Abstract

Nowadays, the number of computer networks proliferates and a vast amount of diverse information sources are becoming increasingly available. The term Hypermedia Digital Libraries (HDLs) may be used to describe these highly dynamic, interactive and distributed information seeking environments. In HDLs information seeking is possible using combinations of opportunistic browsing and analytical, query-based strategies. The advent of HDLs posed several information seeking and architectural problems which must be properly considered before a HDL can be realised efficiently and effectively.

One problem that information seekers face in any distributed electronic environment is the collection fusion problem. This term is used to delineate the problem of selecting information sources from the many available and, the production of a single merged result which can be effectively examined. Several techniques have been proposed to solve the collection fusion problem. These techniques can not be utilised in dynamic and large environments such as HDLs easily, because they require a learning phase or excessive exchange of information. From an architectural perspective, another problem is the difficulty of conventional, "closed" hypermedia architectures to support the design and development of HDLs. Even distributed hypermedia systems like the World Wide Web have several limitations commonly found in "closed" hypermedia systems.

In this Ph.D. thesis a new collection fusion strategy is presented and systematically evaluated which facilitates hypermedia links to solve the collection fusion problem. The link-based strategy is applicable in dynamic HDLs because it does not require any learning phase nor uses excessive information to solve the collection fusion problem. Also, a novel distributed agent-based Open Hypermedia System (OHS) is presented and evaluated. The agent-based OHS can be used as an underlying platform for designing and developing open, interoperable and extensible HDLs in which multiple information seeking strategies can be integrated.

Acknowledgements

A Ph.D. programme is a long journey, "full of adventure full of discovery". The author wishes to acknowledge those who have accompanied him during this journey.

First, I wish to thank my Director of studies, Prof. John Tait, for all the stimulating discussions we had, and for his valuable advice and mindful guidance throughout my Ph.D. Thanks must go also to Dr. Chris Bloor for acting as my second supervisor.

I would like to thank Dr. Christos Batzios for his advice on the statistical analysis, but mostly for his support and advice when things looked difficult in the earlier days of my research. I wish also to thank, my friends, the researchers and other members of the staff in the School of Computing at the University of Sunderland, too numerous to mention here, to whom I would like to express my thanks for their helping along the way. I would like also to thank Colin Hardy, Dr. Clare Harvey and Hugh Sanderson who each read a part of my thesis and made some useful comments.

Also, I would like to thank the 36 volunteers who kindly participated in my experiments and, without whom it would not be possible to conduct the user-centered experiments, an important part of this Ph.D. thesis.

I must thank my entire family for their help and encouragement. Finally and most significantly, I want to give my thanks and express my gratitude to my wife Nikoletta for her unconditional support, patience and love and to my beloved son Nikolas for being the source of my greatest joy.

Ithaca

As you set out for Ithaka
hope your road is a long one,
full of adventure, full of discovery.
Laistrygonians, Cyclops,
angry Poseidon---don't be afraid of them:
you'll never find things like that on your way
as long as you keep your thoughts raised high,
as long as a rare excitement
stirs your spirit and your body.
Laistrygonians, Cyclops,
wild Poseidon---you won't encounter them
unless you bring them along inside your soul,
unless your soul sets them up in front of you.

Hope your road is a long one.
May there be many summer mornings when,
with what pleasure, what joy,
you enter harbors you're seeing for the first time;
may you stop at Phoenician trading stations
to buy fine things,
mother of pearl and coral, amber and ebony,
sensual perfume of every kind---
as many sensual perfumes as you can;
and may you visit many Egyptian cities
to learn and go on learning from their scholars.

Keep Ithaka always in your mind.
Arriving there is what you're destined for.
But don't hurry the journey at all.
Better if it lasts for years,
so you're old by the time you reach the island,
wealthy with all you've gained on the way,
not expecting Ithaka to make you rich.
Ithaka gave you the marvelous journey.
Without her you wouldn't have set out.
She has nothing left to give you now.

And if you find her poor, Ithaka won't have fooled you.
Wise as you will have become, so full of experience,
you'll have understood by then what these Ithakas mean.

Kostas Kafavis, Greek Poet, 1873-1931

List of Figures

List of Tables

Chapter 1

Introduction

Hypermedia and digital libraries are broad areas having large diversity. In this chapter some boundaries are gradually identified for the term “hypermedia digital libraries”, in order to shape the context in which this thesis will develop its ideas. In a similar effort, the basic architectural dimensions which could be used to characterise a digital library are outlined. Effectiveness and efficiency of information seeking are also defined and the collection fusion problem is introduced, a challenge for which a solution is suggested, tested and evaluated in this thesis. Finally, as a roadmap to the rest of the thesis, the hypothesis behind the undertaken research work is presented and its novelty is briefly discussed.

1.1 Rationale

We live in an information society which is becoming increasingly larger, dynamic, interactive, internetworked and distributed. A vast diversity of information sources are now available in digital form. Consequently, information seekers must generate, manage and seek information in electronic environments, while they must protect themselves from becoming overloaded by this vast amount of information.

The research that will be reported in this Ph.D. thesis, has been conducted on the assumption that our information society will continue to become more complex, as more information will continue to become available electronically. Therefore, new architectures may be required to model and develop information systems that, first, will aid users to produce, manage and share information, and second, that will assist them to efficiently and effectively retrieve information in these highly dynamic, interactive and distributed information seeking environments.

1.2 Context of the Thesis

1.2.1 Outer Context: Information Seeking and Digital Libraries

The term Digital Library (DL) is a recent addition to computer science (Fox et al, 1995a), and is used to designate the electronic environments described above. The term may evoke a different impression in people, depending on their profession and background (e.g. librarians, computer scientists, publishers). Digital libraries are, naturally, the focus of emerging areas of study (economic, sociological, educational studies of DLs can be found respectively in Sairamesh, 1996; Wiederhold, 1995; Marchionini and Maurer, 1995). This thesis, however, is concerned with what may be termed as a 'computer and information science' approach to digital libraries. A digital library, for the rest of this thesis, is therefore simply regarded as a distributed information system.

This view embodies an abstraction employed by other computer scientists in the literature. For example, Schatz and Chen (1996, pp. 22) in the guest editor's introduction to a special journal issue on digital libraries say:

> "The term digital library is actually somewhat a misnomer. Digital Libraries basically store materials in electronic format and manipulate large collections of those materials effectively. So research into digital libraries is really research into networked information systems."

Users of these networked information systems are purposefully seeking information driven by an information problem. They search for information in order to change their state of knowledge[1]. The information problem could be of any type, but it must initiate a process moving towards the ultimate goal. This ultimate goal is to attain a new state of knowledge overlapping as much as possible with a particular state of knowledge expressed by an information need (Belkin, 1980). Belkin's information theory provides the underlying framework based on which this thesis studies DLs as *dynamic and highly interactive distributed electronic environments* that *should support users in achieving this goal*.

Digital libraries as they are defined in the paragraph above, provide the *outer* context and are the broad subject of my research work (Figure 1.1). This first abstraction draws an initial dividing line between this Ph.D. work and other works on digital libraries that have been reported in the literature. For example, from an information seeking perspective, DLs which are not related to the work presented in this thesis, are those aimed at supporting learning (e.g. Fox et al, 1995b), question answering (e.g. Jean, 1994), or teaching (e.g. Barker et al, 1995). On the other hand, from an architectural point of view, this dividing line also excludes digital libraries which are relatively static and isolated (e.g. multimedia encyclopaedias or archives in CD ROM's; Crane, 1996), or they do not present a certain degree of information or service distribution (e.g. Heath et al, 1995).

1.2.2 Inner Context: Hypermedia Digital Libraries

Hypermedia Digital Libraries (HDLs) are digital libraries based on a hypermedia paradigm (Balasubramanian, 1995). In fact, the first DL which was extensively used is the World Wide Web (WWW; Berners-Lee et al, 1994), a distributed hypermedia system which can be regarded as the first large scale hypermedia digital library.

The origins of the idea of a HDL can be found in the visionary ideas presented by the hypertext[2] pioneers like Bush in his Memex (never developed) system (Bush, 1945). Also, Nelson's idea of Docuverse as a system which can store the whole humanity's literature

---

[1] this implies the definition of information as anything that can change a person's knowledge (Belkin, 1978).

[2] Usually the terms hypertext and hypermedia are used interchangeably. In this thesis the term hypermedia is mostly used since nowadays text-only systems are very rare.

(Nelson, 1980) expresses the same idea of a large scale hypermedia-based digital library. For both these pioneers, hypertext was originally seen as a natural mechanism for managing and seeking information in a *large scale* (Hall, 1994).

During the course of hypermedia development, however, new types of hypermedia have emerged like literature, scholarly and educational hypermedia (Kolb, 1997). These are small scale applications of hypermedia, experimenting with new types of writing which are different from traditional linear writing, or with new methods of learning. They differ in their use and goals from what may be called *informational hypermedia* (i.e. the hypermedia originally envisioned by Bush). The goals of informational hypermedia are to experiment with and explore new ways of managing and seeking information in a large scale. This thesis is not concerned with small-scale hypermedia applications like literature or educational hypermedia. Instead, informational hypermedia and their utility in developing HDLs are the *inner* context in which this Ph.D. work should be considered (Figure 1.1).

Hypermedia digital libraries differ from other types of DLs, because they explicitly support intuitive, opportunistic browsing strategies for information seeking. In addition to browsing, HDLs will usually support analytical (i.e. query-based) strategies because these are more efficient in large electronic environments. Users in HDLs can employ different information seeking strategies and engage in rich and complex interactions to achieve their goals.

Different views of the relationship between analytical and browsing strategies have been reported in the literature. Marchionini & Schneiderman (1988) and Frisse (1988) describe them as a useful combination which needs careful balance. Rada & Murphy (1992) view analytical and browsing strategies antagonistically. Halasz in his influential paper examining seven issues for hypermedia (1988), views analytical strategies as a complement to browsing. In fact, Halasz revisiting later his seven issues (1991) presented a more radical view describing analytical strategies as an alternative/replacement for browsing.

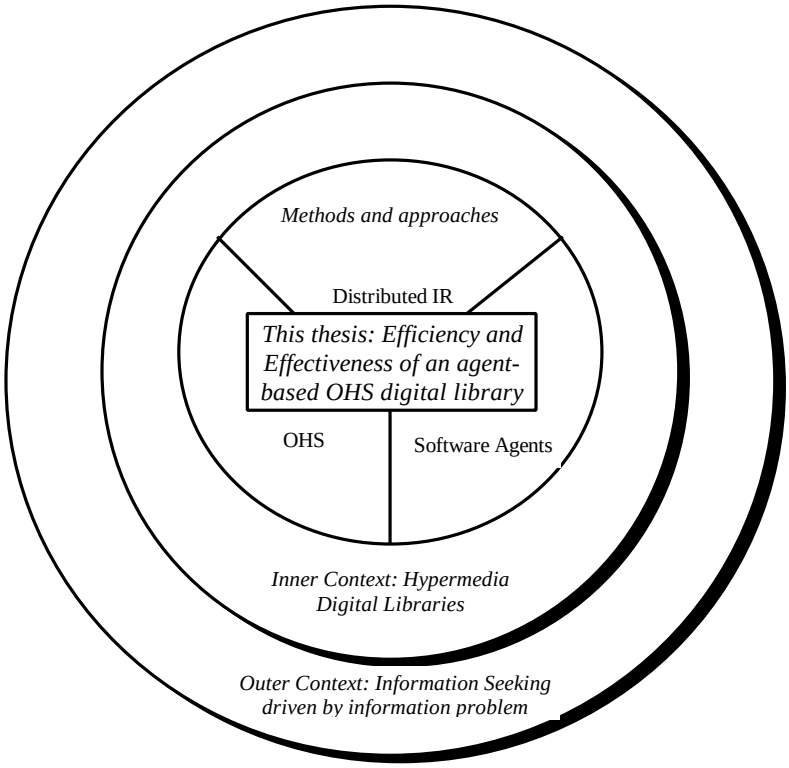


Figure 1.1: The outer, inner context and methods applied in this Ph.D. work.

1.3 Issues, Problems and Challenges

Like any other area which is still in its infancy, it is useful to classify research issues for digital libraries. Fox et al (1995a) presented a broad classification of fields that should be considered in the development of DLs. Another classification has been presented by Nurberg et al (1995). They outlined a generalised taxonomy of DL elements and a mapping of research issues to the elements identified.

Both these classifications, however, are relatively broad and can not be used in the specific context of this Ph.D. work. Therefore, another classification is utilised which serves the specific needs of this thesis. The issues, problems and research challenges posed by digital libraries are classified in two main categories. The first category contains architectural issues and challenges: i.e. which are the best architectures, data models, integration methods, tools and interfaces to support users in managing information in DL. The second challenge is more algorithmic and relates to the question of which procedures, methods and algorithms should be developed, to support end-users in efficiently and effectively seeking information.

1.3.1 Architectural Issues

Some dimensions to characterise DL systems

There are several dimensions based on which digital libraries can be studied and compared from an architectural perspective. *Distribution, heterogeneity, extensibility and scalability* are some of them which broadly characterise the architecture of a digital library. In chapter 2 these dimensions are discussed in more detail and used to critically review DL research and systems.

Interoperability

The term interoperability describes the ability that individual tools, or components of a system, or whole systems may have to communicate, agree and co-ordinate in mutually providing services, handling subtasks, or achieving sub-goals. Interoperability is essential for developing large scale digital libraries in today's electronic environments, where it is unwanted or difficult to obtain central control of this massive information flow. Engelbart (1990) distinguishes three levels of interoperability:

- between tools within a single non-distributed system (level 1);
- between different distributed instances of the same system (level 2);
- between different systems over, usually, a wide area network (level 3).

All levels of interoperability are important, but second and third level interoperability are crucially important in digital libraries because they are essential in achieving distribution.

Personal Digital Libraries

From a user-centered perspective a Personal Digital Library (PDL) should be considered as a DL at the lowest level of granularity. In other words, a PDL is the collection of tools, programs and other resources used by an individual to manage and share his/her personal information workplace. Marchionini (1995, pp. 11) has presented this idea in a broader context and termed it as the *personal information infrastructure*:

> "An individual person's collection of abilities, experience, and resources to gather, use, and communicate information are referred as *personal information infrastructure."*

The use of the term PDL in this thesis refers only to the resources (e.g. machines, programs) and not the other cognitive abilities mentioned in Marchionini's definition. In a broader context the term PDL may also include the collection of first or higher class information objects (e.g. data and indexes) which are produced by or belong to an individual. Given this broader definition, for instance, the manifestation of a PDL in WWW terms should be thought of as each user having a personal WWW server installed in their personal computer[3]. PDLs are a natural and highly distributed way to develop DLs.

1.3.2 Efficiency and Effectiveness Issues

Definitions

Effectiveness of information seeking environments is the factor indicating the degree of success that a user has in finding all or some of the required information. This factor has been traditionally measured with indices such as recall (R) and precision (P) (Sparck Jones, 1981), but also other more user-centered ways have been suggested in the literature for measuring effectiveness (e.g. Hersh et al, 1995). On the other hand, *efficiency* is the factor deciding the level of success that a system has in providing information to users in a certain amount of time, using the minimum level of resources, and, with a fair degree of user effort.

The collection fusion problem

One problem which particularly concerns this Ph.D. work is the *collection fusion problem* (Voorhees, 1994). This problem arises in searching distributed collections using query-based information retrieval. Part of the collection fusion problem is the decision that must be made by information seekers of which collections to search from those available and for how many documents (Callan et al, 1995). It also includes the decisions that must be made in order to produce a single effective merged result, from the separate results produced by the query runs in each individual collection.

The goal of a collection fusion technique is therefore to combine the results from multiple, independent document collections into a single result without reducing the effectiveness, if

[3] actually, this is already happening and many personal web servers have appeared in the market. Also, plans have been published by the company which drives personal computing to make personal web servers intrinsic parts of operating systems very soon.

possible, of searching the entire set of documents as a single collection. Besides effectiveness, a collection fusion strategy affects the efficiency of a DL, since it determines the number of repositories involved in a distributed search.

1.4 Methods and Approaches

The approach taken in this Ph.D. work to address the problems and challenges discussed in the last section was multi-disciplinary. In fact, different disciplines such as Open Hypermedia Systems (OHS) (Hall et al, 1996), Software Agents (SA) (Genesereth & Ketchpel, 1994) and Distributed Information Retrieval (DIR) (Viles & French, 1995) have been utilised to deliver a solution. OHS are hypermedia systems purposefully designed to satisfy architectural requirements such as openness, extensibility and heterogeneity. Their main goal is to deliver hypermedia functionality in an open manner. Agent-based software engineering was developed to facilitate the creation of application programs, which would act as software agents, i.e. autonomous programs that can achieve a goal on behalf of a user or on behalf of another agent. DIR studies the problems arising when searching multiple distributed collection using analytical methods.

The method was to use a high level hypermedia reference model as the starting point and to design an agent-based extensible OHS conceptual architecture for HDLs. In addition to architectural issues, this Ph.D. work considers how information seeking issues can be addressed in this agent-based OHS.

A prototype OHS system and HDL application have been developed based on the agent-based OHS architecture. This prototype system was used as a testbed to evaluate how our design can meet the architectural requirements of HDLs. Also, a novel link-based collection fusion strategy was developed and integrated into the prototype OHS system. Finally, a series of user-centered and system-centered experiments have been conducted to evaluate the effectiveness and the efficiency of the collection fusion strategy, and, also to evaluate the performance of information seekers using the prototype and other WWW-based HDLs.

1.5 The Hypothesis and Novelty of the Work

1.5.1 Hypothesis

This Ph.D. work was initially driven by the broad hypothesis that OHSs form a better underlying platform than other distributed hypermedia systems (e.g. WWW) for designing and

developing HDLs. This initial hypothesis was mainly based on expectations that OHSs could provide a flexible enabling framework for developing rich and effective hypermedia-based information seeking environments.

The initial research work[4] that had been conducted to test the broad hypothesis described above, resulted in the development of one of the two main hypothesis investigated in this thesis. This hypothesis is: a distributed OHS can be engineered according to the principles of agent-based software engineering. An OHS engineered according to these principles can provide a solution to the interoperability issue in OHSs. It was further hypothesised, this agent-based OHS can be used as an underlying platform to develop HDLs because it can provide a useful conceptual framework for modelling *both* the architectural and the information seeking aspects of a dynamic, complex, distributed hypermedia digital library. The resulting HDL will be extensible and flexible enough so it could integrate within a single framework different methods and tools supporting different strategies for information seeking.

The second hypothesis which was investigated is that information seekers could benefit from an HDL which automatically supports some of their activities during an information seeking process. More specifically, it is hypothesised that users can benefit from a new and novel set of algorithms and procedures which can provide a solution to the collection fusion problem in dynamic and large electronic hypermedia-based environments. Information seekers can benefit both by increasing the effectiveness and the efficiency of their analytical information seeking activities, and by reducing the required cognitive load in taking other decisions during the information seeking process (e.g. selection of starting point for browsing).

1.5.2 Novelty

The novelty of this research work can be divided in two separate but complementary categories. The first category of novelty relates to our contribution of a novel and unique idea for solving the collection fusion problem in hypermedia digital libraries. The technique which is suggested, for the first time, facilitates links to solve the collection fusion problem. It can be used in dynamic environments where it is not possible to efficiently use and apply other collection fusion methods.

[4] this included mainly literature surveys and informal comparisons of hypermedia systems

The second category relates to the architectural problems and challenges in designing and developing hypermedia digital libraries. In this thesis, a novel agent-based architecture for developing a distributed OHS is presented. The architecture is novel since for the first time software agents and other concepts and ideas from agent-based technologies are examined in such breadth and depth, shaped, tailored and utilised to design an OHS. The use of an agent communication language is introduced for the first time as a basis for interoperability between different open hypermedia systems. This language addresses most of the problems of existing OHS protocols and, can support all levels of interoperability.

Research in OHS has been mainly driven by the development of system architectures for providing link services and by the integration of external viewers with the hypermedia environment. The approach taken in this thesis is different. This research work focus on the integration of different information seeking strategies through the integration of the methods and tools implementing these strategies. The proposed agent-based OHS architecture focuses on the development of protocols and therefore explicitly and deliberately emphasises the superiority of OHS protocols over OHS architectures.

1.6 Thesis Structure

Chapter 2 discusses different supporting technologies used to develop digital libraries. These different approaches are compared in terms of the basic architectural dimensions, and the information seeking strategies they support. Chapter 3 focuses on (distributed) OHSs. It evaluates them from the point of view of using OHSs as underlying platforms for HDLs. Chapter 4 completes the literature review by presenting the collection fusion problem and reviewing current suggested solutions.

Chapter 5 presents a novel and original solution for the collection fusion problem in hypermedia digital libraries. The methodical system-centered evaluation of this collection fusion strategy using six different hypermedia digital libraries is also presented and discussed.

Chapter 6 focuses on a novel agent-based conceptual OHS architecture for designing and developing hypermedia digital libraries. In chapter 7 a prototype OHS system and HDL application, based on the architecture presented in chapter 6, are discussed. The prototype system and application are also considered from an information seeking perspective.

Chapter 8 presents a user-centered evaluation which aims to assess the effect of distributed searching/fusion strategies in information seeking environments and, also aims to evaluate and compare our prototype agent-based HDL with other analogous HDLs.

Chapter 9 discusses the originality of this Ph.D. work. Finally, this chapter concludes the thesis by suggesting and discussing areas for further work.

Chapter 2

Supporting Technologies for Digital Libraries

This chapter describes literature reviews made to analyse, classify and compare four different supporting technologies for digital libraries. The first technology for DLs is distributed information retrieval. The second technology under review is distributed hypermedia. Third, distributed multi-databases, and fourth, intelligent information systems are discussed as methods for supporting digital libraries. All technologies are critically reviewed and compared using the basic architectural dimensions and information seeking issues identified in Chapter 1.

2.1 Introduction

In Chapter 1, DLs are defined "simply" as networked information systems for supporting information seeking activities driven by an information problem. However, even within this abstract framework different approaches to the development of DLs are conceivable. Croft (1995) examines DLs as distributed text-based information retrieval systems. Wilensky (1995) views DLs as a federated collection of distributed databases and services. Schatz (1995) describes them as a distributed space of interlinked information. Other researchers have exposed the potential of using OHSs for developing digital libraries (Hall et al, 1996 pp 154). Finally, efforts have been reported for developing DLs using WWW technology (e.g. Balasubramanian et al, 1997).

It becomes evident from the above, that reviewing DLs is a complex task which involves multiple disciplines and therefore requires an analytical approach. In this chapter the twofold classification made in Chapter 1 (i.e. architectural and information seeking issues of DLs) is used to undertake an analytical review of DL research.

As a first step towards this approach DL research and systems is "topologically" divided into four main categories:

- those from the IR community;
- those presented within the thread of hypermedia research;
- those developed under the label of multi-database systems; and
- finally, those developed under the label of intelligent agent information systems.

The systems which are going to be reviewed in this chapter are not always labelled as digital libraries. Some of them mention the term DL rarely (even not at all in some cases) in their original presentations in the literature. However, they are all characterised by a defining criteria for this thesis: they are information systems exhibiting distribution. Another criteria is that they all share the goal to support general purpose information seeking activities. These were the two basic criteria's used to select systems for review.

Of course, exhaustive review of the selected systems was impractical. Instead, for each system a digest was extracted of the aspects which closely relate to this thesis. Section 2.7 summarises reviews of individual systems and the basic characteristics of each supporting technology.

2.2 Dimensions of DL Systems

In Chapter 1 *distribution, heterogeneity, extensibility and scalability* were enumerated, amongst others, as DL architectural dimensions. Here, these architectural dimensions are defined in more detail, so they can be later used to assess the DL systems under investigation.

Distribution refers to the capability that a DL system architecture may have to disperse information or/and services among different computer systems across a local or wide area network. A distinction should be made between distribution of information and distribution of services, which are two separate although usually related issues. For instance, information in a DL may be physically distributed, but on the other hand, the tools and the services used by information seekers to access the data may be centralised in a single machine.

Heterogeneity describes the capability of a DL system to incorporate data, tools, data models, interfaces, services which are not similar to those originally provided by the system designers, nor are they based on identical technologies or principles. For example, a DL system which supports two different methods for storing and accessing information (e.g. a relational database and a file system), should be regarded (in terms of storage methods) as heterogeneous.

Extensibility is the capacity of a DL system to embrace new participating systems and/or new information repositories that were initially developed outside the DL system. Extensibility additionally refers to the capability of DLs to extend the services that offer to information seekers, i.e. to extend the functionality of the information workplace by adding new tools, incorporating new methods of interaction and information seeking strategies.

Scalability expresses the capacity of a DL system to scale to larger numbers of users and tasks that can be efficiently handled by the system. This dimension is important since DLs will be usually required to supply information seeking services to hundreds or even to many thousands of users.

