the semantic layer can gradually become richer, and improve the performance of later classification queries.

However, the association of media to concepts may require some care. It is well-known, and has been evident in this work, that the set of low-level features (signature modules) used is a key factor in the success or otherwise of a classification operation. The user of a system may not have a good enough knowledge of the features used in the system to be able to judge which are most appropriate to associate the media to the concept with. This may require some administrator to make the decision. The user might submit a 'suggested association' which the administrator tweaks as necessary, and approves or rejects.

The association of additional media representations is still a relatively simple case. Associating a piece of media to a concept has no impact on the rest of the semantic layer; it does not invalidate any other part of the system. However, if users are allowed to extend the semantic layer itself by adding new concepts and relationships, the consequences to the usefulness of the system may be serious. If a user adds a new concept, it may be the case that representations of other concepts actually better represent the new concept, and thus the results of classification operations may be skewed.

For example, consider a case where a user extends the semantic layer in the museum application by adding a concept *Portraits* as a narrower concept of *Painting and Paintings*. Some representations of *Painting and Paintings* may already be portraits, so classifications of portraits found later are just as likely to be classified as *Painting and Paintings* as *Portraits*. Thus the links offered may be rather too generalised for the user's needs. As more and more concepts are added, this problem is worsened.

This is a well-known problem in classification; once the set of classes has changed, everything usually has to be re-classified. Some compromise might be reached in the case of the semantic layer; however, allowing users to add concepts is highly problematic, and will require much work before it becomes a realistic possiblity.

8.2.4 A Multilingual Semantic Layer

Throughout this thesis, a key point has been the separation of the semantic concepts, and the media objects representing those concepts. Included in this division have been the 'preferred representations' of the concepts, that is, the text term used to display the concept in the concept browser and in classification results. This is achieved by tagging one of the associations between a media object and a concept as the preferred representation.

There is no reason why this tagging cannot be used to further enhance the functionality of an application. One obvious enhancement is to allow the tagging of multiple preferred representations. A separate preferred representation can be made available for any number of languages. A user of the system can then select their desired language, and in all subsequent classification and concept browsing operations the preferred representations corresponding to that language can be presented to the user.

For example, a concept representing the real-world object *horse* could have the preferred text representation 'horse' when an English-speaking user is using the system. This could be switched to 'cheval' if a French user is using the system. If each concept is suitably represented in each language, it effectively becomes a translation tool as well.

Other representations and relevant documents associated with concepts could also be language-tagged. In this way, a single semantic layer could be used by people understanding a variety of languages.

There is also no reason why the preferred representation has to be just text, other than for the convenient ease with which it is displayed and disambiguated. By tagging sets of preferred representations, a semantic layer can be presented in a variety of multimedia 'languages'. For example, the preferred representations could be an audio clip of someone uttering the name of the concept. This may be of use to someone who is visually impaired.

This idea can be extended to a context-sensitive multimedia language. Different concepts have different connotations to people. For example, a car engineer might think of a technical drawing of a car as a representation of a car, and want it presented as such; a non-technical person might not want that representation there, as it may pick up inappropriate technical links and media.

8.2.5 User Interface Design

Chapter 7 has already mentioned that the user interface to the MAVIS 2 system is relatively unsophisticated. Given the wealth of options open to a user when a semantic layer is available, the effectiveness of the system is likely to be greatly affected by the design of the user interface.

Presently, a user must select options determining a query's 'scope'. This involves deciding whether similar media, hypermedia links or concepts should be returned by the query, and what involvement the multimedia thesaurus has. These options are shown in figure 5.7.

A better approach might be to have a small number of buttons, which when pressed initiate a commonly-used query. For example, one could start a 'find similar' query, and another could start a query finding links using the multimedia thesaurus. There is a wide scope for research into improving this aspect of the system.

8.2.6 Further Evaluation

Chapter 7 qualitatively demonstrated the usefulness of the multimedia thesaurus in MAVIS 2, and hence the semantic layer technique. For reasons explained in that chapter, the options for evaluating the multimedia thesaurus in MAVIS 2 are currently limited. Future research should address this, and produce some quantitative proof of the effectiveness of the technique.

Once the user interface is developed sufficiently, user trials can take place. One set of users can attempt to complete a set of tasks by using MAVIS 2 without the multimedia thesaurus, and the other with the multimedia thesaurus enabled. How quickly the users can complete the tasks using each method can be compared.

Empirical evaluation using purely quantitative measures is a far more complex problem. There are no readily-available benchmark datasets for multimedia collections, nor are there any well-researched evaluation methodologies for hypermedia systems that do not include user trials. A useful course of action would be to 'clean up' the user interface to MAVIS 2, perform some user trials, and apply evaluation methodology similar to that proposed by Salampasis *et al.* (Salampasis *et al.*, 1998), described in section 2.3.6.

8.2.7 Integrating with Other Work

Rather than attempting to provide a complete multimedia information system solution, the semantic layer may usefully be integrated with other work in the field.

In the case of the World Wide Web, this has already been partially achieved. Since MAVIS 2 adheres to the principles of open hypermedia, documents can exist anywhere and need not be altered to add links or semantic associations. The museum application included links to external Web sites over which the user has no control. The museum images themselves were also stored on a Web server, to which the MAVIS 2 system had no write access.

Integration with the Web could be enhanced by exploiting new technologies such as the *Resource Description Framework*, or RDF. RDF is a standard for the storage and exchange of metadata. The metadata itself can, theoretically, concern anything, though in practice it is likely to be used to describe the relationships between objects on the Web. The knowledge held within this metadata could be used as the basis for a semantic layer. The semantic layer could either be derived from the RDF metadata directly, or used in order to automatically determine the most appropriate concepts in the semantic layer that a media object concerns. Although RDF has to be assigned to individual objects, techniques are being developed to achieve this automatically (Jenkins *et al.*, 1999).