2.3 Distributed Information Retrieval (DIR)

What is DIR

Information retrieval (IR) is the discipline which studies what has become known as the "information retrieval problem" and can be described as follows (Huibers et al, 1996):

"In which way relevant information can be distinguished from irrelevant information corresponding to a certain information need."

Systems which are trying to solve this problem in an automatic way using query-based analytical strategies are called information retrieval systems. The most publicised IR models are the Boolean (Salton et al, 1983), the Vector Space (Salton et al, 1975) the Probabilistic (Robertson, 1977) and a model of IR based on logic (Van Rijsbergen, 1986).

IR follows a more theoretical treatment of the IR problem, without giving too much attention to architectural issues. Also, conventional IR traditionally deals mostly with the problem of searching a single collection. However, the advent of DLs has influenced IR research and the attention paid to very large distributed information systems has recently increased. Suffice it to say, that all keynote addresses in two recent SIGIR conferences had as their main subject the digital library infrastructure (Winograd, 1995), or the tools and the evaluation of digital libraries (Hopper, 1996; Sarajevic, 1995). At the same time, whole SIGIR conference workshops were dedicated to the study of Distributed IR (DIR).

Figure 2.1 presents a model which largely represents conventional IR systems (Croft, 1990). This model is inadequate for DIR systems which must search multiple collections in distributed environments. A simple model of DIR systems contrasted with the conventional model is presented in Figure 2.2. This new model has some important implications and uncovers problems and challenges that didn't exist in conventional IR. This is precisely the goal of DIR, i.e. to study these new implications, problems and challenges, and to produce effective and efficient solutions.

There are two main approaches in DIR. The first approach relies on maintenance of a centralised index, built by systematically and exhaustively downloading and indexing all the documents from remote collections (Figure 2.3). The second approach is to treat the distributed collections separately. Each collection is searched individually and the separate results are combined to produce a single result (Figure 2.2). In the next sub-sections systems based on both approaches are reviewed.

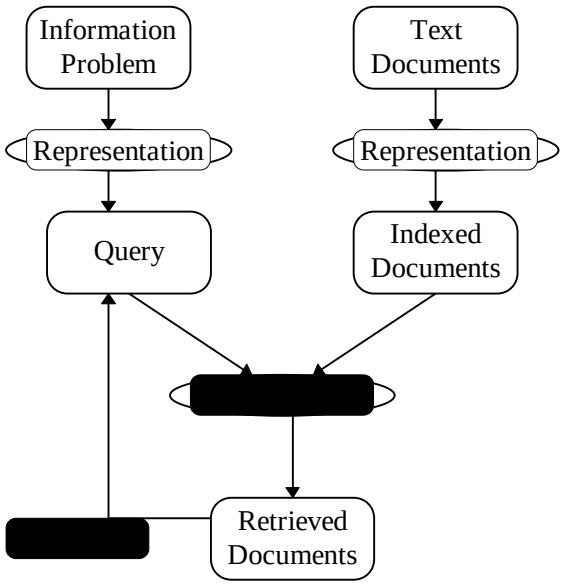


Figure 2.1: A model of conventional IR.

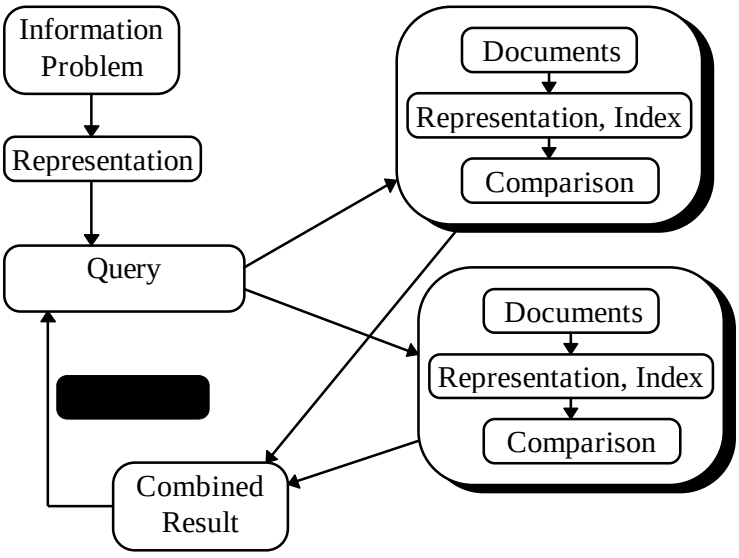


Figure 2.2: A model of distributed IR with distributed indexes.

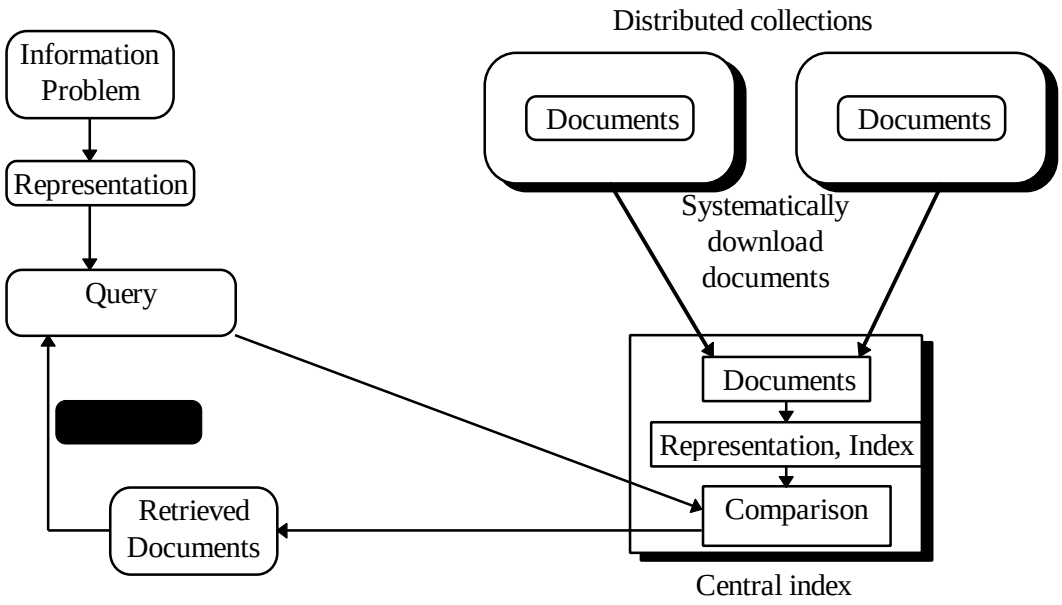


Figure 2.3: A model of distributed IR with centralised index.

The Z39.50 protocol

Z39.50 is a mature standard for information retrieval in networked environments dating back to 1970 (Lynch, 1997). The protocol specifies data structures and interchange rules to allow a client machine to search indices on an information server machine.

A weak point of Z39.50 protocol is that it allows interoperability between two machines only, i.e. one client can interact only with one server at a time (Payette et al, 1997). This could be a critical limitation for DL applications, because lack of support for parallel "broadcast" searching can negatively affect usability and efficiency. This deficiency can be overcome only at the implementation level using multiple concurrent Z39.50 connections to multiple servers on top of the protocol.

WAIS

WAIS (Wide Area Information Server) is the earliest indexing system on the Internet (Stein, 1991). The Z39.50 protocol can be used to search WAIS servers, but the WWW is increasingly becoming the "standard" to access WAIS systems. The WAIS system, similar to Z39.50, allows only a "point to point" connection and searching. Parallel searching of multiple collections is not possible. Also, like Z39.50 the selection of which information server to search must be made explicitly by the information seekers.

Technical Reports Digital Libraries

One of the first applications of digital libraries was the development of computer science technical report services which employ networked information systems instead of a paper based manual system. It was primarily the advent of the WWW which prompted that idea and provided the underlying communication and network mechanism (Fox, 1995). A number of academic institutions in the USA have established several DL efforts to provide access to a number of participating distributed collections of computer science technical reports. The common characteristic of these DLs is that they all used a central or distributed IR indexing and searching method as the primary means for information seeking.

WATERS (Wide Area Technical Report Service) is one of the most developed DL of this type (Maly et al, 1994; French et al, 1995). Contributors to WATERS store locally at their sites documents which become available through the WWW and WAIS search engines. The WATERS system is partially extensible since new collections become easily a part of the DL, simply by installing the WATERS local servers. Local servers are responsible for transmitting index information to the WATERS central server which globally provides indexing information. Probably, WATERS is not very scaleable primarily because of its centralised indexing approach. From an information seeking perspective, WATERS supports full-text searching of the centralised index. Simple browsing is supported through browsable collection lists.

Another DL architecture developed at Cornell University is called Dienst (Davis & Lagoze, 1994). Dienst is a protocol and implementation that provides access over the Internet to a distributed, multi-format document collection. Dienst servers across the Internet interoperate to manage and provide access to Dienst users (Lagoze & Davis, 1995). Dienst is an extensible DL because new sites can participate by simply installing the special software supplied. In theory, Dienst is also a heterogeneous DL, since every site can use its own search engines and indexing tools. However, this possibility hasn't been extensively explored and presented in Dienst's literature.

The distinguishing feature of Dienst as an information seeking environment is its support for parallel searching. In parallel searching, multiple libraries can be concurrently searched and a single result is returned to the user. However, the user must supply the list of sites that s/he wishes to search. The Dienst DL supports bibliographic searching as well as full text retrieval. Simple browsing is also possible through browsable site lists. Dienst can support electronic

publishing and dynamic environments better than WATERS since indexes are distributed together with the responsibility to create and update them.

In general, WATERS and Dienst demonstrate the shortcomings and advantages of two fundamentally different DL architectures: those of centralised and distributed indexing. Centralised indexing and searching can be more effective and manageable, but distributed is more extensible and promotes heterogeneous, dynamic and scaleable DLs. Additionally, as it will be shown in Chapter 4, in theory distributed indexing schemes can be more effective than centralised, if particular conditions are fulfilled.

The UCSTRI system (Unified Computer Science Technical Report Index; VanHeyningen, 1994) is another TR DL system designed and developed at Indiana University. It supports a central index and searching capabilities similar to the WATERS system. Finally, Harvest is a TR DL exhibiting multiple IR methods (Bowman et al, 1994). Harvest is mainly a resource discovery system that it encourages sites to run special gatherer software. Harvest tools gather and index topic specific information at a central location for future central searching.

The six NSF/ARPA DL projects

After the first initiatives described in the last section, many small DL projects have been funded in UK under the eLib (Electronic Libraries) programme (Rusbridge, 1995), in New Zealand (Witten et al, 1995), and in Europe under the European Research Consortium for Informatics and Mathematics (ERCIM). The US government followed a different approach and only six large four-year, research-oriented but comprehensive projects were funded (Schatz & Chen, 1996). This section briefly reviews these six projects. More information about these projects will be given in the last section of this chapter which presents a synthesised view of supporting technologies for DLs.

The University of Illinois project aims to develop an infrastructure for indexing and searching scientific literature as a single federated digital library (Schatz, 1995). This project investigates full-text retrieval of unstructured and structured documents. From an architectural point of view the main design goal is to provide a single Internet interface to multiple, heterogeneous collections in distributed repositories. Concurrently, they are designing and implementing the Interspace, a prototype system to support semantic retrieval of information (Schatz et al, 1996).

The university of Michigan DL (UMDL) project emphasises the diversity and heterogeneity of digital libraries (Birmingham et al, 1995). In designing their DL they follow an agent-based approach which aims to distribute information retrieval tasks to highly specialised agents (Birmingham, 1995). Large numbers of fine-grained agents promote modularity, scalability and provide the framework for interoperability.

Three classes of agents exist in the UMDL architecture (Atkins et al, 1996): user interface agents, mediator agents and collection interface agents. The user interface agents (UIAs) manage the interface between users and resources available in the DL. Typical tasks undertaken by UIAs are to maintain user profiles and help users to create queries in a form that other agents can understand. Mediator agents will deal exclusively with other agents to provide intermediate services. They will typically direct a query from a UIA to a collection, monitor process and transmit results etc. Finally, collection interface agents (CIAs) manage the UMDL resource collections.

The two DL projects described above are closer to the research work presented in this thesis than other DLs of this category. Both projects emphasise distribution, extensibility and heterogeneity as the critical features of a DL. Information seeking in networked environments seems to be also considered as a primary issue which deserves major effort.

The third project at the University of California at Berkeley (UCB), defines a DL as a collection of distributed services (Wilensky, 1995). The architecture of their DL systems, consists of repositories, clients, indexing, searching, interoperability and protocols (Wilensky, 1996). Interoperability at all levels is one of the final aims of this project. The DL project at Stanford University adopts a view of DLs which is similar to UCB, i.e. as a distributed collection of services (Paepcke et al, 1996). Hence, as it would be anticipated, the Stanford DL project also emphasises the importance of interoperability and makes extensive use of standards such as CORBA and Z39.50.

The projects at Carnegie Mellon University (Christel et al, 1995; Wactlar et al, 1996) and the University of California at Santa Barbara (Smith & Frew, 1995) plan to provide access to

new media such as video which are difficult to index and search. In that sense, these two projects are not close to the work reported in this thesis[5].

2.4 Distributed Hypermedia Systems

Back to the future ?

It will be recalled, that the original vision presented by the hypertext pioneers argued for large scale information management and seeking environments. These ideas, however, were lost to some extent in the hypermedia systems presented in Conklin's classic survey (1987). Indeed, Halasz (1988) described first generation hypermedia systems as mainframe-based without having the specific intention to promote large scale electronic environments. Second generation systems followed the same principles. There was only one important difference in that they used advanced display technology and therefore supported more forms of data (e.g. graphics) and had better interfaces.

In recent years, however, the original hypertext vision has attracted interest again. This should be probably attributed to the explosive growth of the Internet. Distributed hypermedia systems such as the World Wide Web, the Hyper-G and its successor HyperWave system have appeared, which support large scale hypermedia systems, i.e. hypermedia digital libraries in this thesis terms. For historical reasons the KMS system, which was probably the first hypermedia system having significant distribution capabilities, is also briefly discussed.

The KMS system

KMS is a distributed hypermedia system developed to help organisations manage their knowledge (Akscyn et al, 1988). It uses a hierarchical data model and a simple interaction model which seeks performance through simplicity and speed. The data model of KMS is implemented using a database which can be distributed to an indefinite number of database servers. However, distribution is limited to data and not to services which are centrally

[5] It is a fact that access and retrieval of media such as video is far behind from retrieval of text-based information (e.g. Lewis et al, 1996). Although indexing and retrieval of these media can be based on textual substitutes (e.g. Dunlop, 1993), this is not a natural and comprehensive solution to this problem, and probably more revolutionary approaches will be required (e.g. Hirata et al, 1997). In this thesis, the term "information seeking" refers mainly to text-based environments.

located. From an information seeking perspective, browsing is supplemented with a program that can search for text strings in any hierarchy of nodes.

The World Wide Web

The World Wide Web (WWW) is a distributed hypermedia system based on a client/server architecture (Berners-Lee et al, 1992). Its tremendous success made the WWW, in the eyes of many people, a separate concept from hypermedia. This phenomenal growth has apparently affected the hypertext research community, and much discussion takes place about the WWW and possible interactions with it (Smith, 1997). The simple data and interaction model of the WWW and the use of embedded links, lead many hypermedia researchers to discount WWW as an efficient hypermedia system. The author's view is that despite its deficiencies the WWW can not be so easily discarded.

The WWW comes as a set of different things that should be distinguished (Berners-Lee et al, 1994):

- a simple directed graph data model coupled with a simple "click on and goto" method of interaction;
- the address system (Universal Resource Locators, URL) for uniquely addressing objects in a highly distributed environment in which all items have a unique reference;
- a network protocol called HyperText Transfer Protocol (HTTP) used by WWW servers; and
- the HyperText Markup Language (HTML) which is used for creating WWW documents.

From an architectural perspective, the WWW successfully demonstrated the importance of using well defined, simple and easy to understand protocols (e.g. HTML). However, its support for interoperability is limited between client/server. The extremely small set of possible interactions (i.e. GET, POST) also severely limits interoperability. The use of the CGI protocol (Common Gateway Interface) to invoke external programs make the WWW flexible and relative extensible. Finally, the system is undoubtedly very scaleable because of its reliance on the Internet, its stateless architecture and the simplicity of HTTP.

From an information seeking perspective, the WWW has limited capabilities. It supports simple browsing, but it does not have any inherent support for more advanced browsing or analytical strategies. Analytical searching can be externally provided by add-on services to the basic WWW system. Usually this comes in the form of centralised discovery and indexing systems which exhaustively download documents from WWW servers (similar to the Harvest system). Despite its recognised deficiencies, the WWW has been established as the de-facto baseline against which other hypermedia digital libraries should be compared in the future.

HyperWave (Hyper-G)

Hyper-G is a distributed hypermedia system which was developed at the Graz University of Technology (Kappe, 1993), and recently has become a commercial product under the name HyperWave. The Hyper-G system aims to support the development of large and highly structured hyperinformation systems (Stubenrauch et al, 1993).

At the backbone of a HyperWave server there is a distributed object oriented database with separate storage of information objects, hyperlinks and structures. Hyper-G systems can be highly distributed and are extensible by invocation of external programs. Hyper-G also supports interoperability at higher levels than the WWW, both in the terms of the components involved (e.g. by contrast to the WWW, Hyper-G supports server to server interoperability), as well as in terms of the interactions allowed between components.

Hyper-G as an information seeking environment is superior to the WWW and that makes it more appealing. First, it natively supports hierarchical organisation of information, and therefore provides advanced browsing strategies (clustered and hierarchical browsing). HyperWave also includes an integrated search engine which inherently provides query-based full-text and attribute-based searching. Unfortunately, like other DL systems reviewed earlier, in Hyper-G the list of databases to be searched must be supplied explicitly by the information seeker.

Other research on distributed hypermedia and prototype systems

The WWW and Hyper-G are complete hypermedia systems already leaving their marks in the development of HDLs. However, there are other research efforts on informational hypermedia which are useful to briefly discuss, despite the fact they do not demonstrate complete or widely used systems.

HyPursuit is a hierarchical network search engine developed over the WWW (Weiss et al, 1996). HyPursuit clusters WWW documents based on content and link similarity. The result of the clustering process is the development of an information space suitable for clustered and hierarchical browsing as well as analytical searching. Wiesener et al (1996) report another hypermedia architecture, called SemaLink, for supporting semantic browsing in large HDLs. To achieve this type of browsing they introduce semantic nodes which capture only semantic knowledge which is used in navigation. Whilst both these research efforts recognised the inefficiency of simple browsing in large electronic environments, they propose different methods for increasing the effectiveness and efficiency of information seekers.

2.5 Distributed Multi-Database Systems

A distributed database is a network of databases stored on different computers, while appears to the end-users as a single database (Bobak, 1996). Users do not have to know how to connect to individual databases and the DDBMS software fetches the results from multiple databases and presents them to the users. Users usually query the database using special structured languages (e.g. SQL) and they must have a knowledge about the structure (i.e. tables and attributes) and the query language of the database.

Magavi et al (1995) reported a prototype distributed database system called HDMS (Heterogeneous Distributed Multimedia System) for information seeking on the Internet. HDMS is based on a client/server architecture. End-users can submit SQL queries to the HDMS database using GSQL (Gateway Structured Query Language) which provides an interface to SQL trough a WWW client.

From an information seeking perspective, HDMS has the advantage of hiding from the information seekers the location of the data and the mechanism used to access these data. By submitting a single query different repositories can be concurrently searched. On the other hand, DDBMS like HDMS are more suitable for what is defined by Van Rijsbergen (1979, pp. 2) as data retrieval (exact match is required, e.g. retrieve all the persons who work in the 'finance' department) rather than for information retrieval.

2.6 Agents for Assisting Information Seekers

Some information systems make use of predefined rules, or make use of past knowledge and other expertise in order to assist users in their information seeking activities. This assistance

usually comes in the form of specialised agents, i.e. autonomous programs which can perform specialised tasks on behalf of the user, or on behalf of other agents.

Agents for assisted browsing

WebWatcher is an agent reported in the literature which is designed to assist WWW users by providing interactive advice about the relevance of a page (Armstrong et al, 1995). From the user's perspective WebWatcher acts as a specialised agent providing them with information about useful links or web pages, while they can retain the overall control of the system.

Balabanovic and Shoham (1995) present another agent having similar goals to WebWatcher, i.e. to assist browsing activities in the WWW. This agent runs in discrete cycles aiming to present a selection of possibly interesting pages every day. In each cycle the agent:

- proactively searches the web using a user's profile;
- applies a filter and selects the best *p* pages to present to the user;
- receives an evaluation from the user; and
- updates the search and selection heuristics (i.e. user profile) according to user feedback.

Letizia is another agent for assisted browsing (Lieberman, 1995). The agent tracks the user's browsing session and tries to identify and emphasise items which might be of interest to the user. In parallel to the user's browsing, Letizia conducts a search to anticipate possible future needs. At any time, Letizia can present a list of recommendations, which the user can inspect and accept, or s/he can return to the browsing activity. In contrast to the WebWatcher system, Letizia does not require the user to state a goal, instead it tries to infer goals by applying some simple heuristics.

Agents for information filtering

Jasper (Joint Access to Stored Pages with Easy retrieval) is an agent-based system for group-based information filtering in the WWW (Davies et al, 1995). Jasper agents store meta-information, summarise and finally inform other agents which may be interest of a particular type of information found on the WWW. Meta-information is used to improve the results when later a new search request is made. Also, clustering techniques have been used in order to improve performance (Davies et al, 1996).

Another agent-based information filtering system is ACORN (Agent-based Community Oriented Retrieval Network; Marsh, 1997). ACORN is based on an information community approach for information filtering. The ACORN architecture comprises of a set of specialised agents. The GateKeeper filters incoming information and is the user interface for creating and examining agents. The InfoAgent has the task of suggesting documents to members of the information community. Finally the SearchAgent handles user's queries.

2.7 Discussion

In this section the reviews of individual systems are synthesised in order to identify the basic characteristics of each supporting technology and to compare them within a single framework.

Tables 2.1 and 2.2 present the basic characteristics of the systems reviewed in the last four sections. Each system is characterised in terms of nine different aspects, five of them architectural and the rest characterising them as information seeking environments. The characterisations are largely based on papers published in the literature, backed up in some cases by personal experiences using these systems (e.g. WWW, Hyper-G, Dienst, UCSTRI, WATERS). The goal of these tables, however, is not to give absolutely accurate characterisations of individual systems. That is usually impossible anyway sometimes because of the complexity of these systems and, sometimes because the characterisations may be based on controversial views (e.g. are the services that the WWW search engines provide part of the WWW system?). Instead, the goal is to discover trends that could be generalised, so to increase the understanding of DL supporting technologies. Another effort to synthesise and contextualize the results of the DLs review is depicted in Figure 2.4. This figure presents a three level semantic network which illustrates the basic relationships of some selected DL systems with the architectural dimensions as well as with different information seeking strategies.

Discussion of information seeking issues

The first general comment which can be made studying the tables and the semantic network is that systems developed under a particular thread of research, support a particular type of information seeking strategy.

DIR-based DLs support analytical strategies which generally are more efficient in large document collections (column 6 of Table 2.1). On the other hand, in these type of DLs across document browsing is very limited. This can be explained considering that browsing strategies

require specially prepared documents (e.g. written in HTML) and the additional effort from the authors to create links between documents. This problem has been early identified (Bernstein, 1989), and several methods for automatic generation of links have been proposed in the literature (Rearick, 1991; Allan, 1995; Cleary & Bareiss, 1995, Kellogg & Subhas, 1996). However, it seems that automatic generation of links has not yet become a primary focus of the development of DIR-based DLs and, therefore systems based on this thread of research do not extensively support across document browsing. None of the DIR-based DLs has also reported any use of clustering methods (like for example those reported in the development of HyPursuit) to prepare the libraries for clustered browsing. Lack of support for browsing strategies makes the DIR-based DLs less suitable for opportunistic information seekers and for information problems which can not be accurately defined.

Another common characteristic of the DIR-based DL systems reviewed is that they do not provide an automatic method to solve the collection fusion problem (column 9 of Table 2.1). Selection of sources must be made manually by the information seekers. This is a weak point which may negatively affect the effectiveness and efficiency of information seeking. For example, searching one of the Technical Reports DLs reviewed in section 2.3 requires explicit selection by the user of the institutions whose sub-collections must be searched. This might not be problematic with a few dozen of institutions as it is currently the population of the TR DLs, but certainly it will become problematic if hundreds or even thousands of institutions eventually participate.

Some of the DIR systems reviewed use a distributed index design while others keep the index centralised by transferring document statistics to a centralised server (column 7 of Table 2.1). However, with the Dienst system being the exception, the systems which have a distributed index design (i.e. Z39.50 and WAIS) do not inherently support parallel concurrent searching of multiple distributed indices (column 8 of Table 2.1).

Browsing strategies which are more suitable for intuitive and opportunistic information seeking are supported by distributed hypermedia systems (column 6 of Table 2.2). It could be also said, that it is easier to support analytical strategies in a hypermedia-based DL system, than to support browsing in DIR-based DLs. The methods and procedures of analytical strategies (e.g. creating indexes) are relatively easy to apply in hypermedia documents. In fact, the distributed hypermedia systems reviewed support inherently (e.g. Hyper-G) or externally (e.g. WWW) analytical strategies. However, despite the support of analytical searching, the

selection of sources is manual (Hyper-G), or the index is centralised (WWW search engines) and searching is not parallel (columns 9, 7 and 8 of Table 2.2 respectively).

DLs which make use of intelligent agents are different from other DL systems, not in terms of the information seeking strategies they support, but in terms of how they apply these strategies. Agents in these systems provide assistance for information seekers in a proactive and advisory way. Information seekers can use whatever strategies they prefer to solve the information problem at hand, but they have the assistance of a program which runs in the background. Finally, the multi-database systems address some of the points outlined above (i.e. they support distributed indices and parallel searching), but the searching strategy which they support is more suitable for data retrieval rather than general purpose information seeking.

The most striking (and probably the most important) outcome of the synthesised review discussed above, is that none of the technologies provides a complete support for multiple information seeking strategies. Different strategies have become the research field for different disciplines and are studied independently of other strategies. However, all these individual strategies are parts of the information seeking process. The author shares the view reported in the literature (e.g. Bates, 1989; Marchionini, 1995 pp. 8), that multiple strategies must be properly supported in order to achieve effective information seeking in large electronic environments such as digital libraries.

Current DL research efforts like USA's NFS projects which plan to support multiple strategies show this weakness is identified and will be explored. Also, reports have been published aiming to design theoretical models which can inherently combine multiple information seeking strategies (e.g. Bruza, 1993; Lucarrela & Zanzi, 1996; Chiamarella & Kheirbek, 1996). Other, more practical, methods which combine browsing and analytical strategies have been also reported in the literature (e.g. scatter/gather, Hearst & Pedersen, 1996; Golovchinsky, 1997). It should be expected that the information explosion and the development of large electronic environments, will cause more systems which aim to support multiple strategies to appear in the near future. In fact, the design of the agent-based OHS architecture which will be introduced in Chapter 6, was partially driven by this goal.

Discussion of architectural dimensions

In terms of distribution, all the DLs reviewed support distribution of raw data, but some of them keep centralised meta-data such as indexes. Also, to a small extend some of the DL

systems can distribute the services that provide to the information seekers, but others (basically those who have a centralised index design) have their services centrally located (column 1 of Tables 2.1 and 2.2).

Extensibility is partially supported. Most of the DLs reviewed can be extended with new collections, but the addition of new tools, information seeking strategies or services is somewhat limited (column 2 of Tables 2.1. and 2.2). Addition of new tools and services is difficult because most of the systems are "over-engineered" with closed interfaces and very limited support of interoperability. Their architecture, data models, protocols are not sufficiently flexible to allow easily the addition of new tools and services.

In terms of heterogeneity, it could be said that most systems support heterogeneity in terms of the forms of data that they can support and also in terms of the nature and the content of different repositories (column 3 of Tables 2.1 and 2.3). On the other hand, heterogeneity in services and implementations is not well supported. For example, in the Dienst system which follows a distributed index design, each participating sub-library could use its own methods for indexing and searching the "local" collection. However, this capability is not actually used and, like most of the rest DLs reviewed, the services of each participating sub-collection in the Dienst system are homogeneous.

For most of the DLs reviewed scalability remains an open question since most of the DL systems have not yet scaled to very large numbers of users. No doubt, most of the DL research efforts will approach the scaling issue build upon experiences with the Internet. It should be expected that DLs which are based on a highly distributed design and rely on the Internet infrastructure could be scaleable (column 4 of Tables 2.1 and 2.3). Of course, the WWW experience should be taken only as an indication and not as a proof of the scalability of DL systems that rely on the Internet. The WWW is a very simple client/server, stateless system without any excessive demands for communication between components of the system. The DLs which will be developed in the future will be much more complex and demanding in terms of network resources.

Interoperability is the architectural dimension which is very weakly supported by most of the DLs reviewed (column 4 of Tables 2.1 and 2.2). First, most DL systems do not support all the levels of interoperability that have been identified in Chapter 1. Additionally, communication and interoperability is in most cases stateless and supports only very simple interactions. For example, interoperability in the WWW is only between clients and servers, is very limited (only two commands are supported, i.e. GET and POST) and, most important it is completely

stateless. Each transaction fetches a document, then stops. All the other DL systems reviewed are more or less based on the same interoperability scenario.

Most of the deficiencies of early DLs that have been discussed in this section systems are addressed in the design of the six NSF/ARPA funded digital library projects (see Table 2.1). These projects are based on a highly distributed design for both data, meta-data and services of the DL. For all these projects deep, semantic and stateful interoperability is one of the main design goals. The three interoperability levels are explored and one of the main goals is to achieve interoperability between different DLs. From an information seeking perspective, distributed and parallel searching is actively explored together with advanced browsing (e.g. semantic browsing) strategies. Quite surprisingly, none of these research efforts has yet reported on the collection fusion problem. It should be remembered, however, that these projects are just ongoing (three year old) research efforts and have not yet presented final results.

Conclusion

There is no doubt that one conclusion can be drawn after the reviews of different supporting technologies for DLs and the synthesised view which was presented in this chapter. This conclusion is that no system has yet offered a full DL service. Most of the DL systems presented lack various features, and have not properly considered many architectural and information seeking issues. This can be explained by the fact that almost none of these system was originally thought of as a DL. Indeed, recent ongoing projects (e.g. NFS DL projects) which purposefully strive to design DL systems, consider a wider range of issues and a wide range of technologies.

It must be said, however, that although the question of which technology (e.g. DDBMS, DIR) has not yet been answered, the author believes that hypermedia technology has a precedence over the other technologies. In the next chapter, a hypermedia technology is reviewed (Open Hypermedia Systems) which the author believes is suitable for digital libraries.

1. **Distribution**: (D) data can be distributed, (S) services in the DL can be distributed
2. **Extensibility**: (N) none (C) it is easy to extent by adding new collections, (S) it is easy to add new services and tools
3. **Heterogeneity**: (N) none, (C) in collections/repositories, (D) in forms of data, e.g. text, graphics, etc. (S) in the services
4. **Scalability**: (L) low (M) medium (H) high
5. **Interoperability**: (N) none (L) limited (L1) level 1 (L2) level 2 (L3) level 3 (these levels are described in chapter 1)
6. **Supported Information seeking strategies**: (Q) querying, (B) browsing, (R) resource discovery, (AB) assisted browsing, (IF) information filtering
7. **Index** (ID) indexes are distributed (IC) index is centralised
8. **Searching** (CS) point to point (central) searching (PS) searching is parallel
9. **Source Selection** (M) Manual, (A) Automatic

ALL (NC) Not considered, (P) Possible, (N/A) not applicable
letters in lower case indicate limited support of the corresponding property

	Architectural dimensions					**Information seeking issues**			
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Z39.50	D, s	C	C, d	L	L	Q	ID	CS	M
WAIS	D, s	C	C, d	L	L	Q	ID	CS	M
WATERS	D	C	d	L	L	Q, b	IC	CS	M
Dienst	D, s	C	d, s	P	L	Q, b	ID	PS	M
UCSTRI	D	C	d	L	L	Q, b	IC	CS	M
Harvest	D	C	d	L	L	R, Q	IC	CS	M
Illinois	D, S	C, S	C, D, s	P	L1,3	Q, B	ID	PS	NC
UMDL	D, S	C, S	C, D, S	P	L1,3	Q, b	ID	PS	NC
UCB	D, S	C, S	C, D, S	P	L1,3	Q, b	ID	PS	NC
Stanford	D, S	C, S	C, D, S	P	L1,3	Q	ID	PS	NC

Table 2.1: Characterisation of DIR-based DL systems.

1. **Distribution**: (D) data can be distributed, (S) services in the DL can be distributed
2. **Extensibility**: (N) none (C) it is easy to extent by adding new collections, (S) it is easy to add new services and tools
3. **Heterogeneity**: (N) none, (C) in collections/repositories, (D) in forms of data, e.g. text, graphics, etc. (S) in terms of services
4. **Scalability**: (L) low (M)edium (H) high
5. **Interoperability**: (N) none (L) limited (L1) level 1 (L2) level 2 (L3) level 3 (as these levels are described in chapter 1)
6. **Supported Information seeking strategies**: (Q) querying, (B) browsing, (R) resource discovery, (AB) assisted browsing, (IF) information filtering
7. **Index** (ID) indexes are distributed (IC) index is centralised
8. **Searching** (CS) point to point (central) searching (PS) searching is parallel
9. **Source Selection** (M) Manual, (A) Automatic

ALL (NC) Not considered, (P) Possible, (N/A) not applicable
letters in lower case indicate limited support of the corresponding property

	Architectural dimensions					**Information seeking aspects**			
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
KMS	D	C	N	L	N	B, Q	NC	CS	M
WWW	D	C, s	d	H	L	B, Q	IC	CS	M
Hyper-G	D	C, s	d	H	L	B, Q	ID	CS	M
HyPursuit	D	c	d	L	L	B, Q	IC	CS	M
WebWatcher	N/A	s	N/A	N/A	L1	AB	N/A	N/A	N/A
Letizia	N/A	s	N/A	N/A	L1	AB	N/A	N/A	N/A
Jasper	D, s	s	N/A	P	L2	IF	N/A	N/A	N/A
ACORN	D, s	s	N/A	P	L2	IF	N/A	N/A	N/A
HDMS	D	C	d	H	L	Q	ID	PS	a

Table 2.2: Characterisation of Hypermedia, DDBMS and agent-based DLs.

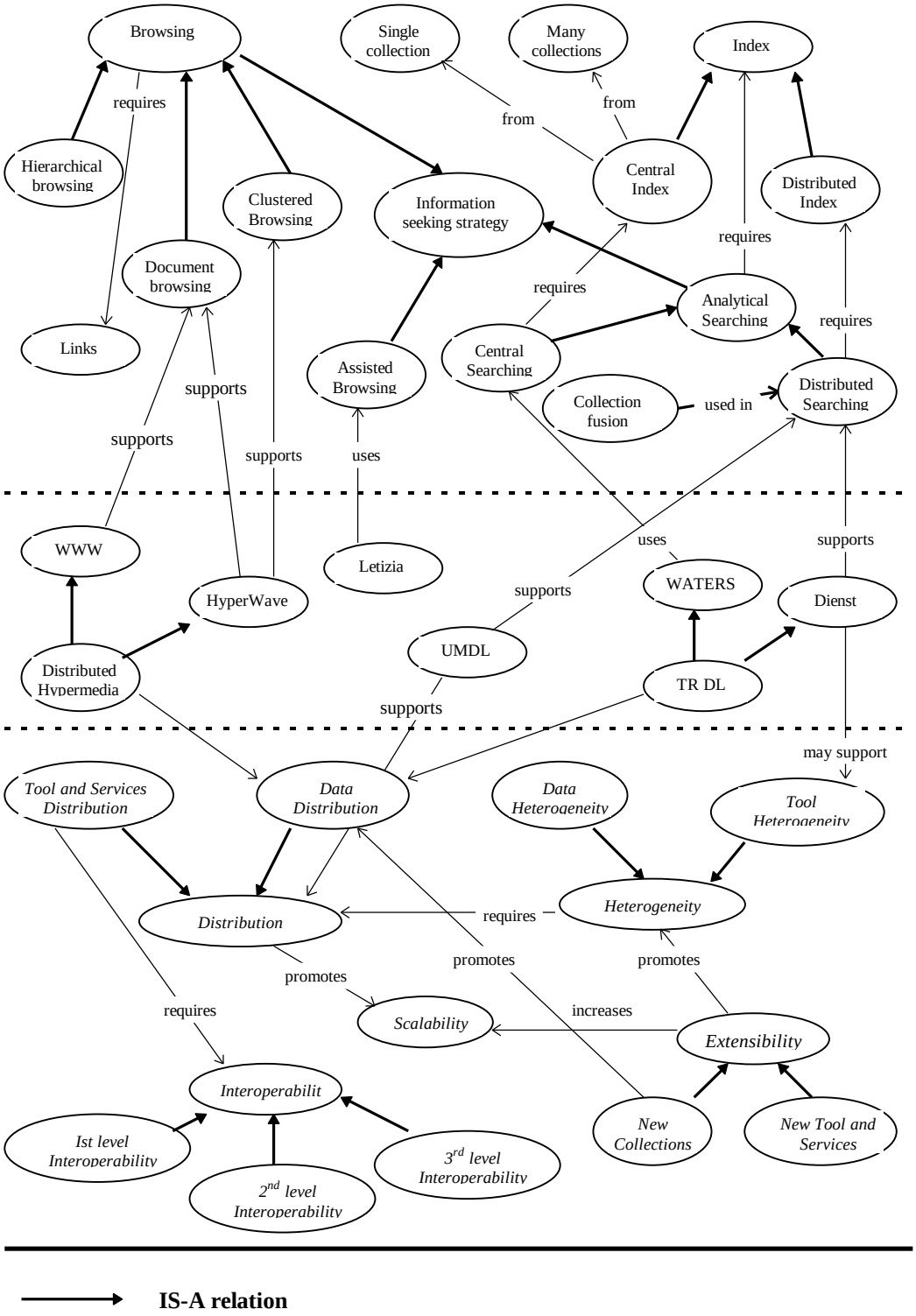


Figure 2.4: Semantic network indicating relationships and characteristics of DL systems.

Chapter 3

Open Hypermedia Systems

This chapter reviews Open Hypermedia Systems (OHSs). Distribution is again applied as the basic eligibility criteria for selecting an OHS to review. Specific OHSs are discussed after two high level hypermedia reference models are presented which are used to study, compare and classify OHSs. OHP which is a protocol for enabling communication between OHSs and third-party applications is also critically discussed. Finally, OHSs are examined as information seeking environments.

3.1 Introduction

In Chapter 2, it was mentioned how the original visions of hypertext pioneers of a universal large-scale hypermedia system, were somehow lost in the development of first and second generation hypermedia systems. The design of these systems as a closed, tightly bounded set of tools within a single application framework was another aspect that was criticised. Malcolm et al. (1991) in a well-known, influential paper have pointed out that “closed” systems can not be integrated in existing information environments. In the same paper a call was made for developing hypermedia technology for integrating and managing data. More precisely, they stated (pp. 15):

> "Hypermedia technology can be used to provide access to data *and* to manage the applications used to create the data. Although hypermedia is often thought of as a technology to deliver information, its use can be greatly expanded if it is perceived as an integrating technology."

Open Hypermedia Systems (OHSs) had first been presented a little before this call[6], to address the issues and problems presented above. In contrast to closed hypermedia systems which provide a static set of tools and functionality completely integrated within a single system, OHSs have as their main goal to deliver hypermedia functionality in an open and flexible manner to existing information environments without replacing them. OHSs manage hypermedia links for client applications. Client applications are viewers (i.e. applications that can display and edit an information object) which have the responsibility to display the data and anchors, and to handle the interaction with the users. An OHS can integrate a variety of clients through a suite of integration methods (Davis et al, 1994).

In order to achieve the goals described above, OHSs are purposefully designed to have architectural dimensions such as openness, heterogeneity and extensibility. From this point of view, it could be said that OHSs can potentially comply with the requirements of DLs as these

---

[6] the idea of open hypermedia was discussed by Meyrowitz (1987), and the first systems purposefully designed as OHSs were Sun's Link Service in 1989 (Pearl, 1989) and Microcosm (Fountain et al, 1990). However, OHS became officially a distinct identified area of hypermedia research after 1994 (year of the first OHS workshop, Wiil & Osterbye, 1994).

described in Chapters 1 and 2. In fact, the potential of using OHSs as underlying platforms for HDLs has been recently mentioned in the literature (Hall et al, 1996 pp. 154; Wiil & Legget, 1996 pp. 148).

The aims of this chapter are twofold. The first aim is to review OHS technology. This review focuses on specific distributed OHSs, in contrast to the more wide ranging and general review in Chapter 2. The second aim is to investigate if and which issues related to the development of HDLs could be or could not be addressed by current OHS technology.

3.2 Background

Before specific OHSs are reviewed, two high level reference hypermedia models are presented. The first is the Dexter hypertext reference model. The second reference architecture is the flag taxonomy. The Dexter model has a special interest for this Ph.D. work because it has been loosely used as the starting point for the design of a new agent-based OHS architecture.

3.2.1 The Dexter Hypertext Reference Model

The Dexter Hypertext Reference Model is an attempt to capture both formally and informally, the important abstractions found in a wide range of existing and future hypermedia systems (Halasz & Schwartz, 1994). The model emerged as the result of a standardisation workshop in which many of the designers of well known hypermedia systems such as Augment (Engelbart, 1984), KMS (Akscyn et al, 1988), Neptune (Delisle & Schwartz, 1986), Intermedia (Haan et al, 1992), NoteCards (Halasz, 1987) participated.

The model is divided into three layers (Figure 3.1). The *storage* layer provides only the data model and mechanisms to organise components (nodes) and links to form a hypermedia network. The *run-time* layer describes the mechanisms for supporting the presentation of the nodes to the users. The *within–component* layer covers the content and the structures within the hypermedia components. Between the three layers, there are two interface mechanisms. The *anchoring* mechanism is used for addressing locations or items within the content of an individual component. The *presentation specifications* encode information about how components are to be presented to the users. Dexter was first presented by Halasz and Schwartz (1990) and this paper has a formal specification of the model.

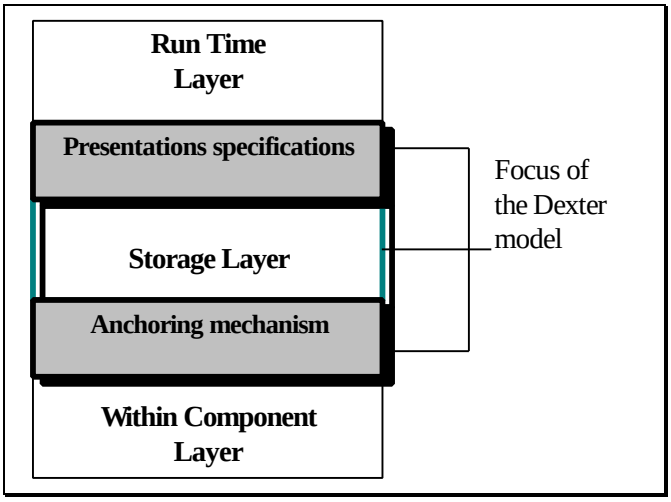


Figure 3.1: The three layered architecture of the Dexter hypertext reference model.

The importance of the Dexter model is not only based on the fact that it captured the accumulated experience of first and second generation systems (Grob? k & Trigg, 1994), but it also introduced concepts which became widespread in hypermedia research (e.g. anchoring mechanism). From this point of view the model was quite successful. This is proven by the fact that several OHSs and non-OHSs have used the Dexter model as starting point for the design of their hypermedia models (e.g. Min & Rada, 1993; Grob? k & Trigg, 1994; Hardman et al, 1994).

On the other hand, the Dexter model has weak points which had not gone without criticism. Legget & Schnase (1994) presented some of those which make Dexter inefficient, in its original form, for supporting hypermedia in a large scale. They pointed out that Dexter has no support for distribution and multiple hypertext systems, and that it implicitly assumes the components that are part of "the hypertext system" are tightly bounded. Another aspect this Ph.D. work considers as a weak point of the Dexter model, is its excessive focus on the architectural aspects of a hypermedia system. The few process/behavioural aspects which are captured in the run-time layer are very abstract, and inefficient in capturing essential notions for designing and developing HDLs.

3.2.2 The Flag Taxonomy

If the Dexter model was an attempt to capture the best design ideas of first and second generation hypermedia systems, the flag taxonomy (Osterbye and Wiil, 1996) was an attempt to classify and describe individual OHSs. The flag taxonomy also aims to characterise what

is an OHS and to compare OHSs in a system independent way. In fact, the flag taxonomy builds upon the terminology of Dexter and retains some of the basic classifications that originally introduced in the Dexter model (e.g. structure and content).

The flag taxonomy distinguishes between the storage and runtime aspects of a hypermedia system on the one hand, and structure and contents aspects on the other hand. This distinction leads to four functional modules and four protocols (Figure 3.2). Figure 3.3 shows for example how the flag describes monolithic systems such as KMS and NoteCards, which have all of their components tightly integrated. Similarly, Figure 3.4 shows the representation of the WWW which only distinguishes between storage and run-time without distinguishing between contents and structure (i.e. links are not separated from documents).

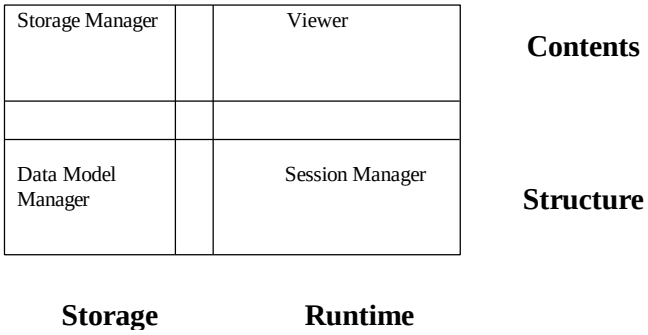


Figure 3.2: The Flag taxonomy.

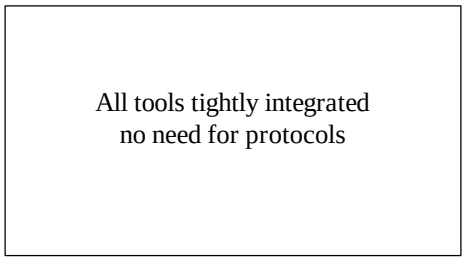


Figure 3.3: Representation of monolithic systems using the flag taxonomy.

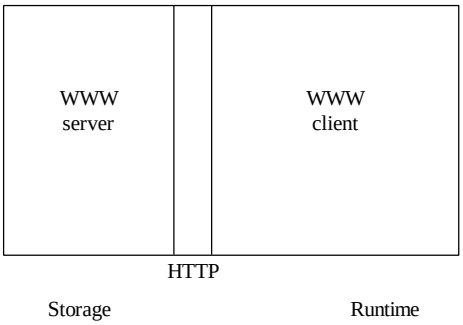


Figure 3.4: Representation of the WWW using the flag taxonomy.

3.3 Open Hypermedia Systems

OHSs are the combination of two relatively independent threads of hypermedia research (Wiil & Legget, 1997): Hyperbase Management Systems (HBMSs) and Link Server Systems (LSSs). The main feature which distinguishes HBMSs from LSSs, is that HBMSs in addition to the management of links and provision of link services to third-party tools, support a storage and sometimes a data model manager subsystem (Wiil & Legget, 1996). This subsystem usually provides the internal management of the data and meta-data (i.e. links and anchors), and sometimes supports collaboration between multiple users.

3.3.1 Hyperbase Management Systems

The typical architecture of an HBMS is the three layer architecture depicted in Figure 3.5. The hyperbase (middle) layer provides the hypermedia data model and acts as the intermediary between the application layer and the storage layer which provides persistent storage. A view of HBMSs using the flag taxonomy is illustrated in Figure 3.6. Some examples of HBMSs are CHS and Cover (Schutt & Streitz, 1990), *DHM* (Grob? k & Trigg, 1994), SP3/HB3 (Legget & Schnase, 1994), *Hyperform* (Wiil & Legget, 1992), ABC (Shakelford et al, 1993), HyperStorm (Bapat et al, 1996), and HyperDisco (Wiil & Legget, 1996).

From the systems outlined above, SP3/HB3, HyperDisco and to a less extent ABC, are purposefully designed to support large scale hypermedia. This is the reason for selecting these HBMS systems for closer review.

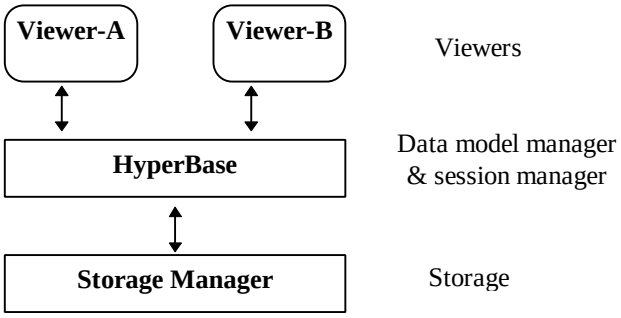


Figure 3.5: The three layer-architecture of HBMSs.

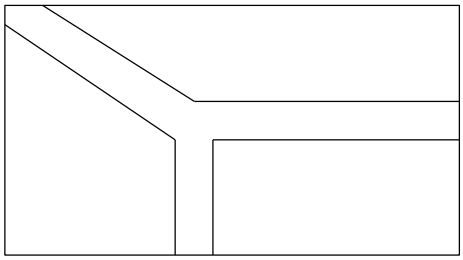


Figure 3.6: Characterisation of open HBMSs using the flag taxonomy.