The development of RDF suggests that the importance of semantic information is being realised within the Web. The techniques described in this thesis can be developed and applied to RDF, as can any other new technologies involving multimedia information and semantic information.

8.3 Summary

The semantic layer technique does succeed in addressing the problems it was originally designed to address. The feasibility of the approach has been demonstrated in this thesis. Chapter 6 described the construction of a semantic layer (the multimedia thesaurus) using entirely automated and semi-automated processes. The semantic layer so generated was shown in section 7 to be an effective aid to retrieval and navigation. Thus, it is not necessary to spend a large amount of time manually constructing semantic layers. They can be constructed using "off the shelf" classification systems and existing automated techniques. The initial time and effort required to add a semantic layer to a multimedia collection need not be prohibitively high.

A key finding of this thesis is that the system relies on the low-level feature processing techniques that have been found wanting in some areas, in particular the automatic batch classification of images described in section 6.6. However, the system was still successful in other areas despite only having access to some very generalised image matching methods offered by QBIC.

There is a definite shift in multimedia information access research, from being concerned with only the low-level features of a medium, to the *meaning* of information held within the medium. This thesis has taken a significant step forward in this area, by describing in detail how a practical multimedia application making explicit use of the meaning of objects has been constructed, and how the knowledge within the application can then be utilised. The result is a multimedia information access system with significantly enhanced browing and searching capabilities.

Appendix A

Selection of Algorithm for Connecting Images to Concepts with Latent Semantic Analysis

This appendix describes how the cosine values indicating the similarity between keywords associated with DDC classes and image text metadata are used to assign the images to those classes.

Closer investigation into the problem soon gives rise to another question: Should images be assigned to a single class, or as many classes as seem relevant? In the case of the museum application, an image may be considered to belong to more than one class. However, as noted in section 3.4 most existing techniques for object classification are concerned with assigning an object $\mathcal{D}$ to a single class $\mathcal{C}$ from a set of mutually exclusive classes $\mathcal{S}$. If those techniques are to be employed for this application, this assumption should be honoured.

It is possible to apply an existing classification technique. However, the information available in this case is rather less than is available when using feature vectors; there is only a single cosine value. The following are the four main classifier types described in section 3.4:

Naïve Bayes classifier. In order to use such a classifier, the probability distribution of the occurence of features and objects with respect to classes must be known. In order to work out these we must have a set of example objects for each class, which in this instance we do not have. Thus the Bayesian classification approach is inapplicable.

k-nearest neighbour classifier techniques are applicable, since we already have a distance (dissimilarity) function, the cosine yielded by LSA.

Decision tree classifier techniques may also be applicable to some degree. However, we only have the cosine value on which to base decisions made at each node, and this may limit its success.

Subspace classifiers rely on a feature space with relatively high dimensionality. We can only obtain distance measures in the form of cosines from the LSA software, so we do not have access to such a feature space.

Given that only two of these types of classifier are applicable, namely *k-nearest neighbour* and *decision tree*, the following two methods for classification were considered:

Best-fit, an adaptation of the *k-nearest neighbour* technique. For each image, the largest cosine is found. The image can be connected to the corresponding concept.

A simple decision tree. Each image is initially connected to the root concept. The image is propagated down to the descendent (narrower concept) with the highest cosine. This has the effect of placing all images in the classes at the 'leaves' of the decision tree. The resulting classification may be similar to that produced by the highest cosine method, but may produce better results, since it may be 'channeled' down a more appropriate route in the concept layer.

An example of this is illustrated in figure A.1. Concepts C and E have the same cosine. Intuitively, class E is likely to be the more appropriate since the cosine for the broader class D is also high. A *best-fit* classifier must either ask the user or make an arbitrary decision as to which is most appropriate. An hierarchical classifier will classify the image into class E, since at class A the decision is made to move the image to class D first.

In order to find the most effective method, each has been tested using a small sample set of images. Each of the two classifiers has been tested with a small set of 20 images. In this way, the resulting categorisation can easily be verified. In order to be able to perform this verification, it was first necessary to define a 'ground truth'. This was decided heuristically by a user and is given in table A.1.

Table A.2 shows the performance of each method. The entries marked with an asterisk ('*') denote results of interest. Image 0761_004 has been classified by

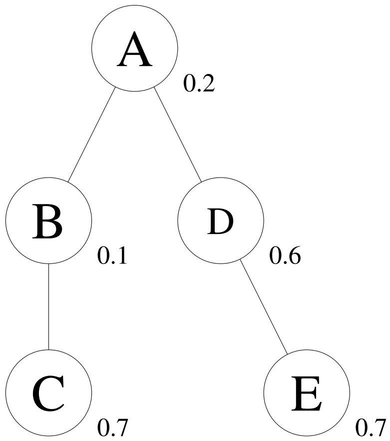


Figure A.1: A Decision Tree Channeling an Image to an Alternate Class

Image ID	**Correct Dewey Class**
0756_003	Porcelain
0756_018	Art Metalwork
0761_004	Furniture & Accessories
0761_015	Glass
0761_034	Photography & Photographs
0761_038	Furniture & Accessories
0761_046	Drawing & Drawings
0761_048	Photography & Photographs
0761_050	Glass
0761_052	Glass
0761_066	Furniture & Accessories
0761_079	Textiles
0761_104	Painting & Paintings
0826_046	Carving & Carvings
0826_059	Textiles
0826_063	Porcelain
0826_078	Earthenware & Stoneware
0826_082	Earthenware & Stoneware
0826_084	Earthenware & Stoneware
0826_088	Art Metalwork

Table A.1: Correct DDC Classes for Small Test Set of Images

Image ID	Best-fit	Hierarchical
0756_003	✓	✗
0756_018	✗	✗
0761_004	✓	✗*
0761_015	✓	✗
0761_034	✗*	✓
0761_038	✓	✗
0761_046	✓	✓
0761_048	✓	✓
0761_050	✓	✗
0761_052	✓	✗
0761_066	✓	✗
0761_079	✓	✗
0761_104	✓	✓
0826_046	✓	✗
0826_059	✓	✗
0826_063	✗	✗
0826_078	✓	✗
0826_082	✓	✗
0826_084	✓	✗
0826_088	✓	✓
Correct:	85%	25%

Table A.2: DDC Classes Assigned by LSA-Based Classification

the hierarchical method as *Carving & Carvings*, even though the *Furniture & Accessories* Dewey class would seem at first to be the most suitable. However, the chair depicted in the image does have distinctive carved legs and back, so the classification is in some sense valid. The best-fit method classified image 0761_034 as *Furniture & Accessories* when the principle suitable Dewey class is *Photography & Photographs*. In this case the image is a digitised photograph that does depict furniture, so again the classification makes some sense.