The distributed version of the HyperDisco system was presented in a recent paper (Wiil and Legget, 1997). Distributed HyperDisco is composed of a set of distributed workspaces, tool integrators and third-party tools. Tool integrators provide the model allowing different (third-party) tools to be integrated and access different workspaces. A workspace is defined as an autonomous HBMS serving as a gateway to the data residing in the HBMS storage system.

In the same paper Wiil and Legget discuss two levels of distribution and the issues raised such as the management and the integrity of links which involve multiple workspaces and the need for name services. The first distribution level is local area network distribution of workspaces. For this level of distribution HyperDisco implements a replication of links strategy and a special tool integrator link class to maintain integrity of links. The second level of HyperDisco distribution is wide-area distribution over the Internet. The main idea here is that a user in one workspace, can open a tool and interact and access files from another workspace.

The SP3/HB3 model consists of applications, components, persistent selections, anchors, links and associations (Leggett and Schnase, 1994). The conceptual model underlying SP3 makes possible distribution across a wide-area network. The HB3 is a process-based system

architecture which combines the SP3 hypermedia data model, the storage manager, association manager and versioning manager to provide an infrastructure for supporting large scale hyperinformation systems. Each distributed site is running an HB3 HBMS. Associations to other HB3 sites are stored on the local storage system.

ABC was designed to support workgroups in collaborative settings developing large and complex artefacts (Smith & Smith, 1991). The emphasis of ABC's design is on modularity of system components with separate concerns. One component is the Distributed Graph Storage system (DGS; Shackelford et al, 1993). This component is the actual manifestation of the distributed design of the ABC system which can be scattered into separate workstations, each running a single DGS component. In ABC, a simple interprocess communication scheme is utilised to facilitate interoperability between components.

The short reviews presented above are sufficient to depict the basic characteristics of distributed HBMS.

- The distribution of these systems across a wide area-network is based on the distribution of different instances of the storage and data model subsystem (e.g. workspaces in HyperDisco, storage and data model manager in SP3/HB3, DGS in ABC). In other words, different instances of the same system is the primary means to achieve distribution.
- Only the first and second levels of interoperability, as these are described in Chapter 1, are supported. Recently, Wiil and Whitehead (1997) reported an ongoing experiment to support (third level) interoperability between HyperDisco and Chimera (Anderson et al, 1994). Interoperability between the two systems was limited in using Chimera as a HyperDisco workspace, while Chimera server couldn't equally access HyperDisco workspaces.
- A proprietary scheme and language is used to achieve communication between components. Protocols that can be globally used have not yet attracted attention.
- The emphasis is given on solving the information management problems (e.g. for example in the HyperDisco system, the management of links having endpoints in two distributed systems).

? Heterogeneity in terms of participating systems is not addressed. Other systems can not participate in and contribute to the hyperinformation system, which is *only* composed from different instances of the same HBMS.

3.3.2 Link Services Systems

The second thread of open hypermedia research is built around the idea of link servers (LS). A LS is an application or a process which provides hypermedia functionality to external applications which choose to communicate with the link server. A characterisation of LS using the flag taxonomy is given in Figure 3.7. The basic characteristics are that link servers explicitly distinguish between structure and contents. Also, external viewers have the responsibility for storing their data, while the LS provides only linking and sometimes anchoring information. The idea of a link service was first introduced by Sun in 1989 (Pearl, 1989) and this idea was extended by the Microcosm project after 1990 (Fountain et al, 1990). Other hypermedia link services systems are *Multicard* (Rizk & Sauter, 1992), *Proxhy* (Kacmar & Legget, 1991), *Chimera* (Anderson et al, 1994), *HyperTED* (Vanzyl, 1994).

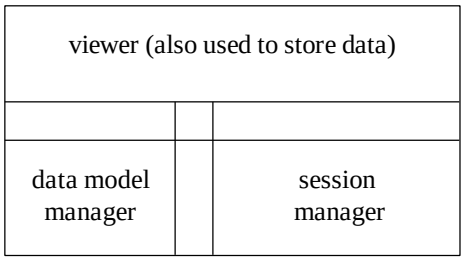


Figure 3.7: Characterisation of link servers using the flag taxonomy.

The importance of the link service idea is now widely accepted and sometimes the term is used to describe any hypermedia system which holds data separately from links. In that sense any Dexter-based system can be regarded as a link server (Davis, 1995 pp. 16). A more strict interpretation views LSS as applications which provide link functionality and handles the communication with external applications asking for link services, but they do not provide storage services to applications.

Sun's Link Service was pioneering because it first demonstrated an alternative approach to monolithic hypermedia systems. The aim was to achieve extensibility of the environment and heterogeneity of the editing tools. Its distribution capabilities derive from the fact that the Sun

workstation environment is distributed and transparently available to users. Thus, data pointed at the end of a link can be available in remote file systems. The link servers providing link management could also be distributed in remote machines. The system defines a very simple protocol to facilitate communication between link servers and editing tools. This protocol includes simple announcing and registering messages from the editing tools to the link server and in the opposite direction messages that ask an editing tool to validate or display a data object.

Proxhy (PRocess Oriented, Object-based eXtensible HYpertext) is an OHS which is based on a process and object-oriented hypertext architecture. Different components of the Proxhy system are implemented as processes which can communicate by message exchange. The main aim of the architecture was to allow third-party viewers to integrate into a hypertext application. A third-party viewer to fully integrate into a hypertext application must use a communication protocol defined by Proxhy. The Proxhy protocol consists of a fixed set of imperatives (commands) which one component may send to another.

3.4 The Microcosm Project

Basics

The Microcosm project started at the University of Southampton in 1988. The design of the Microcosm system was guided by the following design principles (Davis et al, 1992):

- to develop a system which does not impose any mark-up on data;
- to be able to integrate with any tool in the host environment;
- to develop a system which does not prevent data distribution and allows processes to be dispersed across hardware platforms;
- there should be no artificial distinction between authors and readers; and
- a system which is extensible and can incorporate new functionality.

As an OHS, Microcosm separates the data from the links. In fact, the creation of data and links is viewed as two different activities which can be totally separated (Hall et al, 1996 pp. 9). Viewers are responsible for storing data using the host file system. On the other hand the link server stores links in special files called linkbases. Viewers are classified by their level of awareness and adaptation to the Microcosm model (i.e. fully aware, semi-aware, unaware),

and different methods can be used for the integration of these viewers in the Microcosm system (Davis et al, 1994).

In the Microcosm project the idea of separating links and data is stretched even further by keeping anchor information in the linkbase rather than with the original data. This design leads to Microcosm's, the author believes, major contribution which is the concept of *generic links*. A generic link is a link which may be followed from any occurrence of a particular pattern (e.g. a text string) within any document. This is possible because link anchors are not stored within the node component, but within the link server. Using this feature the link server, at any time, from any source document and from any selected source selection, can resolve the links that can be activated. The major benefit of using generic links is that it can substantially degrade the authoring effort in linking large bodies of information (Davis et al, 1993).

In addition to generic links the Microcosm system offers other types of link: specific, local and compute links. Specific links are "normal" hypermedia links having the source anchor entirely specified. Local links are similar to generic links with the difference that may be activated only from a specific document. Compute links are based on classical IR techniques for searching an index which represents the actual documents (Li, 1992).

Process model

From a process oriented perspective, Microcosm comprises a set of autonomous processes (filters in the Microcosm terminology) which communicate by message passing (Hill et al, 1993). Messages are generated by users interacting with a viewer using Microcosm dedicated menus. When a message is generated, it is forwarded to the document control system (DCS), a central tool for organising and managing documents in Microcosm. Messages are dispatched through a chain of filters which take appropriate actions if they are responsible for the message being processed.

Messages in Microcosm are based on a proprietary format and contain a number of tags comprising the type of the tag and its value. Tags exist to indicate the content of the current selection, the type of the message/action required, the identifier of the source file and further information about the selection. The value of the action tag is the one which indicates the type of action envisaged. Typical values are 'FOLLOW LINK', 'SHOW LINKS', 'MAKE LINK' etc.

Distributed Microcosm

The first attempt to develop a distributed version of Microcosm was reported in 1994 (Hill & Hall, 1994). The architecture was based on peer to peer sharing of filters and basically is an extension to the standalone version of Microcosm. Filters in this architecture can be distributed to remote machines. The filter management system (FMS) had to be extended so it allows filters to be "published" to remote machines. Publishing is considered as the process of informing remote systems of the existence and the capabilities of local filters.

Other parts of the original Microcosm system had also to be extended to prepare for operating in such an environment. For example, the DMS was extended to distinguish documents in remote machines. The URL format introduced by the WWW was used to identify the machine where a document is hosted. Finally, the model of message processing had to be altered, so that messages are forwarded only to relevant filters. This model precludes messages being redundantly sent to filters that will eventually ignore them.

The fully distributed version of Microcosm, known as the Microcosm "TNG" (The Next Generation), is currently under investigation (Goose et al, 1996). This distributed version uses the concept of sessions. A session comprises of a set of filters and applications residing in a local or remote machines (Goose et al, 1997). A user can have many sessions concurrently open, each serving a different need.

Microcosm and agents

Recently the Microcosm team explored the role of agents in developing a distributed multimedia information system based on the Microcosm system (De Roure et al, 1996). This paper has outlined the use of agents in the Microcosm system and presented some example agents (e.g. the Advisor agent) which derived as a simple extension to Microcosm filters. Three areas have been also identified in which agents can play a role (resource discovery, information integrity and navigation assistance).

Until now, however, a generalised framework, for developing agents in Microcosm has not yet been presented.

3.5 OHSs and the WWW

A recent trend is the integration of OHSs with the WWW. The main motivation behind this integration is to enhance the functionality of the WWW via OHSs services (Anderson, 1997).

Two simple forms of OHS-WWW integration is to export data from an OHS to HTML, or to use a WWW client as a OHS viewer. The first option may be problematic because of the potential mismatch in the data models between the WWW and the OHS[7]. The second option is better but it is restricted by limitations in current WWW clients (Grob? k et al, 1997).

A more advanced level of integration can be achieved if a WWW server uses an OHS server. Using this approach a WWW server is modified and makes calls to the OHS server in response to specific URLs. On the fly the OHS server combines the data and the links, in order to construct the HTML page which will eventually be presented to the users. The benefits are the separation of links and data, and the use of the OHS as the authoring environment. The Distributed Link Service (DLS) is a service performing this type of integration for the Microcosm OHS (Carr et al, 1995). Similar types of integration between OHSs and the WWW have been also reported by Anderson (1997a) and Grob? k et al (1997).

The most natural way, however, of integrating an OHS to the WWW is demonstrated by Hyper-G and Webcosm[8], a commercial product based on the DLS. In this integration method the OHS server handles directly requests from WWW clients. Of course, this is the most efficient way because the OHSs have the flexibility to interact directly with the users.

3.6 The OHP Protocol

In order to deliver hypermedia functionality to third-party applications, OHSs communicate with them using proprietary protocols. The Open Hypermedia Protocol (OHP) is a peer to peer asynchronous protocol attempting to address the need for a common and standard protocol serving the communication between OHSs and third-party viewers (Davis et al, 1996). The OHP delivers the distilled experience from Multicard's 2000 protocol (Rizk and Sauter, 1992) and Microcosm's message model. The OHP protocol comprises a set of messages (e.g. LaunchDocument), a set of data types and a specification of the legal message format.

---

[7] for example, the problem of exporting Microcosm's generic links to the WWW is discussed in Hall et al (1996, pp. 136).

[8] there are three different ways to integrate Webcosm in the WWW. One of them which is “using the Webcosm as a standalone server”, is the one which corresponds to this type of integration (i.e. OHS server as WWW server).

The OHP protocol also recognises the fact that most OHS have privately developed protocols which are internally used and, therefore suggests an intermediate component, which translates messages written in OHP to the native formats of individual OHS. This component is called a protocol shim. Two shims are defined in the OHP (i.e. one for the OHS and one for the viewer), each residing between the sender and the recipient of a message appropriately translating messages. Each OHS has to produce its own shim so that it is afterwards able to "understand" messages in OHP.

The form of the protocol is quite simple. An OHP message is a text stream comprising of tags following by a backslash and a space character. Everything following the space up to the next tag is the content of the message.

OHP has not gone without criticism. Anderson (1997b) has critically evaluated OHP. The critique which is presented in that paper can be summarised by the following points:

- there are some syntactic inconsistencies in naming messages in the OHP (e.g. between the messages *LaunchDocumen*t and *CloseNode*);
- the semantics of the services provided by OHSs are unspecified;
- much of the behaviour and procedures during interoperation are left unspecified;
- the protocol does not provide any mechanism for an OHS to define which are the correct set of parameters;
- the user interface of the viewer is somewhat redefined by the OHP;
- OHP leaves unspecified how a viewer discovers and contacts an OHS shim.

The most important point, however, which is not sufficiently addressed by the OHP, the author believes, is that it does not support interoperability between components other than the link servers and viewers. The OHP basically addresses interoperability only between link servers and viewers. Wiil and Whitehead (1997) have also stressed that issue in describing an ongoing interoperability experiment between HyperDisco and Chimera. Other researchers in an effort to develop a reference architecture for OHSs, have identified this issue and called for additional protocols (Goose et al, 1997; Grob? k & Wiil, 1997).

Generally the OHP can be characterised as an initial step towards achieving interoperability in OHS. However, there are some issues that it will be very difficult to address within the OHP without revising or rethinking the protocol. In later chapters, a new, more complete protocol

for OHS interoperation will be presented. This protocol addresses the issues pointed out by Anderson and establishes a better framework for OHS interoperation and co-ordination.

3.7 Discussion

It could be said, that OHSs largely satisfy many of the architectural dimensions and requirements presented in Chapter 1, and therefore are promising candidates as supporting technologies for HDLs. For instance, in terms of distribution, OHSs have been reviewed which satisfy both types of data and services distribution. They are generally quite extensible and new tools and data can be easily incorporated into an information workplace based on an OHS.

On the other hand, despite the exposition of some recent exceptions (e.g. Hall & Davis, 1997) and the relatively broad range of different links (e.g. generic, computed) to search for information in Microcosm, it could be generally said that OHSs research until now has primarily focused on methods for enabling document viewers to participate in a common linking environment (Nurberg & Legget, 1997). The review of OHS literature clearly indicates that less emphasis has been given on considering OHSs as information seeking environments and to integration issues other than the integration of viewers. Needless to say, that the central issues from this point of view are quite different from the architectural and the viewer integration issues which dominated OHS research until now.

One could assert that the issue of information seeking in HDLs is an entirely different problem which is the subject of other computer science disciplines (i.e. IR) and, therefore should be kept outside the agenda of OHS research. Partially this is a legitimate view. On the other hand, if someone wants to categorise OHSs, they should be clearly regarded as informational hypermedia systems and, therefore information seeking must be seriously considered. Also, there are interactions and dependencies between how an OHS system is designed and, how information seekers can ultimately search for information in this OHS. Nevertheless, it should be expected that the development of HDLs will touch on many areas. This view is expressed by Fox et al (1995a, pp. 26) in the guest's editors introduction to a special journal issue on digital libraries. In the section discussing the different areas which are involved in the development of digital libraries they say:

> "... Each area can be studied on its own, but special insights are gained by considering an area in the context of digital libraries. Digital libraries that will

be developed with all of these topics properly considered will come much closer to providing full service."

One obvious way to increase the performance of an information seeking environment is to increase the effectiveness of the system's searching algorithms which implement the indexing or the actual searching of document collections. This approach has been traditionally the focus of conventional, system-centered IR and it is not the subject of OHS research.

On the other hand, although the discovery of new information retrieval algorithms could be said to be the subject of IR research, the actual integration and application of these algorithms in electronic environments such as HDLs is a different issue that should be considered by OHS research. For example it will be shown later in the thesis, that the integration of the tools implementing a collection fusion strategy reveals different issues from the actual invention of the fusion strategy itself. It is exactly this point which supports the claim that information seeking should be considered in OHS research. Of course, this requires broadening the focus of OHS research and considering architectural issues and integration problems which are different from the integration of viewers.

But, is it only the problem of integrating information seeking tools which could be addressed by OHSs? There are additional arguments which raise the hopes that OHSs could be effective information seeking environments. These hopes mainly relate to methods that have been reported in the literature for increasing the performance of information seeking environments. Rao et al (1995) suggest that information seeking performance could be increased in rich information workspaces. A rich information workspace is an environment designed to support a variety of information seeking strategies. For example, the scenario for OHSs which was recently presented by Hall and Davis (1997) can be regarded as an example of a rich information workplace. In this scenario different tools in the form of intelligent agents are used for information seeking. This Ph.D. work shares the same view that the open architecture of OHSs may be used to integrate different information seeking tools.

Hendry and Harper (1996a) report another, quite different, approach for increasing effectiveness by designing informal and flexible architectures for information seeking environments. Flexible architectures may be more adaptable to different information seeking practices, hence information seekers can customise their information workplaces. Also, flexible architectures are useful in developing extensible information seeking environments (Hendry & Harper, 1996b). An information seeking environment which is extensible can

provide a variety of different tools and search strategies. This variety of tools and strategies offers to the users the opportunity to engage in rich interactions during the information seeking process. Some of the work which is reported in this thesis is driven by the belief that OHSs can provide the architectural framework to develop the informal, customisable environments suggested by Hendry and Harper.

The other large issue which particularly arises is the lack of a protocol which supports interoperation in all the possible levels presented in Chapter 1 (section 1.3.1). Existing methods for interoperation used in OHSs are based on proprietary protocols and methods. In general, it could be said that in OHS research until now *architectures are superior to protocols*. Each OHS research effort strives to design an architecture which differs in some points to other OHS architectures and protocols are basically sidelines of the research effort.

The OHP is the only protocol which addresses the issue of a standard protocol for OHS interoperability. However, OHP only addresses a small sub-set (i.e. interoperability between link-server and viewer) of the full possible range of interoperability.

Chapter 4

The Collection Fusion Problem

This chapter classifies and reviews different collection fusion strategies. Source selection and search result production are two important subprocesses in most information seeking activities. Collection fusion is the term used to specify the problem of source selection and result merging in distributed information environments. An argument is made in this chapter that in considering digital libraries, special interest should be given to collection fusion strategies, because they can directly affect the effectiveness and efficiency of information seekers.

4.1 Introduction

Information seeking comprises of a number of factors and processes. Typical factors are the information seeker, the task in hand, the search system used to conduct the searching, the subject domain and the setting of the searching (Marchionini, 1989; Marchionini & Schneiderman, 1988). The information seeking process is an iterative activity which is composed of different subprocesses. Figure 4.1 depicts the main subprocesses and possible transitions as they are identified by Marchionini (1996, pp. 50). This figure illustrates how the information seeking process starts by recognising[9] and defining an information problem. The next step is to *select a source* which is likely to contain information satisfying the recognised problem. The next steps are to formulate, execute a query and *examine the results* to extract relevant information.

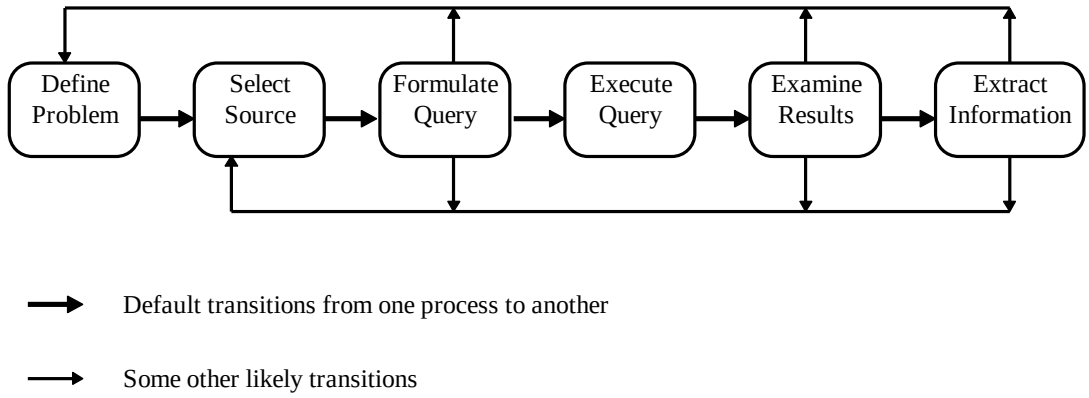


Figure 4.1: Information seeking sub-processes and basic transitions.

The term *collection fusion problem* is used to delineate the problem of distributed IR residing in the second and fifth subprocess of the information seeking framework depicted in Figure 4.1, i.e. the selection of a source likely to provide relevant information, and the production of search results so they can be effectively examined by information seekers. In conventional information seeking environments the searching process is confined to a single collection, so

[9] in terms of Belkin's (1980) information theory information seekers recognise an anomalous state of knowledge, i.e. a gap in their knowledge which does not allow them to solve the information problem.

apparently the source selection problem is trivial. Likewise, the problem of preparing the search results for user examination is relatively less complicated, since all the candidate documents come from a single collection.

Digital libraries, however, have multiple collections and users face the problem of source selection. Collections can be relatively homogeneous, as in the case of a single large collection partitioned and distributed over a, usually local, network, but they can also be extremely heterogeneous where tens or hundreds of different collections are available, over a wide area network, for searching. In theory, one could decide to search all collections, but this will be generally too expensive both in terms of time and computer resources (Callan et al, 1995). Hence, somehow a decision should be made and a list of collections must be selected. Then, individual searches will be conducted and results from disparate collections must be presented for examination.

It could be argued, that users can manually do the source selection. Lynch (1997) explores this. However, there are many convincing reasons for opposing this approach. First, information seekers may be unable to make selections, especially in large and dynamic environments. Second, in large electronic environments (like digital libraries) where thousands of sources may be available, the problem of manual source selection will be extremely difficult (e.g. scanning through a large list of available sources).

Voorhees et al (1994), concisely characterises collection fusion as the *data fusion problem* in which the results of query runs in different, autonomous and distributed document collections must be merged to produce a single, effective result. Collection fusion differs from other data fusion research efforts seeking to combine multiple evidence from different runs of the same query in order to increase the effectiveness of that query to a *single* collection (e.g. Lee, 1995). On the other hand, the goal of a collection fusion technique is to combine the results from *multiple*, separate document collections into a single result without degrading the effectiveness, if possible, of searching the entire set of documents as a single collection.

4.2 More about Collection Fusion

Centralised vs. Distributed designs

The collection fusion problem arises as a result of dealing with multiple collections. One can legitimately assert that instead of trying to solve this problem, a simpler possible alternative is to maintain a centralised index built by systematically and exhaustively downloading and

indexing the documents from remote collections to a central repository. In fact, several systems based on this approach have been developed (Bowman et al, 1994; Maly et al, 1994, Moffat & Zobel, 1996). Some of them have been reviewed in Chapter 2 (e.g. Harvest and WATERS systems). However, this Ph.D. work advocates distributed rather than centralised index designs for the following reasons.

- Maintenance of a central representation is very time and network resources consuming, because any change to a document provokes a potential update of the central repository. Clearly, only partial updating is practicable, and thus at any moment the central index will not necessarily accurately represent the current state of the document collections (Ellman & Tait, 1996).

- Requirements such as reliability and availability can be better met by distributed designs (Viles, 1994). This claim seems to be well justified, considering for example that in a system based on a centralised design failures in the central system will make the services provided totally inaccessible. On the contrary, possible failure of a system in distributed design affects only marginally the services provided. Additionally, requirements such as modular growth and local autonomy of information sources can only be met by distributed architectures. Autonomy and modularity are essential for heterogeneity and extensibility.

- There will be many cases where the original owners of the information repositories may not want to give up control of their data, hence the development and maintenance of a centralised repository is prevented

Another reason for advocating the development of distributed information seeking environments is the so-called *cluster hypothesis* (Jardine & van Rijsbergen, 1971). This hypothesis is based on statistical observations and is stated as (van Rijsbergen, pp. 46, 1979): "closely associated (in content) documents tend to be relevant to the same information needs (requests)". A claim has been made that if systems exploit the cluster hypothesis they can increase the effectiveness and efficiency of retrieval.

So, based on the cluster hypothesis, it could be argued that distributed index designs can be more effective and efficient than centralised index designs. Indeed, if the whole corpus is considered as a single collection, then distributing documents to different "thematic" collections is, in effect, a first step of a clustering process which, according to the cluster hypothesis, is likely to increase the effectiveness and efficiency of information seekers.

Efficiency is increased because searching can be focused into a smaller number of narrower (according to subjects covered) document collections.

The cluster hypothesis should particularly hold in electronic environments which support the notion of personal digital libraries (PDLs) presented in Chapter 1. In such environments documents are not randomly produced, stored and indexed. Each user who "owns" a PDL will usually store documents in a relative small set of subject domains (in comparison to the existing subjects in the whole corpus).

Consider for example, the digital library of a university which comprises different departments. Members of a particular department (e.g. fine art) will be expected to produce, store and index material which is different from the material that it will be produced by the members of the computer science department. Even within a single department, each member will usually store documents about a particular subject domain (e.g. neural networks) than other members in the same department which work in other subject domains (e.g. multimedia). If a single index is produced for all the departments and individuals in the digital library of this university, this kind of differences which can be exploited by clustered information seeking methods will be lost. In opposite, if PDLs are supported and individual indexes are produced, clustered methods for searching (e.g. clustered browsing or clustered analytical searching) can be applied.

The cluster hypothesis should also hold in hypermedia digital libraries which provide the means and explicitly encourage authors to cluster documents (e.g. composites in Dexter-based systems). Nonetheless, it is a sign of good authorship that closely related documents are clustered together. It can be assumed that authors follow the basic principles of information management to collocate and cluster together similar documents.

Classification of collection fusion strategies

Collections fusion strategies can be classified using two criteria. The first criterion is based on the necessity or otherwise of a learning phase before a collection fusion strategy can be utilised. The second criterion classifies fusion strategies into two categories: those using only the ranked list of documents returned to produce the single result, and to those using additional data from remote collections to merge the results.

Collection fusion methods requiring a learning phase before they can be utilised are clearly less convenient. Learning may involve the computation over large amounts of data and

therefore be time consuming. It will also usually require some sort of training data to facilitate the learning phase. Furthermore, if the learning phase produces data which directly or indirectly relate to the content of documents, any change to the content (e.g. add new documents or change existing ones) will gradually invalidate the data produced after the learning phase. Frequently, a new learning phase will need to be conducted again in order to regenerate a new set of (representative) learning data.

The second criterion refers to the type of information required by a fusion strategy for the merging of the individual results (Voorhees, 1996). We call isolated merging strategies the strategies producing the single merged result without using any run-time information from remote collections except the ranked list of documents returned from individual collections. On the other hand, we call integrated merging strategies the strategies which have access to additional information (e.g. collection wide word frequencies) in order to merge the results from multiple document collections.

Since integrated strategies have access to additional information, they can be expected to be more effective than isolated strategies. The shortcoming is that integrated strategies demand larger network resources since additional information must be exchanged and, may involve more steps which potentially makes them less efficient.

Collection fusion and browsing

Source selection is also an important process when using browsing strategies for information seeking. In browsing strategies the subprocesses depicted in Figure 4.1 proceed in a more parallel fashion and more time is spent on the examination of results than analytical strategies. Clearly, the effectiveness of the methodology for producing the results to be examined by information seekers, directly affects the effectiveness of browsing. The challenge is to produce the results in a such a way that users will be assisted to make the best selection for their current information problem.

Browsing also offers significant challenges for information seekers and designers regarding the source selection problem. Information seekers will normally consider source selection during browsing as a method for selecting entry points for browsing. This will mostly happen at the beginning of an information seeking process. Certainly, entry points can be randomly selected, or by opportunistic and iterative examination. However, these methods are usually inefficient and, systems should provide automatic methods for suggesting entry points to information seekers.

4.3 Source Selection Strategies

4.3.1 Isolated strategies

Modelling relevant document distribution (MRDD) and query clustering (QC)

Voorhees (1996) reports two fusion strategies which require training queries. The basic characteristic of these two strategies is that they need a set of training queries and require a learning phase before they can be utilised.

The MRDD fusion strategy uses a set of training queries explicitly to build a model of the distributions of the relevant documents in the individual collections. For each new coming query, the model of relevant document distribution is applied in a maximisation procedure to obtain the number of documents to retrieve from each collection. The QC fusion strategy learns a measure of the quality of search for a particular topic area of the collections. Then, the number of documents retrieved for a new query is proportional to the quality measure computed for this new query based on past training queries.

Both strategies have been evaluated using the TREC-3 experimental collection (Harman, 1994). For the MRDD strategy, Voorhees et al (1995) report an average 8% degradation for precision when all TREC-3 subcollections were used, and an average 23% degradation for precision using only three of the subcollections. Using the same method in TREC-4 (Harman, 1996), they report an average 38% degradation in precision. For the QC fusion strategy, they report an average 7% decrease for precision using all of the TREC-3 subcollections, and an average 25% precision degradation for precision using three of the subcollections. Applying the same method in TREC-4 they report an average 29% degradation in precision. In summary, both learning fusion strategies reduce precision from the levels obtained with the single collection fusion strategy.

The Uniform approach

The most obvious approach to the collection fusion problem is to assume that each distributed collection has the same number of relevant documents as the others. In other words, it assumes that the relevant documents corresponding to a particular query are identically distributed across all document collections. Hence, an equal number of documents should be requested from each collection. For example, if someone wants to retrieve 10 documents from

a digital library with 10 separate collections, 1 document from each distributed collection should be retrieved using the uniform approach.

In practice, this is usually a bad approximation because documents which are relevant to a particular query are usually collocated in a few collections. In fact, heterogeneous digital libraries collections have specialities and they hold documents of a particular interest.

Voorhees et al (1995) characterise the uniform approach as the baseline fusion strategy against which the effectiveness and efficiency of any other fusion strategy should be compared.

Optimal fusion

The optimal fusion strategy can be only applied when the actual relevant document distributions are known and used to determine the cut-off levels for each document collection. So, the optimal method is a retrospective technique that gives an upper bound for the effectiveness and efficiency of any collection fusion strategy. Table 4.1 shows how distributed collections would be selected in a hypothetical digital library system for a particular query Q when the optimal and the uniform strategy are applied. This simple example clearly demonstrates the effectiveness of the optimal strategy in requesting the most appropriate number of documents from different collections.

The random approach

Using this simple strategy the number of documents requested from each document collection are randomly determined. Obviously, this technique produces less effective results in comparison to the other fusion techniques. However, in many cases this technique is the only available one in searching multiple, distributed collections. Surely, the uniform approach is also always available but in large libraries it might be inefficient to retrieve documents uniformly from all the sub-libraries available.

Digital Library X			
Collection	*Number of Relevant documents to Q*	*Allocations if the information seeker wants **20** documents to be retrieved*	
		Optimal Fusion	***Uniform strategy***
A	3	12	4
B	0	0	4
C	0	0	3
D	2	8	3
E	0	0	3
F	0	0	3
Libraries involved		***2***	***6 (all)***

Table 4.1: An Example of using the optimal fusion and the uniform collection fusion strategies.

4.3.2 Integrated strategies

Source selection based on Inference networks

Callan et al (1995) have presented an integrated strategy for source selection using inference networks. Inference networks represent a probabilistic approach for conventional information retrieval (Turtle & Croft, 1991). This approach is used to rank document collections instead of documents in a single collection. In the case of source selection, the arcs in the inference network represent statistics analogous to the statistics used in conventional IR like the term frequency (*tf*) and inverse document frequency (*idf*) (Salton & Buckley, 1988). For example, statistics used in Callan's experiments are the *df* (documents containing a term) and *icf* (the number of collections containing the term).

Using variations of this merging strategy, experiments were conducted using the TREC-4 test collection (Allan et al, 1996). For the different variations of the method the average degradation for precision in comparison to the single run was from 13.4% to 26.8%.

Other fusion methods

EXPERT CONIT was another retrieval system which used a rule-based inferencing system to take automatic decisions about source selections (Marcus, 1983). It decided on a query by query basis which collections are more likely to contain relevant documents for a particular query. Another system is GLOSS (Gravano et al, 1994) which produces an estimation of the relevant documents in a collection by using some collection wide statistics. It is efficient because it stores only term frequency information about collections.

Finally, Fuhr (1996) reports a theoretic approach for optimum database selection in networked probabilistic IR systems. The method uses a function which gives the time and network costs for retrieving a number of relevant documents from a specific database. Although, this method can produce optimum results, in practice it is inapplicable because it requires information which can be only estimated (i.e. number of relevant documents and recall-precision function)

4.4 Merging Results

Interleaving results

A simple method of imposing a single ordering to documents retrieved from multiple collections is to interleave the results. This method is applied when the only information used is the individual ranked lists (i.e. in isolated methods). A simple interleaving method is to use a C-faced die to interleave the results (Figure 4.2). Using this method an equal number of documents are selected from each successive collection and placed into a single ranked list, until all documents from individual ranked lists are selected.

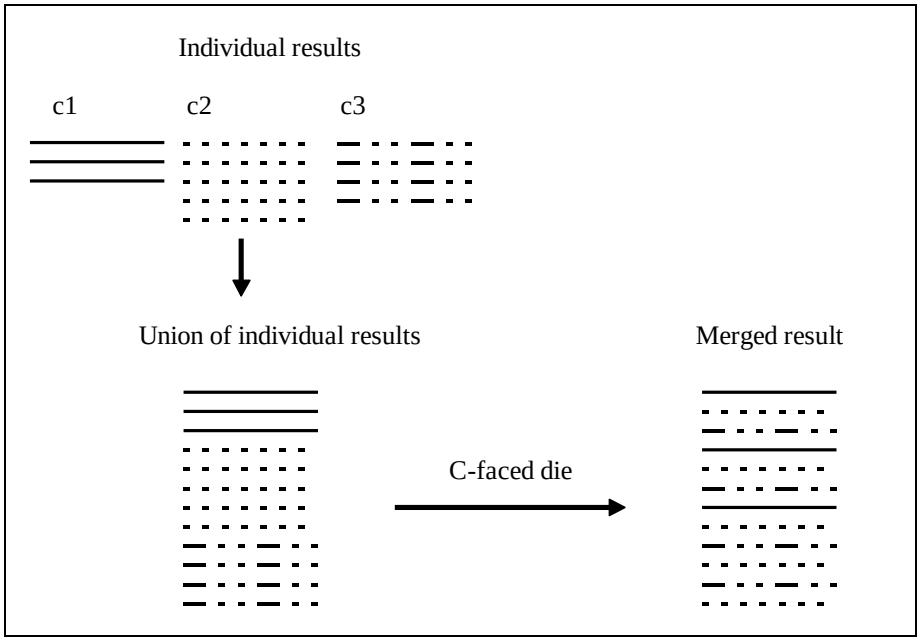


Figure 4.2: Merging documents using a C-faced die.

Another interleaving method is to use a C-faced die biased by the number of documents still to be picked from each collection (Figure 4.3). Using this method the collections are not treated equally. The collections which have more documents to contribute, place their documents first in the single ranked list. This method has the advantage that it respects the ordering produced by collections and gives a preference to collections contributing more documents. The assumption is that collections which contribute more documents are likely to contain more relevant documents.

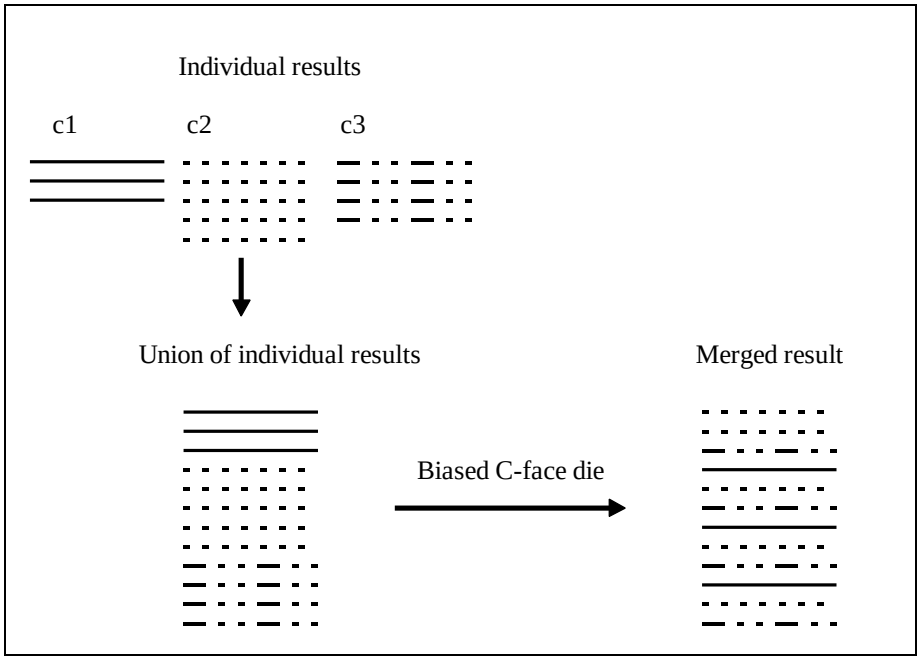


Figure 4.3: Merging documents using a C-faced die biased by the number documents still to be selected.

Comparing similarities across collections

Another approach to merging results is to assume that the similarity values across documents are comparable and to select the documents with the higher similarities values in higher ranks. However, this approach is inapplicable because of the requirement for collection dependent measures such as *idf* which are not comparable across multiple collections. This problem can be overcome if the statistics such as *idf* can be normalised for the set of collections being searched (Kwok et al, 1995). However, this normalisation involves significant communication and computational costs which make it impractical.

4.5 Discussion

It is to be expected that the need for methods solving the collection fusion problem will increase as the number of new information resources and networked information environments proliferate. Recent user studies indicate that multi-database searching is very much accepted by information seekers. In fact, Lynch (1997) reports that 88% of the 43 participants in his study consider multi-database searching as a highly desirable feature.

Multi-database searching, however, should be approached with caution. For example, a multi-database search which treats all the collections the same (i.e. based on a uniform approach),

will probably not be very efficient. Conversely, searching all the available collections for every information need is likely to be inefficient. Also, the uniform approach will probably increase the information overload for the information seekers who will receive and have to examine results from many, possibly heterogeneous, sources.

Some of the fusion strategies reviewed in this chapter, do not focus on source selection (e.g. fusion with inference networks). Instead, they retrieve documents from all available distributed collections and concentrate on the effective merging of the results. This clearly has a negative effect on efficiency. It is probably more desirable to see an appropriate balance between increasing effectiveness and minimising the number of libraries involved (i.e. efficiency). In fact, when this approach was tested (i.e. effort to minimise the libraries involved), the effectiveness results were little affected (Callan et al, 1995).

Multiple searching particularly should be treated with caution in digital libraries. First, because these electronic environments support electronic publishing leading to rapidly changing information. This has an important consequence for fusion strategies using a learning phase: the data produced from the learning phase can quickly become obsolete, in a sense that they do not accurately approximate the environment. If repetition of the learning phase is time consuming then this fusion method is impractical.

Second, digital libraries as they are envisaged in this thesis, are heterogeneous environments with no central authority. Therefore, it is not realistic to presuppose (as the integrated fusion strategies do) that additional information beyond the ranked list of documents can be provided. For example, some of the methods reviewed in this chapter, require individual results to be supplied together with numerical relevance scores. However, in heterogeneous environments retrieval systems conducting the actual searching may not be able to produce and therefore to provide relevance scores (e.g. Boolean based IR).

Finally, there are some architectural issues that should be addressed if multi-database searching is considered. Most of the methods and systems presented in this chapter assume an ideal setting where at any time all collections are accessible. Even further, they implicitly assume some sort of coordination between information servers in disparate collections. But, as it was shown in Chapter 2, existing standards and protocols (e.g. Z39.50) lack support for multi-database searching.

An appealing solution is the development of architectures and protocols which take into account the complexities of these requirements. In the next chapter a novel link-based

collection fusion strategy is presented. In Chapters 6 and 7 an OHS architecture and protocol are presented which, amongst other goals, can address the architectural problem of integrating the collection fusion strategy into a real information seeking environment.

Chapter 5

A Link-Based Collection Fusion Strategy

This chapter presents a novel idea for solving the collection fusion problem in hypermedia digital libraries. The proposition which is explored and evaluated is that across document links in distributed hypermedia collections can supply useful information which can be utilised for providing an effective and efficient solution to the collection fusion problem. In contrast to other methods reviewed in Chapter 4, the link-based fusion strategy does not require a learning phase before it can be utilised and, also does not use any information from remote collections other than the returned list of documents. Systematic evaluation of the link-based fusion strategy demonstrates that the proposed strategy is more effective and efficient than other fusion strategies that can be applied under the same conditions.

5.1 Using Links as a basis for Collection Fusion Strategies

From the early days of computerised IR research, relationships between documents based on bibliographic links have been utilised for a variety of reasons (e.g. Salton, 1971). More recently, links have been utilised in different settings in order to increase the effectiveness of information retrieval. Savoy (1996) developed an extended vector processing scheme using additional information expressed by bibliographic links, in order to increase the effectiveness of retrieval in hypermedia collections. Frisse (1988) suggested an extended vector processing model which uses hierarchical links to search an electronic medical handbook. Frei & Stieger (1992) have also suggested a vector space model accounting for different types of links. Links have also been used to make possible the retrieval of multimedia objects based on the content of associated text documents (Dunlop and van Rijsbergen, 1993). Turtle & Croft (1991) also incorporate links in a probabilistic IR model to enhance retrieval effectiveness.

The research efforts mentioned above, aimed to increase the effectiveness of conventional IR algorithms by utilising links between documents in order to get additional word and document statistics about the documents to be indexed. The idea which is presented in this chapter is novel because for the first time links are exploited for solving the collection fusion problem. The ultimate goal is to increase the effectiveness and the efficiency of the information seeking process in dynamic hypermedia digital libraries. The method presupposes the presence of hypermedia links and can be utilised in any hypermedia environment comprising different distributed hypermedia collections (Figure 5.1).

The proposed fusion strategy exhibits two features not found in the collection fusion methods reviewed in Chapter 4. These two characteristics make the method practicable and applicable to dynamic and large information seeking environments such as hypermedia digital libraries.

First, it solves the source selection problem solely by the use of linkage information extracted from local linkbases. Information (other than the ranked list of documents) from remote collections is not needed at run-time or at any time before to select the sources. In that respect, our collection fusion strategy is an isolated collection fusion strategy and, as such has the advantages that have been discussed in sections 4.2 and 4.5.

Second, the proposed link-based fusion method does not require any learning phase to be undertaken before it can be utilised. That makes the proposed method applicable to dynamic environments in which fusion strategies (e.g. the MRDD and QC reviewed in last chapter) requiring an expensive learning phase are practically inapplicable.

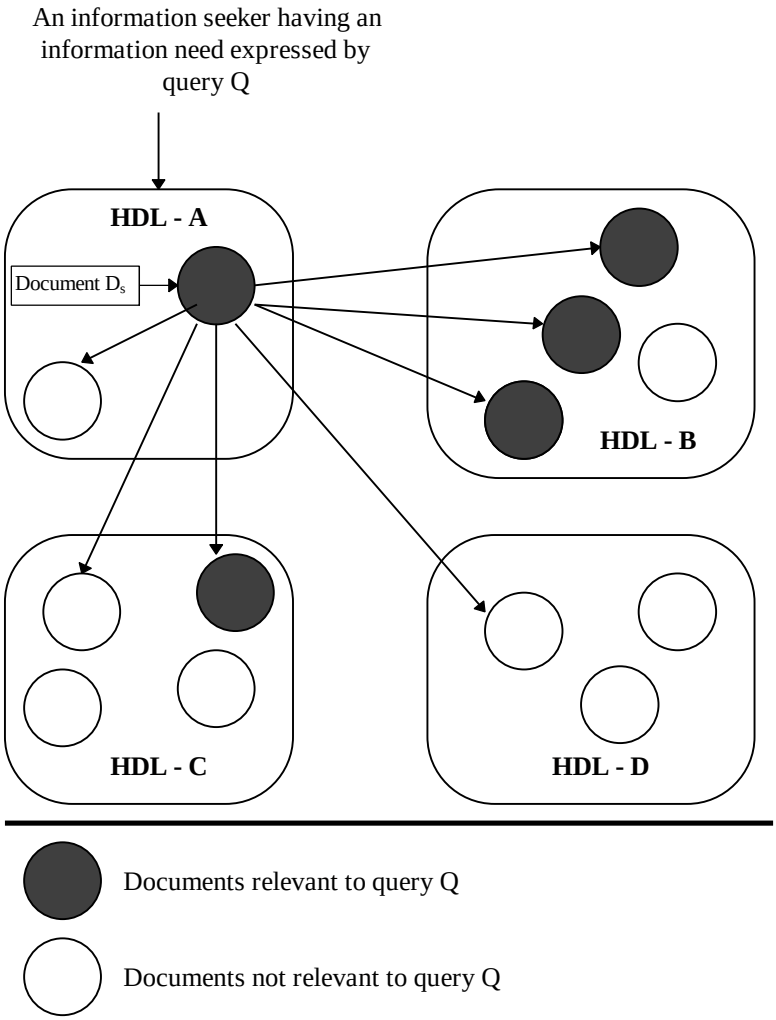


Figure 5.1: An overview of an environment which the link-based fusion strategy can be applied and a simple example of the link hypothesis.

5.2 Methodology Explained

The defining feature of the link-based fusion strategy is that it utilises information extracted from a linkbase to determine the distribution of relevant documents in remote hypermedia collections. The distribution of relevant documents is approximated using the distribution of hypermedia links having their starting points in relevant documents retrieved from a *sampling*[10] hypermedia collection at the beginning of each search process.

[10] adjectival complement is used to indicate that the collection which is used for sampling can be different in each search process and can be constantly changing.

The rationale of the method is based on a hypothesis analogous to the cluster hypothesis mentioned in Chapter 4. This hypothesis has been implicitly used in the works that were outlined in the first paragraph of this chapter. A small contribution of this Ph.D. work is to explicitly identify the use of the *link-hypothesis* and also stated as (see also Figure 5.1):

"closely interlinked documents tend to be relevant to the same information needs"

Hence, if the link hypothesis holds for a particular hyperinformation environment then discovery of one or a few relevant documents can accommodate the discovery of more relevant documents by following or otherwise exploiting links found in the relevant retrieved.

The link-based fusion strategy is based on the link-hypothesis and operates in three separate steps (Figure 5.2). First, given an information need expressed by a query Q, an initial search is made using Q to retrieve the most relevant hypermedia documents from a sampling hypermedia collection C_s. Search results are processed to extract the links having as their starting points one of the hypermedia documents retrieved from C_s. Second, linkage information is passed to a maximisation function which determines (based on the linkage information and running an approximation algorithm) the distribution of relevant documents in other remote collections. This approximation together with the total number of documents to be retrieved (i.e. the cut-off level specified by the user), is used by the fusion method to calculate the number of documents that must be requested from each collection. Third, individual parallel searches are conducted and finally the separate results from individual collections are returned back as a response. Finally, individual results are merged and clustered to produce a single result.

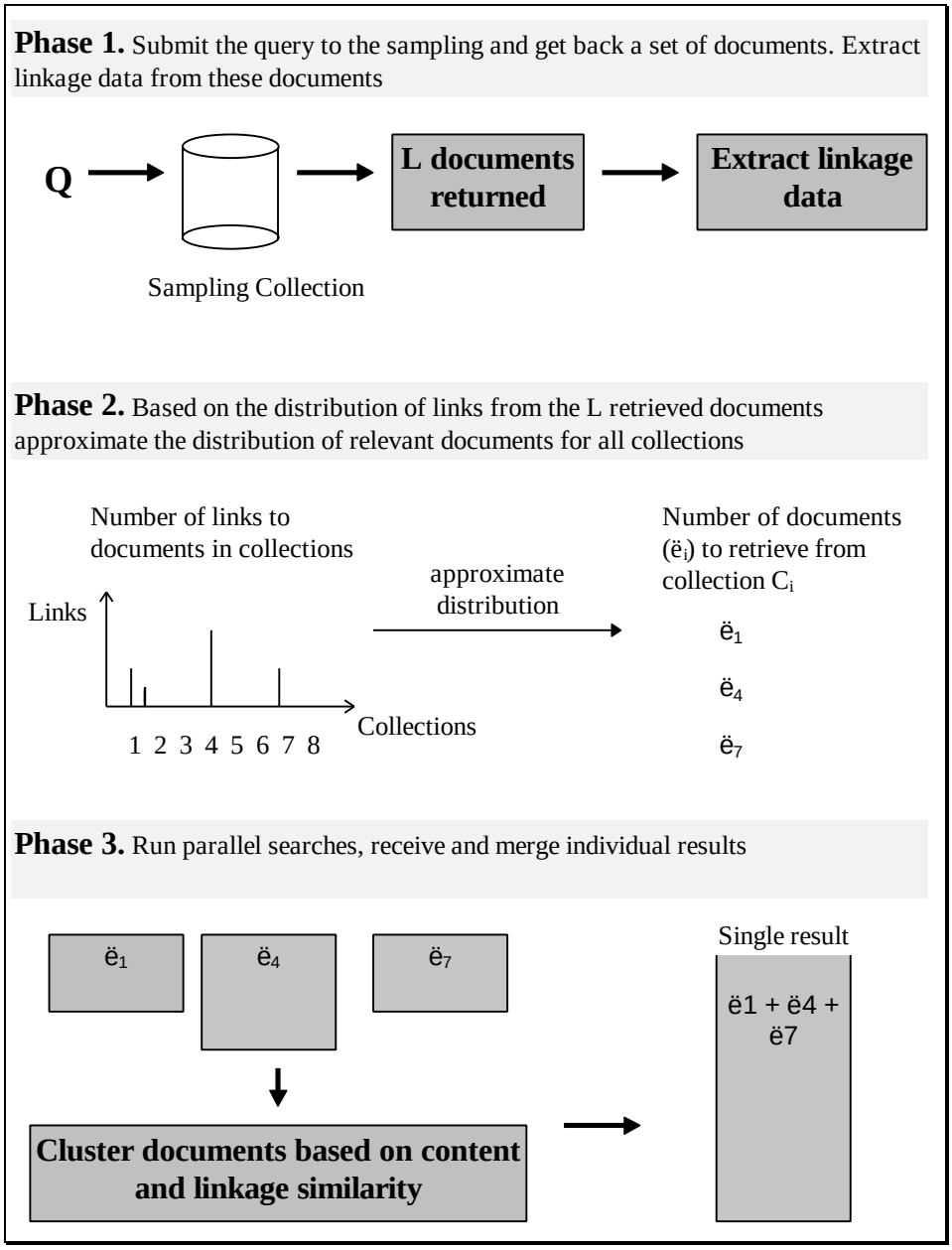


Figure 5.2: The three phases of the link-based collection fusion strategy.