Of course, all of the images in the dataset could be classed as photographs since they are all digitised photographs. However, in this application it is the content of these images that are considered of most importance, and image 0761_034 does depict an old photograph, which in turn depicts furniture.

It is clear that in some cases it is difficult for even a human to choose the best possible class for an image. However the performance of each automatic classification method differs so much, that the best-fit method is a clear winner.

References

Agius, Harry W., & Angelides, Marios C. 1999. COSMOS—Content Oriented Semantic Modelling Overlay Scheme. *The Computer Journal*, **42**(3), 153–176.

Agosti, Maristella, Melucci, Massimo, & Crestani, Fabio. 1995. Automatic Authoring and Construction of Hypermedia for Information Retrievel. *Multimedia Systems*, **3**(1), 15–24.

Agosti, Maristella, Crestani, Fabio, & Melucci, Massimo. 1996. Design and Implementation of a Tool for the Automatic Construction of Hypertexts for Information Retrieval. *Information Processing and Management*, **32**(4), 459–476.

Aitchison, Jean, & Gilchrist, Alan. 1987. *Thesaurus Construction - A Practical Manual*. Second edn. London: Aslib.

Amato, Guiseppe, Mainetto, Gianni, Savino, Pasquale, & Zezula, Pavel. 1996 (March). Modelling Multimedia Objects for Content-Based Retrieval. *Pages 56–70 of: Proceedings of the 9th ECRIM Database Group Workshop on Multimedia Database Systems*.

Anderson, K. M., Taylor, R. M., & Whitehead, E. J. 1994 (Sept.). Chimera: Hypertext for Heterogeneous Software Environments. *Pages 157–166 of: Proceedings of the ACM Hypertext '94 Conference*.

Andrews, Keith, Kappe, Frank, & Maurer, Hermann. 1995a. The Hyper-G Network Information System. *Journal of Universal Computer Science*, **1**(4), 206–220.

Andrews, Keith, Kappe, Frank, & Maurer, Hermann. 1995b. Serving Information to the Web with Hyper-G. *Computer Networks and ISDN Systems*, **27**(6), 919–926.

Arents, Hans C., & Bogaerts, Walter F. L. 1993. Concept-based Retrieval of Hypermedia Information: From Term Indexing to Semantic Hyperindexing. *Information Processing and Management*, **29**(3), 373–386.

Bach, Jeffrey R., Fuller, Charles, Gupta, Amarnath, Hampapur, Arun, Horowitz, Bradley, Humphrey, Rich, Jain, Ramesh, & fe Shu, Chiao. 1996. The Virage Image Search Engine: An Open Framework for Image Management. *Pages 76–87 of: Storage and Retrieval for Image and Video Databases IV*. SPIE, San Jose, US.

Beaulieu, M. 1997. Experiments on Interfaces to Support Query Expansion. *Journal of Documentation*, **53**(1), 8–19.

Beitner, Nechemia Daniel. 1995. *Microcosm++: Development of a Loosely Coupled Object Based Architecture for Open Hypermedia Systems*. Ph.D. thesis, Department of Electronics and Computer Science, University of Southampton.

Berkeley Digital Library Project. 1996. *Cypress On-line Image Retrieval System Demonstration*. `http://elib.cs.berkeley.edu/cypress.html`.

Bernes-Lee, T. 1996. WWW: Past, Present, and Future. *Computer*, **29**(10), 69–77.

Bird, Colin L., & Elliot, Peter J. 1998 (February). Search Techniques within a Multiple Database Environment. *Pages 1–7 of:* Eakins, John P., Harper, David J., & Jose, Joeman (eds), *The Challenge of Image Retrieval: Papers presented at a Workshop on Image Retrieval*.

Blackburn, Steven, & DeRoure, David. 1998 (Sept.). A Tool for Content Based Navigation of Music. *Pages 361–368 of: Proceedings of ACM Multimedia '98*.

Blair, D. C. 1990. *Language and Representation in Information Retrieval*. Elsevier.

Bouet, Marinette, & Djeraba, Chabane. 1998 (Sept.). Powerful Image Organisation in Visual Retrieval Systems. *Pages 315–322 of: Proceedings of ACM Multimedia '98*.

Brink, Anne, Marcus, Sherry, & Subrahmanian, V. S. 1995. Heterogenous Multimedia Reasoning. *Computer*, **28**(9), 33–39.

Bullock, Joe, & Goble, Carol. 1998 (June). TourisT: The Application of a Description Logic Based Semantic Hypermedia System for Tourism. *Pages 132–141 of:* Grønbæk, Kaj, Mylonas, Elli, & III, Frank M. Shipman (eds), *Proceedings of ACM Hypertext '98*.

Bush, Vannevar. 1945. As We May Think. *The Atlantic Monthly*, **176**(1), 101–108.

C & C Research Laboratories. 1997. *AMORE System Demonstration*. `http://www.ccrl.com/amore/`.

Caetano, Artur, & Guimarães, Nuno. 1998 (February). A model for content representation of multimedia information. *Pages 1–8 of:* Eakins, John P., Harper, David J., & Jose, Joeman (eds), *The Challenge of Image Retrieval: Papers presented at a Workshop on Image Retrieval.*

Carr, Les A., DeRoure, David C., Hall, Wendy, & Hill, Gary J. 1995. The Distributed Link Service: A Tool for Publishers, Authors and Readers. *Pages 647–656 of: Fourth International World Wide Web Conference: The Web Revolution.* Boston, Massachusetts, USA: O'Reilly & Associates. Appears in World Wide Web Journal issue 1, ISBN 1-56592-169-0, ISSN 1085-2301.

Carson, Chad, Thomas, Megan, Belongie, Serge, Hellerstein, Joseph M., & Malik, Jitendra. 1999. Blobworld: A System for Region-Based Image Indexing and Retrieval. *In: Third International Conference on Visual Information Systems (VISUAL '99).* Springer-Verlag.

Charniak, E., & McDermott, D. 1985. *Introduction to Artificial Intelligence.* Addison-Wesley Publishing Company.

Chen, J. L., & Stockman, C. C. 1996. Indexing to 3D model aspects using 2D contour features. *Pages 913–920 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Chen, P. P. 1976. The Entity-Relationship Model — Towards a Unified View of Data. *ACM Transactions on Database Systems*, **1**(1), 9–36.

Christel, M., & Martin, D. 1998. Information Visualization Within a Digital Video Library. *Journal of Intelligent Information Systems*, **11**(3), 235–257.

Christel, M., Stevens, S., & Wactlar, H. 1994 (Oct.). Informedia Digital Video Library. *Pages 480–481 of: Proceedings of ACM Multimedia '94.*

Clitherow, Peter, Riecken, Doug, & Muller, Michael. 1989 (November). VISAR: A System for Inference and Navigation in Hypertext. *Pages 294–304 of: Proceedings of ACM Hypertext '89.*

Codd, E. F. 1979. Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems*, **4**(4), 397–434.

Colombo, Carlo, Bimbo, Alberto Del, & Pala, Pietro. 1999. Semantics in Visual Information Retrieval. *IEEE Multimedia*, **6**(3), 38–53.

Craven, T. C. 1986. *String Indexing.* Academic Press, Inc.

Crouch, Donald B., Crouch, Carolyn J., & Andreas, Glenn. 1989 (November). The Use of Cluster Hierarchies in Hypertext Information Retrieval. *Pages 225–237 of: Proceedings of ACM Hypertext '89.*

Cunliffe, Danial, Taylor, Carl, & Tudhope, Douglas. 1997. Query-based Navigation in Semantically Indexed Hypermedia. *Pages 87–95 of:* Bernstein, Mark, Carr, Leslie, & Østerbye, Kasper (eds), *Proceedings of ACM Hypertext '97*. Southampton, UK: ACM Press, for ACM.

Davis, H. C. 1995. *Data Integrity Problems in an Open Hypermedia Link Service*. PhD Thesis, Department of Electronics and Computer Science, University of Southampton.

de Saussure, F. 1986. *Course in General Linguistics*. Open Court. edited by C. Bally and A. Sechehaye with the collaboration of A. Riedlinger, translated and annotated by R. Harris, La Salle.

Dobie, M., Tansley, R., Joyce, D., Weal, M., Lewis, P., & Hall, W. 1999a (Feb.). A Flexible Architecture for Content and Concept Based Multimedia Information Exploration. *Pages 1–12 of: Proceedings of the Challenge of Image Retrieval (CIR'99)*.

Dobie, Mark R., Tansley, Robert H., Joyce, Dan W., Weal, Mark J., Lewis, Paul H., & Hall, Wendy. 1999b (Jan.). MAVIS 2: A New Approach to Content and Concept Based Navigation. *Pages 9/1–9/5 of: Proceedings of the IEE Colloquim on Multimedia Databases and MPEG-7*, vol. 99.

Dumais, S. T., Furnas, G. W., & Landauer, T. K. 1988. Using Latent Semantic Analysis to Improve Access to Textual Information. *Pages 281–283 of: ACM Computer-Human Interaction '88 Proceedings*.

Dunlop, M. D., & van Rijsbergen, C. J. 1993. Hypermedia and Free Text Retrieval. *Information Processing and Management*, **29**(3), 287–298.

Dupplaw, D., Lewis, P., & Dobie, M. 1999 (Feb.). Spatial Colour Matching for Content Based Retrieval and Navigation. *In: Proceedings of the Challenge of Image Retrieval (CIR'99)*.

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., & Niblack, W. 1994. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, **3**, 231–262.

Feder, J. 1996. Towards Image Content-Based Retrieval for the World-Wide Web. *Advanced Imaging*, **11**(1), 26–29.

Flickner, Myron, Sawhney, Harpreet, Niblack, Wayne, Ashley, Jonathan, Huang, Qian, Dom, Byron, Gorkani, Monika, Hafner, Jim, Lee, Denis, Petkovic, Dragutin, Steele, David, & Yanker, Peter. 1995. Query by Image and Video Content: The QBIC System. *IEEE Computer*, **28**(9), 23–32.

Foote, Jonathan T. 1997. Content-based Retrieval of Music and Audio. *Pages 138–147 of: Proceedings of SPIE '97.*

Fountain, A., Hall, W., Heath, I., & Davis, H. 1990. Microcosm: An Open Model with Dynamic Linking. *Pages 298–311 of:* Rizk, A., Streitz, N., & Andre, J. (eds), *Hypertext: Concepts, Systems and Applications. Proceedings of the European Conference on Hypertext.* Cambridge University Press.

Furnas, George W., & Zacks, Jeff. 1994 (April). Multitrees: Enriching and Reusing Hierarchical Structure. *Pages 330–336 of: ACM Computer-Human Interaction '94 Proceedings.*

Gaines, Brian R., & Shaw, Mildred G. 1995. Concept Maps as Hypermedia Components. *International Journal of Human-Computer Studies*, **43**(3), 323–361.