5.2.1 First Phase: Extraction of Linkage Information

Selection of the sampling repository

The first phase of the method starts by submitting the query Q to a sampling hypermedia collection in order to retrieve the L most relevant documents[11]. In this step a document is regarded as relevant if it is in the top L retrieved documents. At this stage, the first issue is how to select the sampling hypermedia collection. In fact, in a large hypermedia digital library comprising many participating collections multiple choices for selecting the sampling repository are available. Because this initial run is only for extracting and feeding data the approximation function and not to retrieve relevant documents, the most efficient choice is to select the collection which is less costly to access, query and retrieve the sampling results.

Generally, this will be the collection residing in the digital library hosting the information seeker initiating the searching process (i.e. HDL-A in Figure 5.1). If this is not feasible, other remote collections can be equally used to extract linkage information. Of course, this second alternate choice will be apparently less efficient because it involves message and data exchange over a wide-area network.

Extracting the linkage information

The next step of this phase is to extract and calculate the number of links having as their starting points one of the L documents retrieved from the sampling collection. At this stage, another issue which arises is the selection of the types of links that will constitute the linkage information. This is a decision affecting the final output of the fusion method, because linkage information will be passed in the second phase of the method to the function approximating the distribution of relevant documents in the digital library.

Of course, this is a single solution problem if there is only one type of link which can compose the linkage information (e.g. like in the WWW). However, most hypermedia systems support links of many different types. In these cases, a decision must be made about the types of links

---

[11] note that this process should not be confused with the learning phases required, for example, in MRDD or QC methods. Retrieving documents from the sampling is a process which can take place *on-the-fly* when a user submits a query because it involves only *one* collection (i.e. the sampling collection). The learning processes of MRDD and QC must take place *before* any search can be made and it is an expensive process which involves *all* the sub-collections.

that will be used to extract linkage information. The link hypothesis implicitly suggests the types of links that should be considered. The best candidate links are those expressing content similarity between hypermedia documents.

The type of links to be included is mostly a qualitative problem which should be resolved taking into account the intended semantics of different types of hypermedia links. There is another more quantitative issue which needs examination: the number of links to be extracted and finally passed to the approximation function in the second phase of the fusion method. Clearly, the number of links is proportional to the number of documents retrieved from the sampling collection and may affect the performance of the method. Extraction of a large number of links increases the possibility of more libraries to be finally selected for the distributed searching. In principle, more libraries means increased effectiveness, but also decreased efficiency should be anticipated. Effectiveness will be increased normally because more libraries are searched. On the other hand, efficiency decreases because searching more libraries normally causes larger communication and time costs.

Apparently, the three issues (i.e. selecting the sampling repository, types of links considered and number of documents retrieved from the sampling hypermedia collection) discussed in the paragraphs above, can become parameters of the link-based fusion strategy. This can give the choice to the system designers and information seekers, or to the method itself to manually/automatically adapt the link-based fusion strategy to the information environment and to the information seeking task at hand.

5.2.2 Second Phase: Approximation of Relevant Documents Distribution

The second phase of the link-based fusion technique takes as an input the distribution of links starting from documents retrieved from the sampling collection and ending at other hypermedia documents. For example, the links distribution for query Q in Figure 5.1 is {HDL-A:1, HDL-B:3, HDL-C:2, HDL-D:1}, if only the document D_s was retrieved during the sampling process. This information is passed to an approximation function which determines the number of documents that should be requested from each collection, given that T documents should be retrieved in total. For example, a simple approximation function can allocate to each collection C_i a number of documents which is proportional to the number of links pointing to documents in C_i. The expectation (according to the link-hypothesis) is that

the number of relevant documents in C_i is proportional to the number of links pointing to documents in C_i.

For instance, if T is the total number of documents that should be finally retrieved from all collections, the set of link frequencies are used to apportion the retrieved set such that when T documents are to be returned and L_i is the link frequency for collection *i*, the number of documents to be retrieved by each collection is determined by the formula:

$$\frac{L_i}{\sum_{i=1}^{N} L_i} * T = \text{Number of documents to retrieve from collection } i \text{ } \textit{(rounded appropriately)}$$

Using this approximation formula the number of documents N_i to be retrieved from each collection C_i can be determined. The query Q is consequently submitted to all collections C_i with a request to return back N_i documents. Of course, if N_i =0 collection C_i will not be involved in the distributed searching. The methods used to conduct the individual searches and to produce the results are not the concern of the fusion strategy. Designers of remote hypermedia collections can decide and apply different retrieval methods. This already guarantees a certain degree of autonomy and heterogeneity.

5.2.3 Third Phase: Merging Results

In this phase individual results returned back are merged to produce a single result. The method to produce a single result is to cluster documents and display a ranked list of clustered documents. This method in mainly inspired by the work undertaken for several years now in a Scatter/Cluster information seeking method (Cutting et al, 1992). Lately, this cluster-based approach was applied for presenting retrieval results (Hearst & Pedersen, 1996). Systematic evaluation of clustering retrieval results provides significant advantages over a ranked list of documents.

In the clustering method which is used in the third phase of the link-based fusion method, however, a different set of values is used to produce the clusters. To be precise, documents are clustered not by only using content-based similarity, but links between the documents returned from individual collections are additionally taken into account for producing the clusters in the final merged list. In this respect this approach is similar to the approach taken by Weiss et al (1996) in developing HyPursuit, a hierarchical search engine which is reviewed in Chapter 2. What is new in the link-based fusion strategy is that it applies the multiple evidence clustering

method not to organise documents and increase efficiency of search engines, but to effectively merge individual search results and present them to information seekers.

5.3 Evaluation of the Collection Fusion Strategy

A number of user-centered and system-centered experiments have been conducted to evaluate the effectiveness and the efficiency of the link-based collection fusion method. The user-centered evaluations are reported and discussed in Chapter 8. This section reports only the system-centered evaluations using a distributed IR system and two standard IR test collections.

5.3.1 Aims

The aim of the experiment is to evaluate the effectiveness and the efficiency of the link-based collection fusion strategy, and to compare it with other collection fusion strategies reviewed in Chapter 4. The uniform fusion strategy is used as the baseline against which our link-based fusion method is evaluated. It is generally accepted that in order to prove its usefulness, a fusion strategy must be compared with the uniform strategy (Voorhees et al, 1995).

The random fusion strategy is also used in the comparative evaluation for two reasons. First, because it is the method which is widely used, by many information seekers in real electronic environments since it is usually the only available one. Second and most important, if the link hypothesis is not valid, our fusion strategy will perform similarly to the random method.

In order to have a more complete view, the link-based fusion method is also compared with the optimal fusion strategy. It will be recalled that the optimal method is a retrospective technique which can not really applied in real environments, but it gives an upper bound for the effectiveness and efficiency of any collection fusion strategy. Finally, the results of the fusion strategies are compared with the results produced if treating the distributed collection as a single collection.

The effectiveness of all the fusion strategies is calculated using the uninterpolated precision (P) and recall (R) of the final merged results. These two indices have been traditionally used for the evaluation of information retrieval algorithms.

The indices mentioned above measure the effectiveness of a fusion strategy. However, in Chapters 1 and 4 the importance of efficiency in large distributed information seeking environments has been stressed. Therefore, another measure is introduced to evaluate and

compare the efficiency of our fusion strategy with the others: the total number of libraries finally involved in the distributed searching. The goal of fusion strategies should be to minimise this number, because smaller number of libraries involved means more efficient searching.

5.3.2 Experimental Environment

The experimental environment used for the evaluation of the fusion strategies comprises a simple vector-based distributed information retrieval system and the CACM and CISI test collections. The author developed the DIR system based on a set of well known freely available algorithms for vector-based information retrieval (Frakes and Yates, 1992). The experimental DIR system basically provided the capability to use multiple collections for running a set of controlled parallel distributed searches (Figure 5.3), and save their results.

A test collection is a document collection which comes together with a set of queries and associated relevance assessments (Robertson, 1981). CACM and CISI are two typical medium size test collections used in the past to conduct hundreds of IR experiments. Documents in the CACM collection contain 7 type of concepts (title, author, abstract, keywords, cocitations, citations and bibliographic couplings) for the 3204 computer science articles published in the journal communications of the ACM from 1958 through 1979. It comes with 64 natural language queries but only 51 of them have relevance assessments. The CISI documents contain 4 types of concepts (i.e. title, author, abstract and citations) for the 1460 information science articles published in the 1969-1977 period. This collection comes with 76 queries but only 54 have relevance assessments.

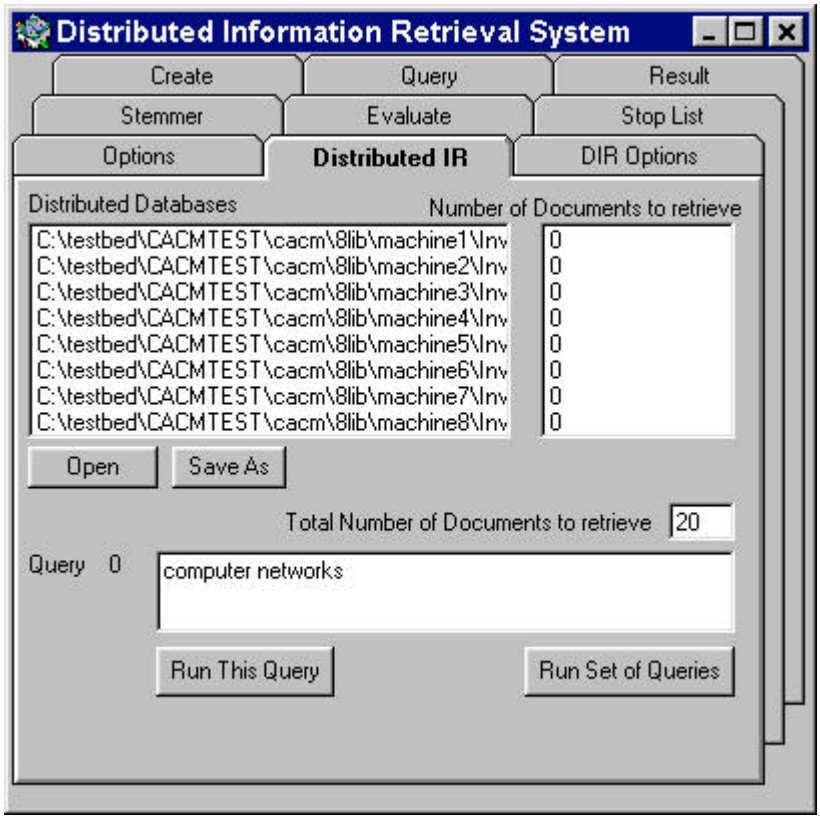


Figure 5.3: A snapshot of the DIR system which was used to conduct the experiments described in this chapter.

For both test collections the bibliographic information (i.e. citations, cocitations and bibliographic couplings) comes as a linkbase which contains all the links between the documents in the collection. These links between the documents are necessary in order to be able to utilise our link-based collection fusion strategy. Also, the variety of link types offered by the CACM collection, permitted to evaluate the effect of using different link types in the composition of linkage information. Some of the documents in the CACM collection do not have links with other documents and therefore could not be used in the experiments. In the results reported in the rest of this chapter a subset of the CACM collection has been used ($CACM_B$) which contains all the documents which have links to other documents in the collection. Some statistical information about the test collections is given in Table 5.1.

	No. of Docs.	No. of Terms	Av. No. of Terms per Document	No. of Queries	No. of citations	No. of cocitations	No. of Bibliog. couplings
CACM	3.204	10.446	40.1	51	6.786	12.456	5.789
CISI	1460	7.392	104.9	54	16.456	N/A	N/A
$CACM_B$	1751	5.479	47.2	51	6.786	12.456	5.789

Table 5.1: Basic characteristics of the CACM and CISI test collections.

The collections were both single collections of documents. In fact, this served our need to compare the effectiveness of the link-based fusion strategy with the effectiveness of retrieval runs using a single collection. The main goal was however to compare different fusion methods. For this reason, distributed versions of both collections had to be developed in order to evaluate different fusion strategies in an environment approximating a distributed hypermedia digital library.

A complete-link hierarchical clustering method has been used to cluster the test collections (Voorhees, 1986). The content similarity between documents used as the measure to create the similarity matrix. Using the clusters produced CACM and CISI collections divided into subcollections, each playing the role of a distributed autonomous collection.

Based on the experiences of other collection fusion experiments in the past, three sets of distributed collections have been developed for each test collection. One comprising 8 distributed sub-collections, one having 18 and one comprising 36. So, in total six different distributed hypermedia collections have been developed to test the fusion strategies. This decision was made in order to evaluate the fusion methods in a wide range of hypermedia digital libraries in terms of participating sub-libraries. Previous experiments evaluating collection fusion strategies have been conducted in a relatively small number of collections (10 maximum). The evaluations reported here are the first made in an environment with is characterised by larger distribution (i.e. a distributed environment having 18 and 36 distributed collections).

The actual searches have been conducted using 51 queries from the CACM collections (i.e. all the queries available with relevance assessments), and, for practical reasons 51 queries out of the 54 available from the CISI collection. The results which are reported in this chapter are based on inverted indices having normalised term word frequencies. Finally, the inner product between normalised terms and unweighted query terms was used as the similarity function (Salton & Buckley, 1988).

The effectiveness of the distributed IR system/engine in absolutely terms was not important, because the aim of the experiment is to evaluate the relative effectiveness between different fusion strategies using the same search engine. However, in absolutely terms the results of our single run is comparable with other reported single runs using the same test collections (e.g. Savoy 1996). So, the common thread for all the fusion strategies is that they have all used the same hypermedia digital libraries, the same retrieval engine, the same set of queries, same stemmer and stop list. The only parameter which is variable is the collection fusion method and any difference in the results must be attributed only to the differences in the fusion strategies.

5.3.4 Methods

In summary, for each of the 6 hypermedia digital libraries (i.e. distributed versions of CISI with 8, 18, 36, and versions of CACM with 8, 18, 36 sub-libraries), 4 collection fusion strategies have been tested (i.e. uniform, optimal, random and link-based fusion). The approximation function found in section 5.2.2 was used by the link-based fusion strategy. For each fusion strategy five different retrieval runs were made, each requesting a different total number of documents (i.e. 5 different runs were made each using 5, 10, 30, 50, 100 as the cut-off value). For each run the R, P and the number of libraries involved in the distributed search have been calculated.

Statistical analysis of the results obtained was performed using the SPSS programme. Both parametric and non-parametric statistical methods were applied. Analysis of variance (2 way Anova) were used to identify possible interaction effects between fusion strategy and number of distributed libraries. Fusion method or number of distributed libraries categories data were analysed using one-way analysis of variance (complete randomised design), in cases of normality of frequency distributions with or without transformed data. This type of statistical analysis is generally accepted as the most appropriate in IR if the sample (i.e. queries) is larger than 30 and follows the normal distribution (Kraaij & Pohlmann, 1996). Differences

between mean values were tested for significance using Duncan's new multiple range test. The homogeneity of variances was evaluated using the Bartlett-Box F (Kanji 1994). Kruskal-Wallis's non-parametric test was also applied to seek differences between fusion method categories in case of non normality. All statistical tests were performed under a significance level of 5%.

5.3.4 Discussion of Retrieval Results

General comments

The experimental results are very positive and to a large extent verify the hopes about the link-based fusion strategy. First, the link-based collection fusion method has achieved statistically significant higher recall and precision than the random approach in all the conditions tested. This is a very positive result validating the link hypothesis. Additionally, the link-based collection fusion strategy has steadily produced better results than the uniform approach. The differences observed are statistically significant in most of the cases.

Further expectations regarding the effectiveness of the fusion strategies have been confirmed from the experiment. First, the optimal fusion strategy performed better than any other collection fusion. Moreover, this method has performed better than searching a single collection. This result is in line with other collection fusion experiments conducted in the past (e.g. Voorhees, 1997). It also verifies that the goal of realising distributed information systems without sacrificing effectiveness is feasible. The rest of the collection fusion strategies tested have performed less effectively than the single run. However, in several cases the degradation of effectiveness was small and not statistically significant.

Analysis of the results reveals even more positive results for the link-based fusion strategy in comparison to the efficiency of the other fusion methods. The link-based collection fusion strategy has performed better than the uniform approach in all conditions. Even in the few cases where the difference in effectiveness between the link-based and the uniform method was small, the corresponding difference between the method regarding the number of libraries involved in distributed searching was large and statistically significant. This is very positive result considering the emphasis that should be given to efficiency when searching in large electronic environments such as hypermedia digital libraries.

The optimal fusion method has also produced the best results in terms of efficiency. This is a stimulating result indicating that it is possible to achieve simultaneously both high

effectiveness and efficiency. The following sections discuss individual results in more detail and outline some points which need further consideration.

Effectiveness results

Full range of cut-off levels

Figures 5.4, 5.5 and Table 5.2 illustrate[12] the averaged recall and precision results using three different distributed versions of the CACM collection and for the full range of cut-off levels (i.e. three different hypermedia digital libraries each comprising 8, 18 and 36 sub-libraries). The table clearly demonstrates that the link-based fusion method produced better recall and precision values than the random in all cases, and than the uniform fusion strategies in most cases. The differences between the random and the link-based methods were statistically significant in all the cases. The differences in precision between the link-based and the uniform approach were statistically significant in the digital library having 36 libraries (main effects between the fusion methods within the same distributed library are indicated in Table 5.2 by the lower case superscripts).

Another observation which is remarkable is that the difference in recall and precision between the uniform, the random and the link-based approach increases as the number of libraries increase. Similarly, the difference between the optimal and the single approach increases in the same way. In other words, a higher distributed digital libraries amplifies further the differences between different fusion methods.

This effect can be explained by the possibility that as the number of sub-libraries increases (e.g. from 8 to 18 and to 36) the cluster and the link hypotheses have a greater influence to the collection fusion strategies. Based on this hypothesis a two-way Anova analysis has been conducted to identify possible interactions between independent variables. Indeed, this analysis reveals interactions between the number of libraries and the collection fusion methods (the interaction effects that the number of sub-libraries has to a fusion method are indicated in

[12] the decision to use both figures and tables to illustrate the same information was driven by the need for readability (found in figures) and the need to present statistical results (found in tables) which can not be shown in the figures. In discussion of the results the tables are mainly used since they have all the information that figures present plus statistical results. However, figures may be used as an auxiliary source to view, examine and compare mean results easier.

Table 5.2 by the capital superscripts). So, that means that the number of libraries has an interaction effect to the performance of fusion strategies. For example, the difference in the performance of the link-based and the uniform strategies is not constant, but it increases as the number of libraries increase.

Although, the results of the link-based fusion method in comparison to the results of the single run show a decrease in effectiveness, the effectiveness achieved is still satisfactory especially for interactive environments (as it will be shown in Chapter 8).

The performance of our link-based fusion strategy in comparison to other integrated collection fusion strategies presented in Chapter 4 is also quite encouraging. Generally, the link-based method presents a degradation of effectiveness similar to the degradation that these methods present in their evaluations.

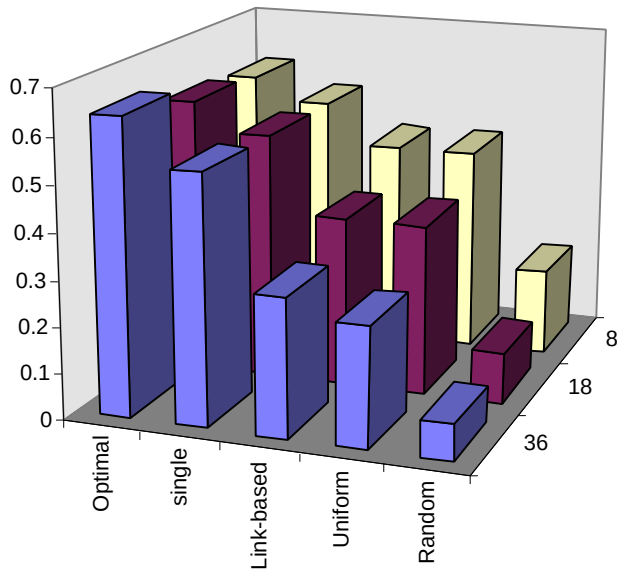


Figure 5.4: Averaged Recall results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

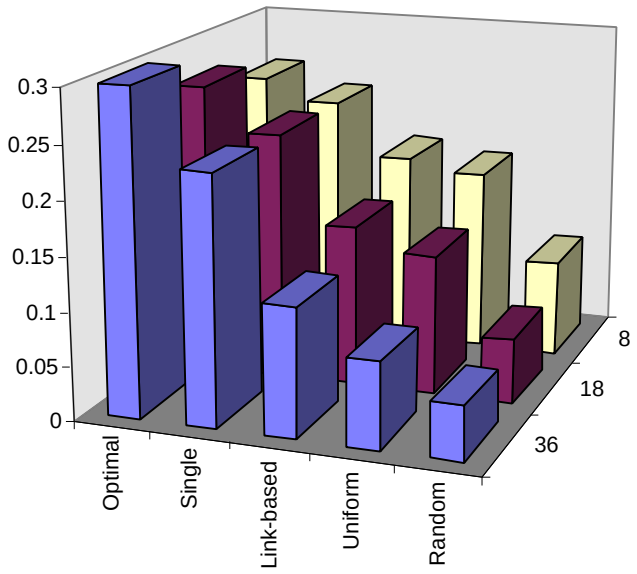


Figure 5.5: Averaged Precision results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

HDL	Methods	Recall	Precision
CACM-8	*uniform*	0.45 ? 0.0343 [a, A]	0.17 ? 0.022 [a, A]
		- 16%	- 25%
	optimal	0.59 ? 0.0374 [b, A]	0.25 ? 0.0246 [b, A]
		+ 9%	+ 9%
	random	0.19 ? 0.0195 [c, A]	0.09 ? 0.0113 [c, A]
		- 66%	- 61%
	link-based	0.45 ? 0.0322 [a, A]	0.18 ? 0.0226 [a, b, A]
		- 16%	- 23%
	single	0.54 ? 0.0375 [a, b, A]	0.23 ? 0.0226 [b, A]
CACM-18	*uniform*	0.37 ? 0.0326 [a, A]	0.13 ? 0.0183 [a, A]
		- 31%	- 45%
	optimal	0.60 ? 0.0366 [b, A]	0.27 ? 0.0242 [b, A]
		+ 12%	+ 17%
	random	0.11 ? 0.0135 [c, B]	0.06 ? 0.0076 [c, B]
		- 80%	- 75%
	link-based	0.37 ? 0.0269 [a, A, B]	0.15 ? 0.0190 [a, A, B]
		- 31%	- 36%
	single	0.54 ? 0.0375 [b, A]	0.23 ? 0.0226 [b, A]
CACM-36	*uniform*	0.26 ? 0.0242 [a, B]	0.08 ? 0.0133 [a, B]
		- 51%	- 65%
	optimal	0.64 ? 0.0357 [c, A]	0.30 ? 0.0248 [b, A]
		+ 18%	+ 30%
	random	0.08 ? 0.0109 [b, B]	0.05 ? 0.0076 [c, B]
		- 86%	- 80%
	link-based	0.30 ? 0.0222 [a, B]	0.12 ? 0.0147 [d, B]
		- 45%	- 50%
	single	0.54 ? 0.0375 [d, A]	0.23 ? 0.0226 [e, A]

Values are expressed as mean values ? standard error of the mean (X ? SEM). Percentage differences are in respect to single run (the minus symbol indicates degradation)

a, b, c, d, e: mean scores in the same column and for the same digital library without a superscript in common are significantly different

A, B: mean scores in the same column and for the same collection fusion method without a superscript in common are significantly different

Table 5.2: Averaged Recall and Precision results using three CACM hypermedia digital libraries (each having 8, 18, 36 sub-libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

Small range of cut-off levels

The retrieval results in Table 5.2 illustrate precision and recall values averaged over five different runs each requesting 5, 10, 30, 50 and 100 from the collections eventually involved in the distributed searching. So, this table gives the average performance of the fusion methods in a broad range of cut-off levels. But, usually information seekers in interactive environments specify smaller cut-off numbers (e.g. 10). It was therefore important to study the effectiveness of the fusion strategies using smaller cut-off numbers. Figures 5.6, 5.7 and Table 5.3 illustrate the averaged recall and precision results for the retrieval runs using two cut-off levels (5 and 10).