Garzotto, F., & Matera, M. 1997. A Systematic Method for Hypermedia Usability Evaluation. *The New Review of Hypermedia and Multimedia*, **3**, 39–65.

Ghias, Asif, Logan, Jonathan, Chamberlain, David, & Smith, Brain C. 1995 (Nov.). Query by Humming — Musical Information Retrieval in an Audio Database. *Pages 231–236 of: Proceedings of ACM Multimedia '95.*

Golovchinsky, Gene. 1997. What the Query Told the Link: The Integration of Hypertext and Information Retrieval. *Pages 67–74 of:* Bernstein, Mark, Carr, Leslie, & Østerbye, Kasper (eds), *Proceedings of ACM Hypertext '97.* Southampton, UK: ACM Press, for ACM.

Golovchinsky, Gene, & Chignell, Mark. 1993 (Apr.). Queries-R-Links: Graphical Markup for Text Navigation. *Pages 454–460 of: INTERCHI '93 Proceedings.*

Golshani, F., & Dimitrova, N. 1994. Retrieval and Delivery of Information in Multimedia Database Systems. *Information and Software Technology*, **36**(4), 235–242.

Goose, Stuart, & Hall, Wendy. 1995. The Development Of A Sound Viewer For An Open Hypermedia System. *New Review of Hypermedia and Multimedia: Applications and Research*, **1**, 213–232. (published 1996).

Goose, Stuart, Dale, Jonathan, Hill, Gary J., DeRoure, David C., & Hall, Wendy. 1996 (June). An Open Framework for Integrating Widely Distributed Hypermedia Resources. *Pages 364–371 of: Proceedings of IEEE International Conference on Multimedia and Computing Systems.*

Gordon, A. D. 1987. A Review of Hierarchical Classification. *Journal of the Royal Statistical Society*, **150**(Part 2), 119–137.

Grønbæk, K., & Trigg, R. H. 1996 (Mar.). Towards a Dexter-Based Model for Open Hypermedia: Unifying Embedded References and Link Objects. *Pages 149–160 of: Proceedings of the ACM Hypertext '96 Conference.*

Grønbæk, K., Hem, J. A., Madsen, O. L., & Sloth, L. 1994. Cooperative Hypermedia Systems: A Dexter-Based Architecture. *Communications of the ACM*, **37**(2), 64–75.

Gupta, Amarnath, Santini, Simone, & Jain, Ramesh. 1997. In Search of Information in Visual Media. *Communications of the ACM*, **40**(12), 34–42.

Halasz, Frank, & Schwartz, Mayer. 1994. The Dexter Hypertext Reference Model. *Communications of the ACM*, **37**(2), 30–39.

Hall, W. 1994. Ending the Tyranny of the Button. *IEEE Multimedia*, **1**(1), 60–68.

Hall, Wendy, Lewis, Paul, & Davis, Hugh. 1996 (February). *Multimedia Thesaurus and Intelligent Agent Support for Content Based Navigation and Retrieval.* Tech. rept. University of Southampton. Updated MAVIS 2 proposal to EPSRC.

Heath, Ian. 1992 (Aug.). *An Open Model for Hypermedia: Abstracting Links from Documents.* PhD Thesis, Department of Electronics and Computer Science, University of Southampton.

Hill, Gary, & Hall, Wendy. 1994 (Sept.). Extending the Microcosm Model to a Distributed Environment. *Pages 32–40 of: ECHT '94 Proceedings.*

Hirata, Kyoji, Hara, Yoshinori, Shibata, Naoki, & Hirabayashi, Fusako. 1993. Media-based Navigation for Hypermedia Systems. *Pages 159–173 of: Proceedings of ACM Hypertext '93.* Seattle, Washington, USA: ACM Press.

Hirata, Kyoji, Hara, Yoshinoi, Takano, Hajime, & Kawasaki, Shigehito. 1996 (Mar.). Content-Oriented Integration in Hypermedia Systems. *Pages 11–21 of: Proceedings of ACM Hypertext '96.*

Hirata, Kyoji, Mukherjea, Sougata, Okamura, Yusaku, Li, Wen-Syan, & Hara, Yoshinori. 1997. Object-based Navigation: An Intuitive Navigation Style for Content-Oriented Integration Environment. *Pages 75–86 of:* Bernstein, Mark, Carr, Leslie, & Østerbye, Kasper (eds), *Proceedings of ACM Hypertext '97.* Southampton, UK: ACM Press, for ACM.

Huhns, Michael N., & Singh, Munindar P. 1997. Ontologies for Agents. *IEEE Internet Computing*, **1**(6), 81–83.

Hutchings, Gerard A. 1993 (June). *Patterns of Interaction with a Hypermedia System: A Study of Authors and Users*. PhD Thesis, Department of Electronics and Computer Science, University of Southampton.

Idris, F., & Panchanathan, S. 1997. Review of Image and Video Indexing Techniques. *Journal of Visual Communication and Image Representation*, **8**(2), 146–166.

Institution of Electrical Engineers. 1991. *Inspec Thesaurus 1991*. Institution of Electrical Engineers.

Jain, A. K., Zhong, Y., & DubuissonJolly, M. P. 1998. Deformable Template Models: A Review. *Signal Processing*, **71**(2), 109–129.

Jain, R., Murthy, S., Chen, P., & Chatterjee, S. 1995. Similarity measures for image databases. *Pages 58–65 of: SPIE Proceedings, Storage and Retrieval for Image and Video Databases*, vol. 3.

Jenkins, C., Jackson, M., Burden, P., & Wallis, J. 1999. Automatic RDF Metadata Generation for Resource Discovery. *Computer Networks—The International Journal of Computer and Telecommunications Networking*, **31**(11–16), 1305–1320.

Jing, Y., & Croft, W. B. 1994. *An Association Thesaurus for Information Retrieval*. UMass Technical Report 94-17. Centre for Intelligent Information Retrieval, UMASS - Amherst.

Jones, Susan. 1993. A Thesaurus Data Model for an Intelligent Retrieval System. *Journal of Information Science*, **19**, 167–178.

Jones, Susan, Gatford, Mike, & Hancock-Beaulieu, Micheline. 1994. Support Strategies for Interactive Thesaurus Navigation. *Pages 366–373 of:* Albrechtsen, H., & Oernager, A. (eds), *Knowledge Organization and quality management. Indeks Verlag, Frankfurt.*

Joyce, Dan W., Lewis, Paul H., Tansley, Robert H., Dobie, Mark R., & Hall, Wendy. 2000 (Jan.). Semiotics and Agents for Integrating and Navigating through Multimedia Representations of Concepts. *Pages 120–131 of:* Yeung, Minerva M., Yeo, Boon-Lock, & Bourman, Charles A. (eds), *Storage and Retrieval for Media Databases 2000*. Proceedings of SPIE, vol. 3972.

Kacmar, C. J., & Leggett, J. J. 1991. A Process-Oriented Extensible Architecture. *ACM Transactions on Information Systems*, **9**(4), 399–419.

Kappe, Frank. 1993 (Aug.). Hyper-G: A Distributed Hypermedia System. *Pages DCC–1–DCC–9 of: Proceedings of INET '93.*

Knussen, Christina, Tanner, Gary R., & Kibby, Michael R. 1991. An Approach to the Evaluation of Hypermedia. *Computers and Education*, **17**(1), 13–24.

Koenemann, Jürgen, & Belkin, Nicholas J. 1996. A Case for Interaction: A Study of Interactive Information Retrieval Behaviour and Effectiveness. *Pages 205–212 of: Proceedings of the ACM SIGCHI conference on Human Factors in Computing Systems*. ACM Press.

Lai, Ting-Sheng, Tait, John, & McDonald, Sharon. 1999 (Feb.). Image Browsing and Navigation Using Hierarchical Classification. *In:* Harper, David J., & Eakins, John P. (eds), *The Challenge of Image Retrieval*.

Landauer, Thomas K., Foltz, Peter W., & Laham, Darrell. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, **25**, 259–284.

Larkey, Leah, & Croft, W. Bruce. 1996. Combining Classifiers in Text Categorization. *Pages 289–297 of: Proceedings of the 19th ACM SIGIR Internation Conference on Research and Development in Information Retrieval*. Zurich, Switzerland: ACM Press.

Letsche, T. A., & Berry, M. W. 1997. Large-scale Information Retrieval with Latent Semantic Indexing. *Information Sciences*, **100**(1-4), 105–137.

Lewis, P. H., Davis, H. C., Griffiths, S. R., Hall, W., & Wilkins, R. J. 1996a. Media-based Navigation with Generic Links. *Pages 215–223 of: ACM Hypertext 96 Proceedings*.

Lewis, P. H., Davis, H. C., Dobie, M. R., & Hall, W. 1996b (August). Towards Multimedia Thesaurus Support for Media-based Navigation. *Pages 83–90 of:* Smeulders, Arnold W. M., & Jain, Ramesh (eds), *Image Databases and Multimedia Search: Proceedings of the First International Workshop, IDB-MMS '96*.

Lewis, P. H., Wilkins, R. J., Griffiths, S. R., Davis, H. C., & Hall, W. 1997a. Content Based Navigation in an Open Hypermedia Environment. *Image and Vision Computing*, **16**, 921–929.

Lewis, P. H., Kuan, J., Perry, S. T., Dobie, M. R., Davis, H. C., & Hall, W. 1997b. Navigating from images using generic links based on image content. *Pages 238–248 of: Storage and Retrieval for Image and Video Databases*, vol. 3022. SPIE.

Lewis, P. H., Kuan, J., Perry, S. T., Dobie, M. R., Davis, H. C., & Hall, W. 1998. Content based navigation from images. *Journal of Electronic Imaging*, **7**(2), 275–281.

Li, Y. H., & Jain, A. K. 1998. Classification of Text Documents. *Computer Journal*, **41**(8), 537–546.

Ma, W. Y., & Manjunath, B. S. 1998. A texture thesaurus for browsing large aerial photographs. *Journal of the American Society for Information Science*, **49**(7), 633–648.

MacDonald, Sharon, & Stevenson, Rosemary J. 1998. Navigation in Hyperspace: An Evaluation of the Effects of Navigational Tools and Subject Matter Expertise on Browsing and Information Retrieval in Hypertext. *Interacting With Computers*, **10**, 129–142.

Miller, G. A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, **63**, 81–97.

Mitchell, T. 1997. *Machine Learning.* New York: McGraw-Hill.

Mukherjea, S., & Hirata, K. Hara, Y. 1997. Towards a Multimedia World-Wide Web Information Retrieval Engine. *Computer Networks and ISDN Systems*, **29**(8–13), 1181–1191.

Mukherjea, Sougata, & Cho, Junghoo. 1999. Automatically Determining Semantics for World Wide Web multimedia information retrieval. *Journal of Visual Languages and Computing*, **10**(6), 585–606.

Nelson, Ted. 1965. The Hypertext. *In: Proceedings of the World Documentation Federation.*

Nelson, Ted. 1981. *Literary Machines.* Self-published.

Niblack, Wayne, Barber, Ron, Equitz, Will, Flickner, Myron, Glasman, Eduardo, Petkovic, Dragutin, Yanker, Peter, Faloutsos, Christos, & Taubin, Gabriel. 1993 (Feb.). *The QBIC Project: Querying Images by Content using Color, Texture, and Shape.* Research Report RJ 9203 (81511). IBM Research Division, Almaden Research Center, San Jose, California.