The results presented in Table 5.3 are even more positive than these illustrated in Table 5.2. Once more, the link-based fusion strategy produces better, statistically significant, recall and precision results than the random strategy in every condition. Likewise, the performance of the link-based fusion strategy was superior of the performance of the uniform approach. Also, these differences are statistically significant in more cases than these depicted in Table 5.2. For example, from Table 5.2 the average precision of the uniform and the link-based strategies in the CACM-18 HDL are 13% and 15% respectively. This is not a statistically significant difference. The corresponding retrieval results using the smaller cut-offs is 15% for the uniform method and 20% for the link-based fusion method, and are statistically significant. In other words, the relative difference between the performance of the methods has been increased, although both methods have increased their performance in absolutely terms.

Further observations and comments made based on the results presented in Table 5.2, are generally reinforced and confirmed from Table 5.3. The difference in the performance between the uniform, the random and the link-based fusion strategies is increased as the number of libraries increases. Indeed, a two-way Anova analysis revealed similar interactions as those identified in Table 5.2.

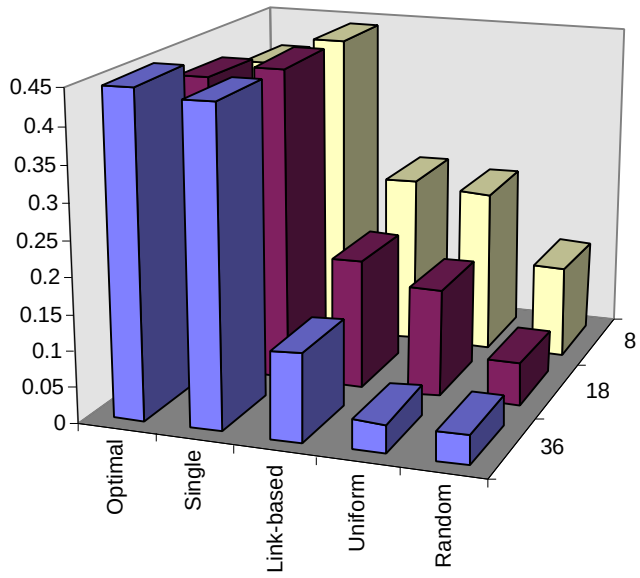


Figure 5.6: Averaged Recall results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the small set of cut off levels (5,10 documents).

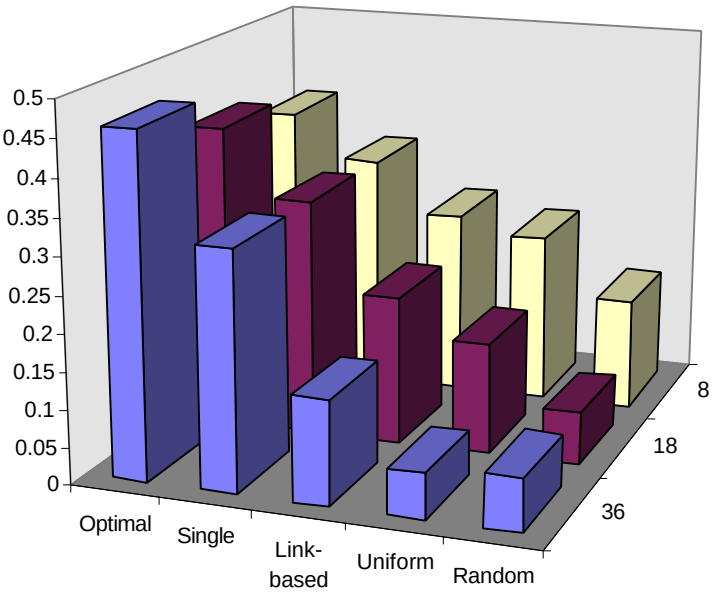


Figure 5.7: Averaged Precision results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the small set of cut off levels (5,10 documents).

HDL	Methods	Recall	Precision
CACM-8	*uniform*	0.23 ? 0.0373 [a, A]	0.23 ? 0.0274 [a, A]
		- 49%	- 25%
	optimal	0.40 ? 0.0458 [b, A]	0.38 ? 0.0330 [b, A]
		- 11%	+ 22%
	random	0.13 ? 0.0244 [c, A]	0.15 ? 0.0210 [c, A]
		- 70%	- 54%
	link-based	0.24 ? 0.0347 [a, A]	0.25 ? 0.0278 [a, d, A]
		- 46%	- 20%
	single	0.44 ? 0.0387 [b]	0.32 ? 0.0309 [b, d]
CACM-18	*uniform*	0.15 ? 0.0311 [a, B]	0.15 ? 0.0214 [a, B]
		- 66%	- 53%
	optimal	0.42 ? 0.0453 [b, A]	0.41 ? 0.0322 [b, A]
		- 5%	+ 31%
	random	0.06 ? 0.0145 [c, B]	0.07 ? 0.0122 [c, B]
		- 87%	- 76%
	link-based	0.18 ? 0.0258 [d, A, B]	0.20 ? 0.0217 [d, A, B]
		- 60%	- 35%
	single	0.44 ? 0.0387 [b]	0.32 ? 0.0309 [e]
CACM-36	*uniform*	0.04 ? 0.0123 [a, C]	0.06 ? 0.0126 [a, C]
		- 90%	- 79%
	optimal	0.45 ? 0.0450 [b, A]	0.46 ? 0.0333 [b, A]
		+ 3%	+ 47%
	random	0.04 ? 0.0080 [a, B]	0.07 ? 0.0148 [a, c, B]
		- 92%	- 79%
	link-based	0.12 ? 0.0186 [c, B]	0.14 ? 0.0155 [d, B]
		- 73%	- 55%
	single	0.44 ? 0.0387 [b]	0.32 ? 0.0399 [e]

Values are expressed as mean values ? standard error of the mean (X ? SEM). Percentage differences are in respect to single run (the minus symbol indicates degradation)

a, b, c, d, e: mean scores in the same column and for the same digital library without a superscript in common are significantly different

A, B: mean scores in the same column and for the same collection fusion method without a superscript in common are significantly different

Table 5.3: Averaged Recall and Precision results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the small set of cut off levels (5,10 documents).

Effectiveness results using the CISI collection

Undoubtedly, the retrieval results depicted in Tables 5.2 and 5.3 are very positive and illustrate the superiority of our link-based fusion strategy in comparison with the uniform and random approaches using three hypermedia digital libraries based on the CACM collection. The same experiments, this time using three CISI-based hypermedia digital libraries, are presented in Figures 5.8, 5.9 and Table 5.4. This table presents the performance of the fusion strategies using the full range of cut-off levels. In that sense, this table is analogous to the Table 5.2 for the CACM digital libraries.

In absolute terms, comparing the values for recall and precision between the CACM and the CISI, all the runs using the CISI digital libraries obtain better precision results and worse results for recall. This is explained by the different nature of the two collections. For example, CACM has less relevant documents per query, so therefore it is a collection favouring increased values for recall.

The results illustrated in Table 5.4 generally verify the discussion made before based on the results produced using the CACM collections. Again, the link-based fusion method consistently produced better recall and precision values than the random and the uniform fusion strategies using the CISI collections. The differences observed between the random and the link-based were statistically different in all conditions (i.e. using 8, 18 and 36 libraries).

Similarly to results obtained using the CACM collections, the optimal fusion strategy in CISI digital libraries outperformed all the other collection fusion methods. It additionally consistently outperformed in terms of precision and recall the single collection retrieval runs. The rest of the collection fusion methods have performed worse than the single run.

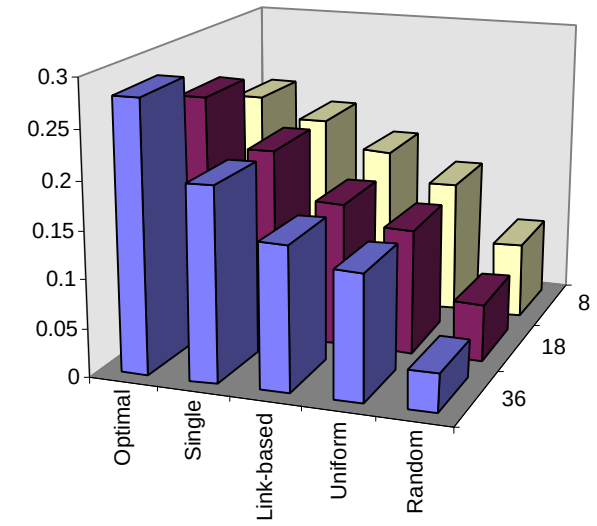


Figure 5.8: Averaged Recall results using three CISI hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

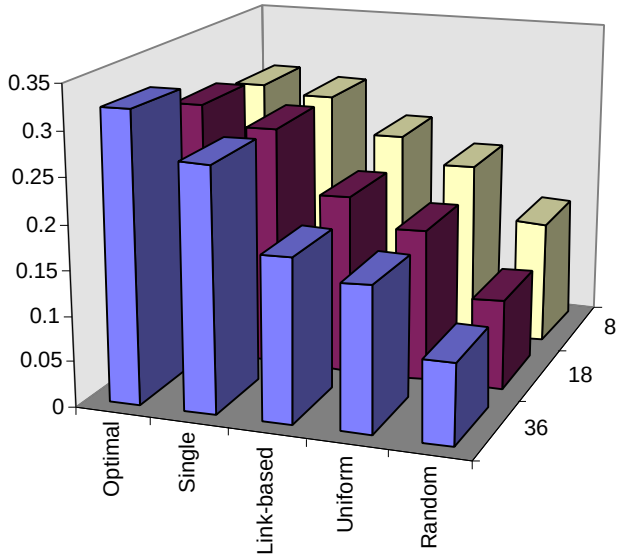


Figure 5.9: Averaged Precision results using three CISI hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

HDL	Methods	Recall	Precision
CISI-8	*uniform*	0.14 ? 0.0013 [a, A]	0.20 ? 0.0226 [a, d, A]
		- 47%	- 25%
	optimal	0.22 ? 0.0190 [b, A,]	0.28 ? 0.0256 [b, A]
		- 20%	+ 1%
	random	0.08 ? 0.0068 [c, A]	0.14 ? 0.0165 [a, A]
		- 71%	- 49%
	link-based	0.17 ? 0.0143 [a, d, A]	0.23 ? 0.0236 [b, d, A]
		- 38%	- 16%
	single	0.20 ? 0.0176 [b, d]	0.27 ? 0.0279 [c, b]
CISI-18	*uniform*	0.13 ? 0.0137 [a, A]	0.17 ? 0.0205 [a, A]
		- 52%	- 36%
	optimal	0.25 ? 0.0230 [b, A, B]	0.29 ? 0.0258 [b, A]
		- 10%	+ 7%
	random	0.06 ? 0.0079 [c, B]	0.10 ? 0.0137 [c, A, B]
		- 79%	- 62%
	link-based	0.15 ? 0.0186 [a, A]	0.20 ? 0.0205 [a, A]
		- 44%	- 27%
	single	0.20 ? 0.0176 [b]	0.27 ? 0.0279 [b]
CISI-36	*uniform*	0.13 ? 0.0157 [a, A]	0.16 ? 0.0188 [a, A]
		- 52%	- 42%
	optimal	0.28 ? 0.0235 [b, B]	0.32 ? 0.0241 [b, A]
		+ 3%	+ 18%
	random	0.04 ? 0.0043 [c, B]	0.09 ? 0.0109 [c, B]
		- 85%	- 68%
	link-based	0.15 ? 0.0184 [a, A]	0.18 ? 0.0187 [a, A]
		- 48%	- 37%
	single	0.20 ? 0.0176 [d]	0.27 ? 0.0279 [b]

Values are expressed as mean values ? standard error of the mean (X ? SEM). Percentage differences are in respect to single run (the minus symbol indicates degradation)

a, b, c, d, e: mean scores in the same column and for the same digital library without a superscript in common are significantly different

A, B: mean scores in the same column and for the same collection fusion method without a superscript in common are significantly different

Table 5.4. Averaged Recall and Precision results using three CISI hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

The increase of the differences between different fusion strategies as we move to higher distributed digital libraries is also generally confirmed. However, the effect of higher distribution seems to be smaller than in the CACM collections. For example, in the retrieval results using the CACM collection the difference in precision values was increasing between the link-based and uniform strategies as the number of libraries were increasing (i.e. higher distribution). The results from the CISI collection show no increase in the difference between the two fusion methods moving from 8 to 18 libraries. For example P is 0.20 and 0.23 for the uniform and link-based methods respectively in CISI-8 and, it goes to 0.17 and 0.20 respectively in CISI-18.

This can be explained by the different characteristics of the hypermedia collections. The CISI collection has more relevant documents per query and these relevant documents are more uniformly distributed between the collections. Figure 5.10 illustrates the distribution of relevant documents to participating collections for the CACM and CISI digital libraries. Relevant document in CISI digital libraries are distributed to more collections. This has a positive effect to the uniform and the random strategies, because the uniform/random way of approximating distribution of relevant documents conforms better to digital libraries having relevant documents in many collections.

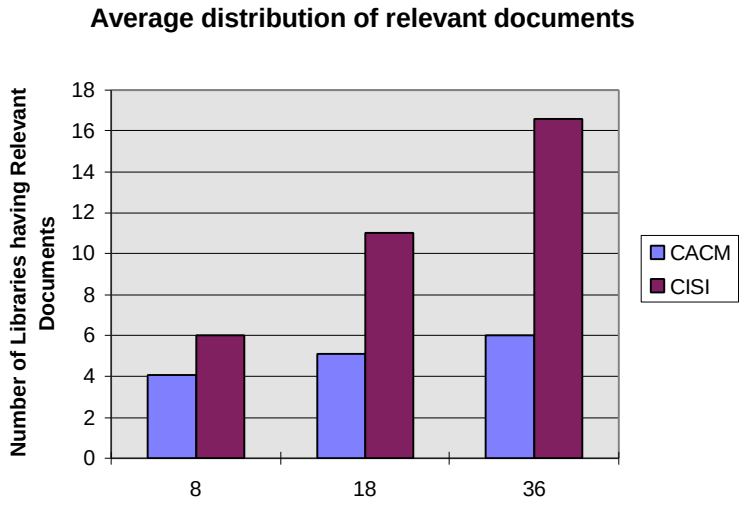


Figure 5.10: Average distribution of relevant documents per query to different sub-collections for the CACM and CISI libraries.

Efficiency results

Full range of cut-off levels

Figure 5.11 and Table 5.6 illustrate the results obtained using the CACM digital libraries regarding the number of libraries involved in the distributed searching. The assumption which is made to connect this number with efficiency is that the time for contacting remote collections and to exchange data is substantially larger than the time spent in actually retrieving documents in remote collections. Therefore, practically this time may represent the overall time for a distributed search. The smaller this number is (i.e. the less libraries involved) the more efficient is the distributed searching process.

The results in Table 5.6 clearly illustrate that the link-based fusion strategy has consistently produced better, statistically significant, results than the uniform approach. In other words, in distributed searches using the link-based approach less libraries have been finally used. For example, in the CACM hypermedia digital library with 8 collections the link-based fusion strategy uses on average 72% of the available collections in contrast to the uniform method which uses 93% of the available libraries. The respective numbers for the CACM library having 18 collections are 49% and 77%. Finally, for 36 collections the link-based method involves 35% of the available collections in the distributed searching while the uniform uses 65% of them.

These results prove that our link-based method is not only more effective, but is also significantly more efficient than the uniform strategy. For example, for the CACM digital library having 36 distributed collections, the uniform approach involves 23 libraries in the distributed searching process to finally achieve 26% and 8% recall and precision respectively (R and P values are taken from Table 5.2). On the other hand, the link-based fusion method uses 12.4 libraries to achieve 30% and 12% recall and precision respectively. In other words, not only the link-based fusion method involves less collections than the uniform approach in the distributed searching, but it additionally produces better R and P results.

The results in Table 5.6 also confirm the expectations about the optimal fusion strategy. The optimal fusion method produced the best results from the rest of the collection fusion strategies. The random approach also produces good results in terms of efficiency. This should be explained by the random way of requesting documents from remote libraries which will usually lead in making requests only to a small number of libraries. However, as it is illustrated in Table 5.2 (i.e. the table having the corresponding effectiveness results) and also

previously discussed, the random strategy produces poor effectiveness making the method practically inapplicable.

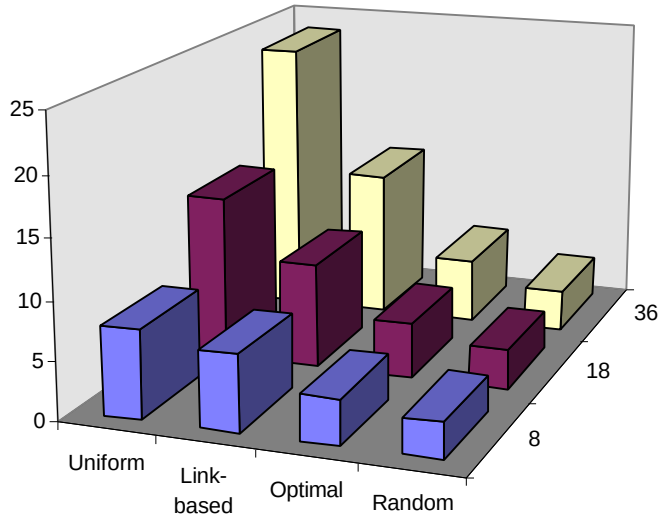


Figure 5.11: Averaged "number of libraries involved" results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

HDL	Methods	Libraries involved
CACM-8	*uniform*	7.400 ? 0.0000 [a, A] 93%
	optimal	3.8275 ? 0.2332 [b, A] 48%
	random	3.0275 ? 0.0582 [c, A] 38%
	link-based	6.3827 ? 0.0283 [d, A] 72%
CACM-18	*uniform*	13.8000 ? 0.0000 [a, B] 77%
	optimal	4.6824 ? 0.3079 [b, A, B] 26%
	random	3.3333 ? 0.0809 [c, B] 19%
	link-based	8.8043 ? 0.0966 [d, B] 49%
CACM-36	*uniform*	23.4000 ? 0.0000 [a, C] 65%
	optimal	5.4902 ? 0.3944 [b, B] 14%
	random	3.5804 ? 0.0789 [c, C] 10%
	link-based	12.4776 ? 0.2194 [d, C] 35%

Values are expressed as mean values ? standard error of the mean (X ? SEM). Percentages are in respect to the maximum number of collections available for searching

a, b, c, d, e: mean scores in the same column and for the same digital library without a superscript in common are significantly different

A, B: mean scores in the same column and for the same collection fusion method without a superscript in common are significantly different

Table 5.6: Averaged number of libraries involved for the full range of cut off levels (5,10,30,50,100) using the CACM collection and for the 8,18,36 distributed hypermedia libraries.

Small range of cut-off levels

Similar efficiency results are obtained when only the CACM retrieval runs with the smaller cut-off levels are considered and are presented in Figure 5.12 and Table 5.7. This table presents the averaged numbers of libraries involved for the runs having as cut-off levels 5 and 10 documents.

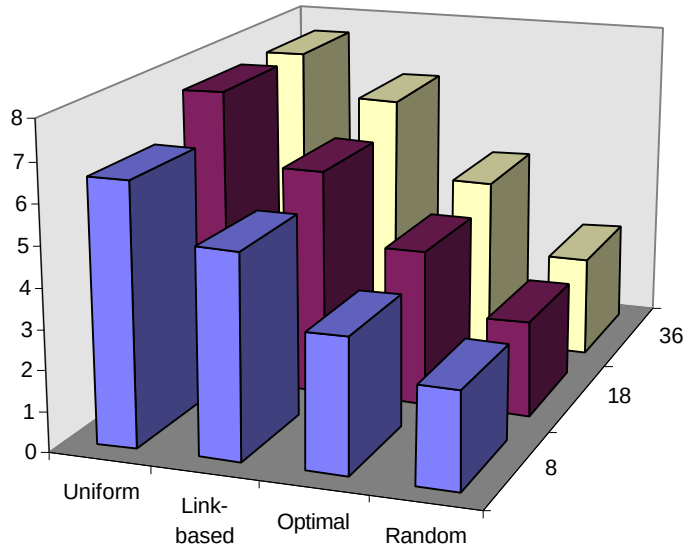


Figure 5.12: Averaged "number of libraries involved" results using three CACM hypermedia digital libraries (each having 8, 18, 36 libraries) and for the small range of cut off levels (5 and10 documents).

HDL	Methods	Libraries involved
CACM-8	*uniform*	6.500 ? 0.0000 [a, A] 81%
	optimal	3.3431 ? 0.1785 [b, A] 42%
	random	2.4020 ? 0.1039 [c, A] 30%
	link-based	5.0788 ? 0.2566 [d, A] 57%
CACM-18	*uniform*	7.5000 ? 0.0000 [a, B] 42%
	optimal	3.9020 ? 0.2110 [b, A, B] 22%
	random	2.3922 ? 0.0963 [c, C] 13%
	link-based	5.7012 ? 0.0385 [d, B] 31%
CACM-36	*uniform*	7.5000 ? 0.0000 [a, B] 21%
	optimal	4.3922 ? 0.2514 [b, B] 12%
	random	2.5392 ? 0.1153 [c, C] 7%
	link-based	6.4053 ? 0.0332 [d, C] 18%

Values are expressed as mean values ? standard error of the mean (X ? SEM). Percentages are in respect to the maximum number of collections available for searching

a, b, c, d, e: mean scores in the same column and for the same digital library without a superscript in common are significantly different

A, B: mean scores in the same column and for the same collection fusion method without a superscript in common are significantly different

Table 5.7: Averaged number of libraries involved for the smaller set cut-off levels (5,10) using the CACM collection and for the 8,18,36 distributed hypermedia libraries.

Efficiency results using the CISI collection

The efficiency results obtained using the CISI digital libraries are illustrated in Figure 5.13 and Table 5.8. Once again, the link-based fusion strategy performs better than the uniform strategy and the optimal method performs better than any other fusion strategy.

Additionally, from this table it is confirmed the effect that the distribution of relevant documents to participating collections may have to fusion strategies. In the section discussing the effectiveness results it is pointed out that distribution of relevant documents to more collections accounts for increased performance for random and uniform strategies. Now, similar observation can be made for the efficiency results of the CISI collections. Once again, despite the fact that the link-based method outperforms the uniform approach the differences are smaller than these obtained in the CACM collection where relevant documents to a query are less and tend to be located in fewer collections.

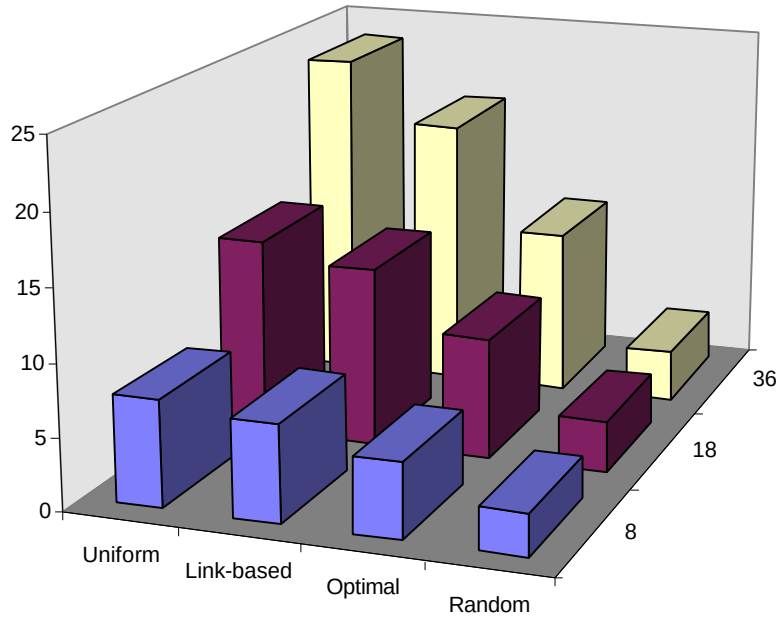


Figure 5.13: Averaged "number of libraries involved" results using three CISI hypermedia digital libraries (each having 8, 18, 36 libraries) and for the full range of cut off levels (5,10,30,50,100 documents).