Niblack, Wayne, Zhu, Xiaoming, Hafner, James Lee, Breuel, Tom, Ponceleón, Dulce, Petkovic, Dragutn, Flickner, Myron, Upfal, Eli, Nin, Sigfredo I., Sull, Sanghoon, Dom, Byron, Yeo, Boon-Lock, Srinivasan, Savitha, Zivkovic, Dan, & Penner, Mike. 1998 (Jan.). Updates to the QBIC System. *Pages 150–161 of: Storage and Retrieval for Image and Video Databases VI*, vol. 3312. SPIE, San Jose, California.

Nishiyama, Haruhiko, Kin, Sumi, Yokoyama, Teruo, & Matsushita, Yutaka. 1994 (April). An Image Retrieval System Considering Subjective Preception. *Pages 30–36 of: ACM Computer-Human Interaction '94 Proceedings.*

Nubila, B. Di, Gagliardi, I., Macchi, D., Milanesi, L., Padula, M., & Pagani, R. 1994. Concept-based Indexing and Retrieval of Multimedia Documents. *Journal of Information Science*, **20**(3).

Nürnberg, Peter J., Leggett, John J., & Wiil, Uffe K. 1998 (June). An Agenda for Open Hypermedia Research. *Pages 198–206 of:* Grønbæk, Kaj, Mylonas, Elli, & Frank M. Shipman, III (eds), *Hypertext '98: Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia.*

Ogle, V. E., & Stonebraker, M. 1995. Chabot—Retrieval From a Relational Database of Images. *IEEE Computer*, **28**(9), 40–48.

Oja, E. 1983. *Subspace Methods of Pattern Recognition.* New York: Wiley.

Oka, Ryuichi. 1998. Spotting Method for Classification of Real World Data. *Computer Journal*, **41**(8), 559–565.

Ortega, M., Rui, Y., Chakrabarti, K., Mehrotra, S., & Huang, T. S. 1997 (November). Supporting Similarity Queries in MARS. *Pages 403–413 of: Proceedings of ACM Multimedia '97.*

Ortega, Michael, Rui, Yong, Chakrabarti, Kaushik, Porkaew, Kriengkrai, Huang, Thomas S., & Mehrotra, Sharad. 1998. Supporting Ranked Boolean Similarity Queries in MARS. *IEEE Transactions on Knowledge and Data Engineering*, **10**(6), 905–925.

Pearl, A. 1989 (November). Sun's Link Service: A Protocol for Open Linking. *Pages 137–146 of: Proceedings of ACM Hypertext '89.*

Pečenović, Zoran. 1997. *Image Retrieval Using Latent Semantic Indexing.* Final Year Graduate Thesis, Swiss Federal Institute of Technology.

Pentland, A., Picard, R.W., & Sclaroff, S. 1996. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, **18**(3), 233–254.

Porkaew, Kriengkrai, Ortega, Michael, & Mehrota, Sharad. 1999 (Oct. 30–Nov. 5). Query Reformulation for Multimedia Similarity Retrieval in MARS. *Pages 235–238 of: Proceedings of ACM Multimedia '99.*

Porter, M. 1980. An Algorithm for Suffix Stripping. *Program*, **14**(3), 130–137.

Ranganathan, S. R. 1959. *Elements of Library Classification.* Association of Assistant Librarians.

Rizk, A., & Sauter, I. 1992 (Nov.). Multicard: An Open Hypermedia System. *Pages 4–10 of: Proceedings of the ACM Hypertext '92 Conference.*

Robertson, S. E. 1997. Overview of the Okapi projects. *Journal of Documentation*, **53**(1), 3–7.

Rui, Yong, Huang, Thomas S., Ortega, Michael, & Mehrota, Sharad. 1998. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Video Technology. Special Issue on Segmentation, Description and Retrieval of Video Content*, **8**(5), 644–655.

Safavian, S. Rasoul, & Landgrebe, David. 1991. A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man and Cybernetics*, **21**(3), 660–674.

Salampasis, Michail, Tait, John, & Bloor, Chris. 1998. Evaluation of Information-seeking Performance in Hypermedia Digital Libraries. *Interacting With Computers*, **10**(3), 269–284.

Salton, G. 1968. *Automatic Information Organisation and Retrieval*. McGraw-Hill Book Company.

Salton, G., & Buckley, C. 1990. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, **4**(41), 288–297.

Salton, G., & McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill Computer Science Series.

Salton, Gerard. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Santini, S. 1998 (Jan.). *Explorations in Image Databases*. PhD thesis, University of California, San Diego.

Santini, Simone, & Jain, Ramesh. 1999 (Jan.). Interfaces for Emergent Semantics in Multimedia Databases. *Pages 167–175 of:* Yeung, Minerva M., Yeo, Boon-Lock, & Bouman, Charles A. (eds), *Storage and Retrieval for Image and Video Databases VII*, vol. 3656. SPIE, San Jose, California.

Schnase, John L., Legget, John J., Hicks, David L., & Szabo, Ron L. 1993. Semantic Data Modelling of Hypermedia Associations. *ACM Transactions on Information Systems*, **11**(1), 27–50.

Sclaroff, Stan, Cascia, Marco La, & Sethi, Saratendu. 1999. Unifying Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web. *Computer Vision and Image Understanding*, **75**(2), 86–98.

Seal, A. 1995. The Creation of an Electronic Image Bank: Photo-CD at the V&A. *Managing Information*, **1**(1), 42–44.

Sheikholeslami, Gholamhosein, Chang, Wendy, & Zhang, Aidong. 1998 (Sept.). Semantic Clustering and Querying on Heterogenous Features for Visual Data. *Pages 3–12 of: Proceedings ACM Multimedia '98.*

Smeaton, A. F., & Harman, D. 1997. The TREC experiments and their impact on Europe. *Journal of Information Science*, **23**(2), 169–174.

Smith, J. R. 1997 (Feb.). *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis.* PhD Thesis, Graduate School of Arts and Sciences, Columbia University.

Smith, J. R., & Chang, S. 1996 (November). VisualSEEk: A Fully Automated Content-based image Query System. *Pages 87–98 of: Proceedings of ACM Multimedia '96.*

Smith, J. R., & Li, C.-S. 1998 (June). Image retrieval evaluation. *Pages 112–113 of: IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL-98).*

Smith, John R. 1996a. *Spatial and Feature Query System.* `http://disney.ctr.columbia.edu/safe/`.

Smith, John R. 1996b. *WebSEEk Content-Based Search Engine.* `http://disney.ctr.columbia.edu/webseek/`.

Smith, John R., & Chang, Shih-Fu. 1999. Integrated Spatial and Feature Image Query. *Multimedia System Journal*, **7**(2), 129–140.

Smith, M., & Kanade, T. 1998. Video Skimming and Characterization through the Combination of Image and Language Understanding. *In: IEEE International Workshop on Content-Based Access of Image and Video Databases (ICCV98).*

Smith, Martin P., & Pollit, A Steven. 1995 (December). Ranking and Relevance Feedback Extensions to a View-Based Searching System. *Pages 231–240 of: Online 95, 19th International Online Information Meeting.*

Smoliar, S., & Zhang, H. 1994. Content-based video indexing and retrieval. *IEEE Multimedia*, **1**(2), 62–72.

Smoliar, Stephen W., Baker, James D., Nakayama, Takehiro, & Wilcox, Lynn. 1996 (August). Multimedia Search: An Authoring Perspective. *Pages 1–8 of:* Smeulders, Arnold W. M., & Jain, Ramesh (eds), *Image Databases and Multimedia Search: Proceedings of the First International Workshop, IDB-MMS '96.*

Smoliar, S.W., & Wilcox, L.D. 1997. Indexing the Content of Multimedia Documents. *Pages 53–60 of: Proceedings: VISUAL'97; Second International Conference on Visual Information Systems.*

Sonka, Milan, Hlavac, Vaclav, & Boyle, Roger. 1993. *Image Processing, Analysis and Machine Vision.* Chapman & Hall Computing.

Sowa, John F. 1984. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley.

Sparck Jones, K., & Needham, R. M. 1968. Automatic Term Classification and Retrieval. *Information Processing and Management*, **4**, 91–100.

Sparck Jones, K., Walker, S., & Robertson, S. E. 1998. *A probabilistic model of information retrieval: Development and status.* Tech. rept. TR446. University of Cambridge, Computer Laboratory. Available from `http://www.ftp.cl.cam.ac.uk/ftp/papers/reports/`.

Stricker, M., & Dimai, A. 1996 (Feb.). Color Indexing with Weak Spatial Constraints. *Pages 29–39 of: Storage and Retrieval of Still Image and Video Databases IV*, vol. 2670.

Swain, Michael J., & Ballard, Dana H. 1991. Color Indexing. *International Journal of Computer Vision*, **7**(1), 11–32.

Terzopoulos, D., & Metaxas, D . 1991. Dynamic 3D Models with Local and Global Deformations - Deformable Superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(7), 703–714.

The J. Paul Getty Trust. 2000. *The Art and Architecture Thesaurus Browser.* `http://shiva.pub.getty.edu/aat_browser`.

Trucco, Emanuele, & Verri, Alessandro. 1998. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall.

Tudhope, Douglas, & Taylor, Carl. 1997. Naviagtion Via Similarity: Automatic Linking Based on Semantic Closeness. *Information Processing and Management*, **33**(2), 233–242.

van Dam, Andries. 1988. Hypertext '87 Keynote Address. *Communications of the ACM*, **31**(7), 887–895.

van der Heijden, Gerie, & Worring, Marcel. 1996 (August). Domain Concept to Feature Mapping for a Plant Variety Image Database. *Pages 218–225 of:* Smeulders, Arnold W. M., & Jain, Ramesh (eds), *Image Databases and Multimedia Search: Proceedings of the First International Workshop, IDB-MMS '96.*

van Rijsbergen, C. J. 1979. *Information Retrieval.* Second edn. London: Butterworths.

Vellaikal, A., & Kuo, C.-C. J. 1995 (Oct.). Content-based Retrieval of Color and Multispectral Images using Joing Spatial-Spectral Indexing. *Pages 232–243 of: Digitial Image Storage Archiving Systems*, vol. 2602.

Vickery, B. C. 1960. *Faceted Classification.* Aslib.

Vickery, B. C. 1986. Knowledge Representation: A Brief Review. *Journal of Documentation*, **42**(3), 145–149.

Vorhees, Ellen, Gupta, Narendra K., & Johnson-Laird, Ben. 1995. The Collection Fusion Problem. *Pages 95–104 of: Proceedings of the Third Text Retrieval Conference (TREC-3).*

Walker, S., & Vere, R. De. 1990. Relevance feedback and query expansion. *In: Improving subject retrieval in online catalogues.* British Library research paper, no. 2. British Library. Research and Development Department.

Wang, Weigang, & Rada, Roy. 1998. Structured Hypertext with Domain Semantics. *ACM Transactions on Information Systems*, **16**(4), 372–412.

Wiil, Uffe Kock, & Leggett, John J. 1996 (Mar.). The HyperDisco Approach to Open Hypermedia Systems. *Pages 140–148 of: Proceedings of ACM Hypertext '96.*

Xu, Jinxi, & Croft, W. B. 1998. Corpus-Based Stemming using Co-occurrence. *ACM Transactions on Information Systems*, **16**(1), 61–81.

Yahoo! Inc. 1994. *The Yahoo! World Wide Web Search Engine.* `http://www.yahoo.com`.

Yankelovich, N., Haan, B. J., Meyrowitz, N., & Drucker, S. M. 1988. Intermedia: The Concept and the Construction of a Seamless Information Environment. *IEEE Computer*, **21**(1), 81–96.