HDL	Methods	Libraries involved
CISI-8	*uniform*	7.40 ? 0.000 [a, A]
	optimal	5.17 ? 0.1822 [b, A]
	random	2.91 ? 0.0620 [c, A]
	link-based	6.78 ? 0.0279 [d, A]
CISI-18	*uniform*	13.80 ? 0.000 [a, B]
	optimal	8.45 ? 0.3956 [b, B]
	random	3.52 ? 0.0825 [c, B]
	link-based	12.54 ? 0.0290 [d, B]
CISI-36	*uniform*	23.40 ? 0.000 [a, C]
	optimal	11.58 ? 0.6057 [b, C]
	random	3.60 ? 0.1081 [c, B]
	link-based	19.08 ? 0.0619 [d, C]

Values are expressed as mean values ? standard error of the mean (X ? SEM). Percentages are in respect to the maximum number of collections available for searching

a, b, c, d, e: mean scores in the same column and for the same digital library without a superscript in common are significantly different

A, B: mean scores in the same column and for the same collection fusion method without a superscript in common are significantly different

Table 5.8. Averaged number of libraries involved for cut-off levels (5,10, 30, 50, 100) using the CISI collection and for the 8,18,36 distributed hypermedia libraries.

5.4 Effects of Method Parameters on Performance

5.4.1 Selecting Sampling Collection

In section 5.2 the issue of selecting the sampling hypermedia collection was discussed. This section now presents and discusses the actual effects from selecting different sampling collections in our experiments. Figure 5.14 presents the recall and precision values of different retrieval runs each using a different sampling collection. These runs made using the CACM digital library comprising 8 collections. Thus, 8 different results for both recall and precision are illustrated. The ninth bar represents the average result.

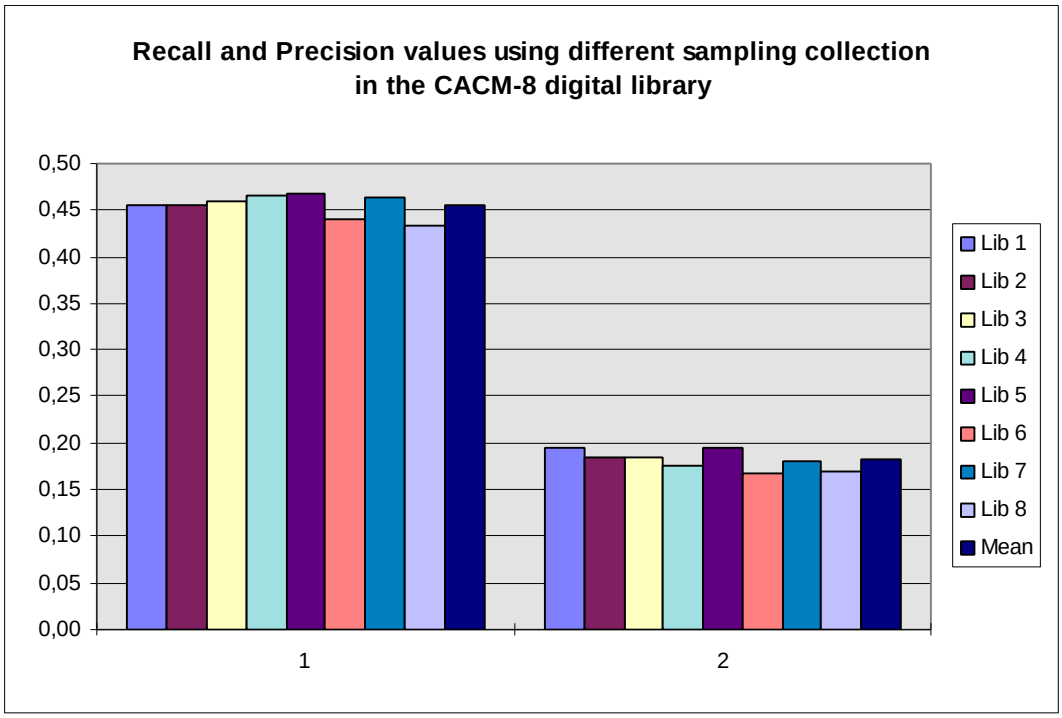


Figure 5.14: The effect of using a different sampling collection to the performance of the link-based fusion method in the CACM-8 digital library.

In this figure, it can be actually observed that there is an effect from selecting a different sampling collection to the performance of the link-based fusion strategy. For example, using the sixth and the eighth collections for sampling, produces worse results for both recall and precision than the average values. Inversely, using the fifth collection the fusion method produces better results than the average values. The fact that the same collections produce the best/worse results for both recall and precision induces that there must be a reason for this

particular behaviour. The results could be explained considering the allocation of relevant documents to collections (Figure 5.15). The sixth and eighth collection have the smaller number of relevant documents. Relevant documents contribute to successfully approximating the distribution of relevant documents to other collections. Thus, it is likely that extracting less relevant documents in the sampling process may result in less effective approximations.

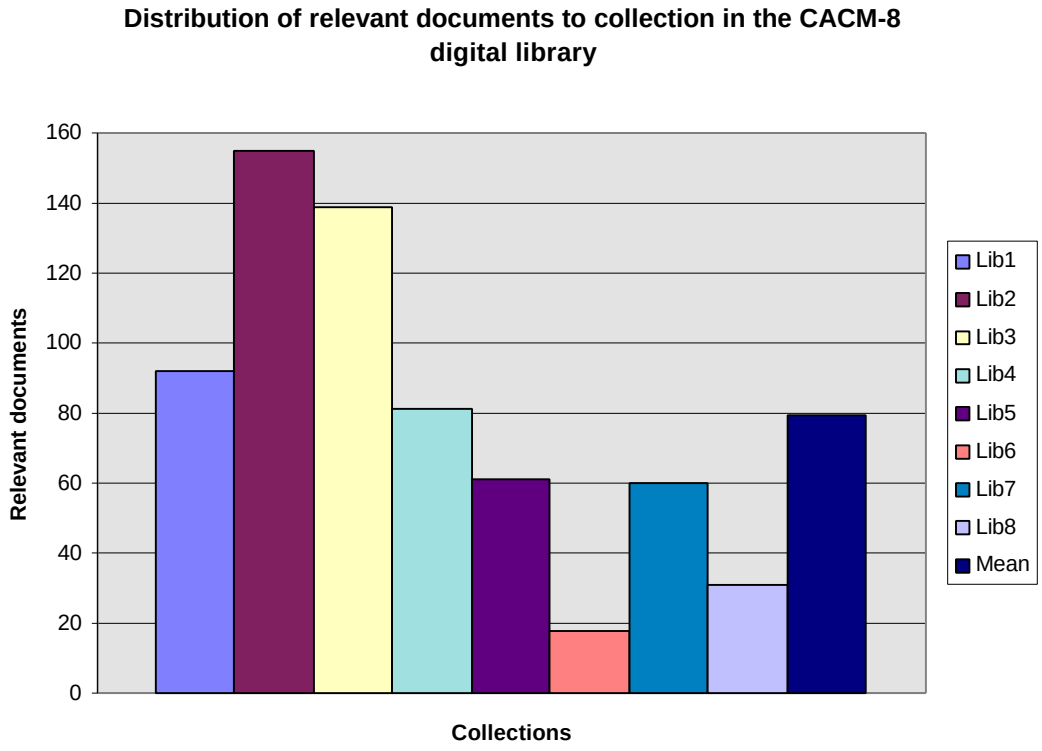


Figure 5.15 Distribution of relevant documents in the CACM-8 digital library.

The variances observed, however, are very small (the standard deviation is 0.0125 and 0.0103 for recall and precision respectively). The fact that changing sampling collections does not have a substantial effect to performance is encouraging, because it increases the stability and applicability of the link-based fusion method.

5.4.2 Number of Documents Retrieved from the Sampling Collection

The second parameter which may have an effect on the performance of the method is the number of relevant documents retrieved from the sampling collection in order to extract

linkage data. It will be recalled that a document is regarded as relevant if it is in the top X number of documents.

All the retrieval results that they presented in this chapter for the link-based fusion strategy represent averaged values for different runs, each using a different number of top documents from the sampling collection to extract linkage data. More precisely, seven different runs are averaged each using the 4, 8, 12, 16, 25, 45, 100 top documents from the sampling collection.

Figure 5.16 presents the corresponding retrieval results using the CACM digital library having 18 collections. In this figure it is shown that the number of top documents used has actually a small effect on the performance of the link-based fusion strategy.

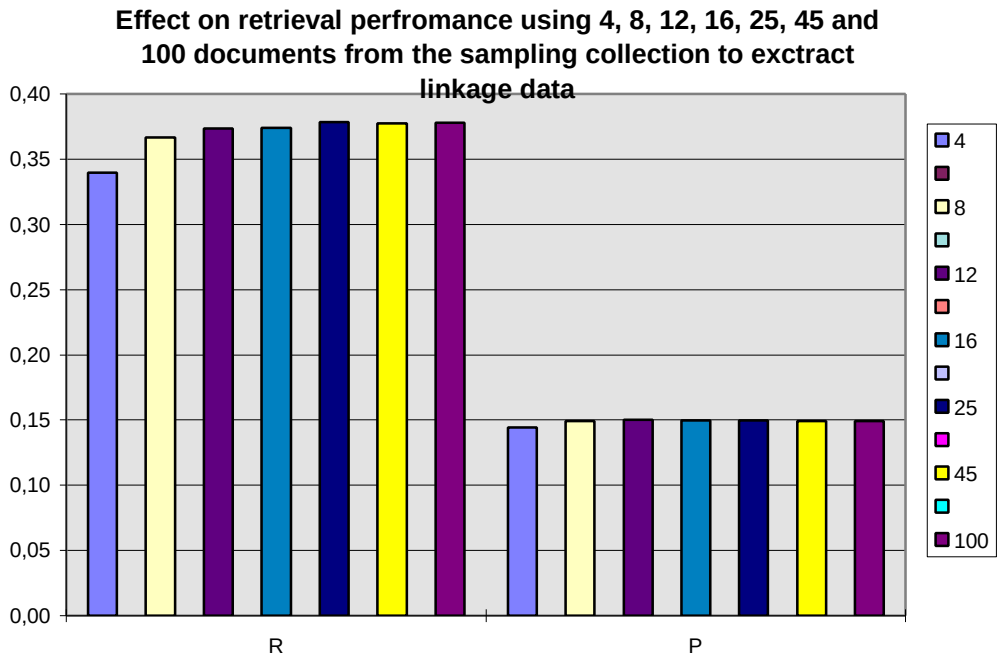


Figure 5.16: The effect of using different number of documents to extract linkage data.

It could be generally said that the number of documents used to produce linkage data does not have a major effect on the performance of the link-based fusion strategy. However, there are some benefits in considering this parameter. What seems to be the best strategy is to identify the threshold after which increasing the number of documents does not have an important effect on performance (e.g. 8 sampling in CACM-18), and subsequently use this threshold to retrieve documents from the sampling collection.

There are two reasons advocating this strategy. First, extracting fewer documents during the sampling phase is more efficient in terms of computation times. Second, this threshold seems to have an effect on efficiency. Figure 5.17 presents the efficiency results using the CACM digital library comprising 18 collections. This figure illustrates the effect of the number of documents to the number of collection finally involved in distributed searching. In contrast with what is observed for effectiveness, the efficiency is affected to a higher degree. For example, using 8 documents from the sampling collection results in involving 8.5 collections in the distributed searching. Using 100 documents results in involving 9.3 collections. This is a significant difference (about one collection more) considering that practically both achieve almost same effectiveness results.

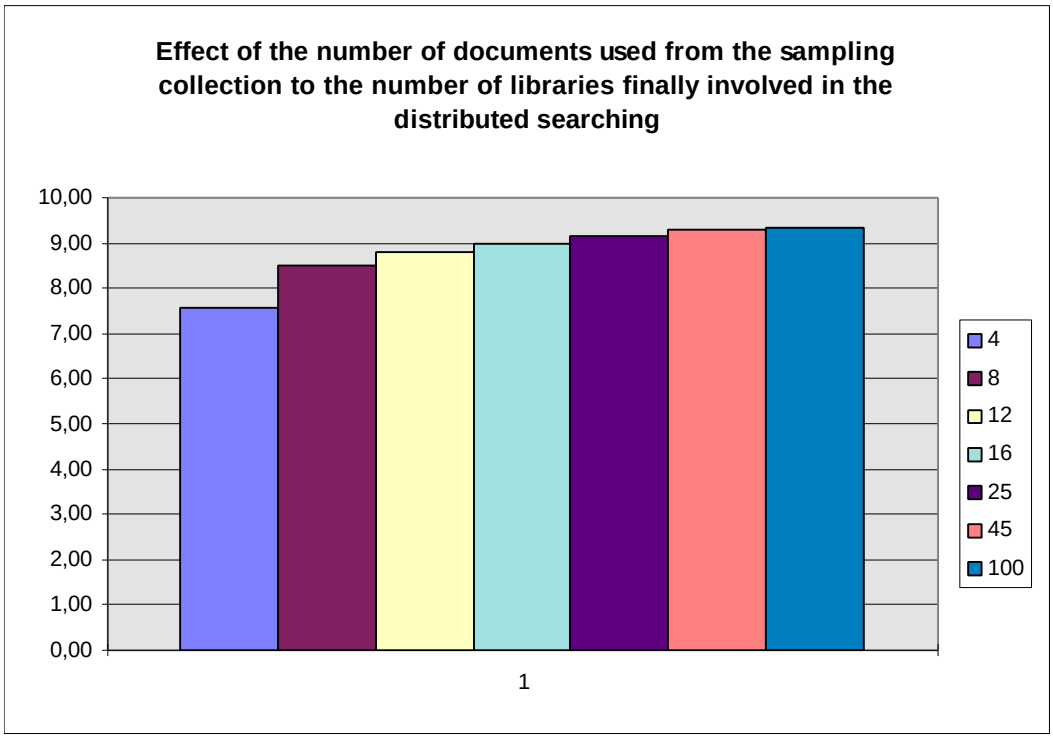


Figure 5.17: Number of collections involved in distributed searching for different number of documents retrieved form the sampling collection.

5.4.3 Using actual relevant documents in sampling stage

A variation of the link-based fusion strategy can be applied if it was possible during the sampling stage to distinguish actual relevant documents. In this variation, only these documents will be considered from the top X documents retrieved from the sampling collection. Figures 5.18 and 5.19 present the retrieval runs of this variation using the CACM-

8 and CISI-18 digital libraries, in comparison with the "normal" results which consider as relevant all the top X documents.

These two figures illustrate that the effect of including documents which are actually relevant is very small and in most of the cases, quite surprisingly, negative. From these results it can be inferred that the full (i.e. relevant and non-relevant) set of documents retrieved from the sampling collection is slightly more informative from the set containing only strictly relevant documents. It must be said, however, that these results might be a consequence of a special characteristic of the test collections, or of their relevance judgements.

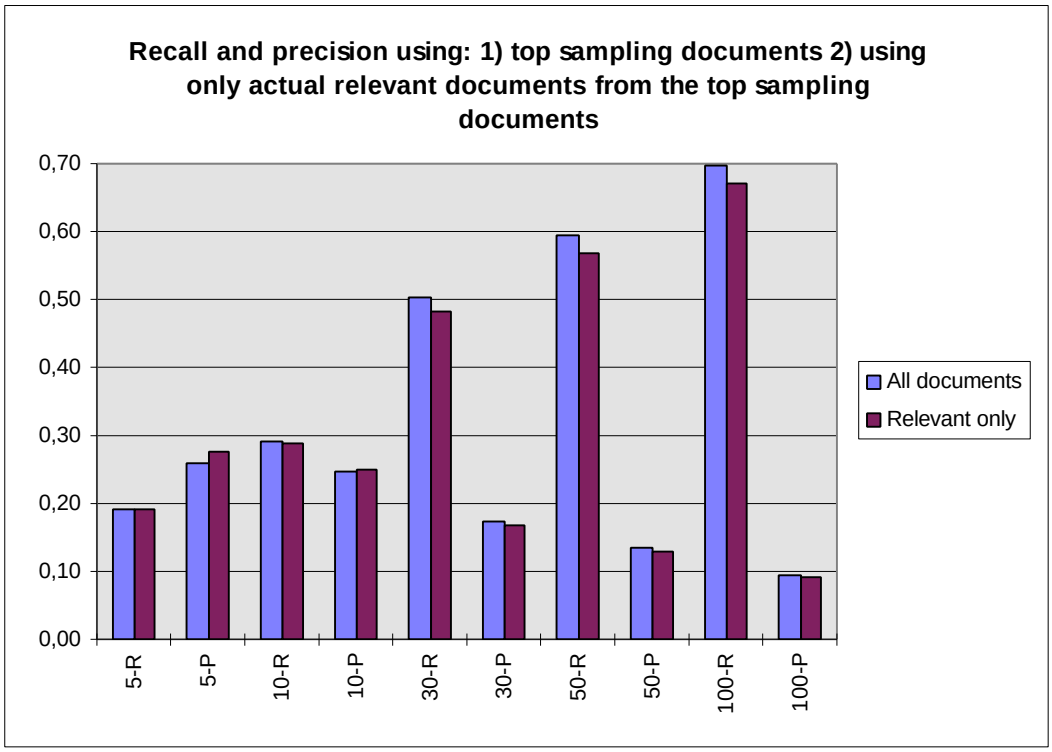


Figure 5.18: The effect of using only actual relevant documents from the sampling collection in the CACM-8 digital library.

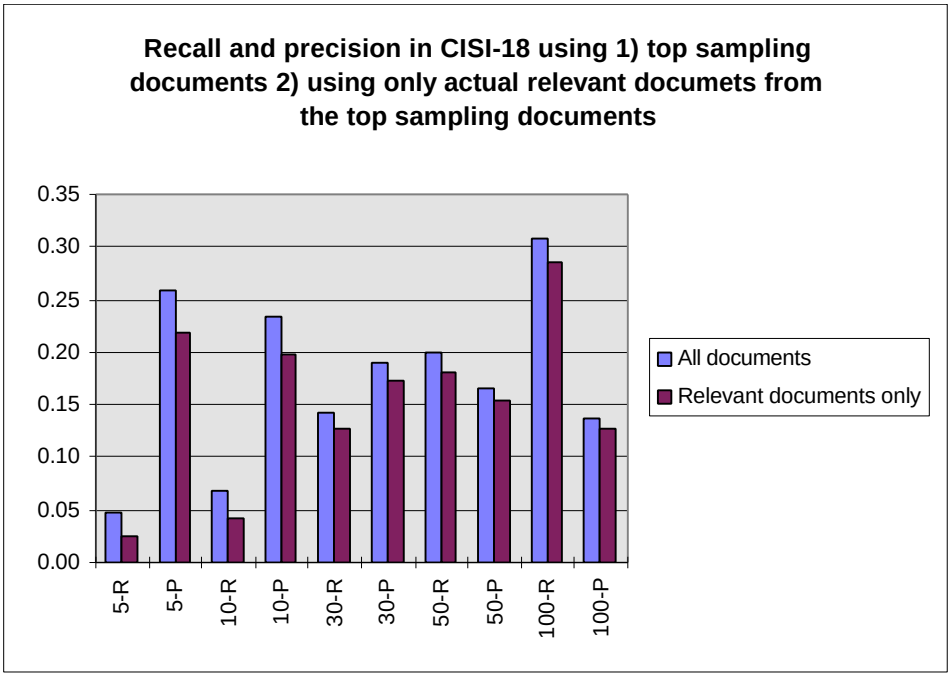


Figure 5.19: The effect of using only actual relevant documents from the sampling collection in the CISI-18 digital library.

5.4.4 Type of links considered from the sampling collection

The final parameter of the method that its effect was tested is the type of links passed to the approximation function. Of course, for the CISI collection which contains only one type of link the effect of this parameter couldn't be tested. However, this parameter was tested in the CACM collection which contains three different types of links. Figure 5.20 presents the results from two equivalent results: one using all types of links and one using only citations. In theory, citations should be the best type of links for the method since they represent direct links between documents, in contrast to cocitations and bibliographic couplings which express indirect relationships.

These results illustrate that there is a very small difference between the two retrieval runs. In fact, the run which uses all types of links produces better results in most of cases.

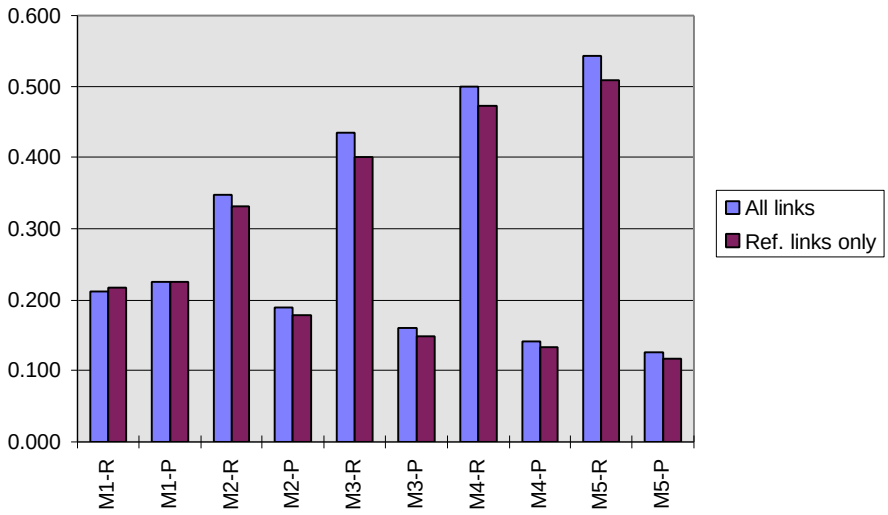


Figure 5.20: Retrieval results using all links and reference links.

5.5 Limitations of the experiments

One possible limitation of the experiments presented and discussed in this chapter is probably the artificial way in which distributed hypermedia digital libraries have been produced. More precisely, documents have been taken from single collections (i.e. CACM and CISI) and, have been clustered and allocated to different sub-libraries using an automatic clustering method. In other words, sub-libraries are not manually produced as would happen in a real environment. However, there is no real reason to suspect that automatically produced sub-libraries invalidate the results of the experiments. The fact is that in real environments documents are not randomly distributed, but similar in content documents tend to be collocated (see section 4.2) and, our artificially created distributed libraries are exactly based on this principle.

In fact, users in real environments may use automatic methods like the one used in our experiment to cluster documents to sub-collections. Besides, if it is accepted that it is a sign of good authorship and that generally people store closer (i.e. in the same sub-libraries) closely related document, then the clustering mechanism which was used to produced distributed versions of the single collections simply automates that process that people would do manually in real environments.

A second point which may cause some concern is about the nature of documents and links that have been used in the experiment. In terms of links, they represent actual links that have been manually produced by the actual authors of the documents. Thus, the links that are used in our

experiment are not automatically produced links, but they are manually created links by the authors of the documents when these documents were published in the CACM journal and other journals (for the CISI collection). Nonetheless, it should be pointed out that the CACM and the CISI collections are the most widely used test collections for hypermedia information retrieval experiments.

Another issue which should be considered about the experiments presented in this chapter, is that they are system-centered experiments which evaluate the performance of the collection fusion methods, but they do not evaluate the performance of the users using these methods in real environments. This issue, however, is fully addressed in Chapter 8 which presents a user-centered evaluation of the collection fusion strategies.

5.6 Conclusions

In this chapter a new and novel collection fusion strategy has been presented, discussed and evaluated. The fusion strategy is new because it uses a new set of algorithms and methods to solve the collection fusion problem. It is novel because for the first time it introduces the use of links to solve the collection fusion problem. In the past, several research efforts have shown that retrieval effectiveness can be increased if links are incorporated into classical IR algorithms. Now, it is shown that links can be additionally used to provide necessary information for solving the collection fusion problem.

The link-based fusion strategy is very appropriate and easily applicable in dynamic information seeking environments such as hypermedia digital libraries for two reasons:

- it solves the source selection problem solely by the use of linkage information extracted from local linkbases at run-time. Information from remote collections is not required in order to make the source selection or the merging of the results. So the link-based fusion method is an isolated strategy and therefore is efficient in terms of network and communication resources;
- it does not require any learning phase to be undertaken before it can be utilised. That makes the proposed method applicable to dynamic environments in which other fusion strategies (e.g. the MRDD and QC) which require an expensive learning phase are inapplicable.

The evaluation of the fusion strategy demonstrates that the method is more effective and efficient than other fusion strategies that can be applied under the same conditions. More precisely the evaluation shows:

- ? the link-based strategy constantly performed more effectively than the random approach. Also, these differences were statistically important. This is a very important result which largely supports the link-hypothesis which was explicitly used in this Ph.D. work and stated at the beginning of this chapter.
- ? the link-based strategy performed better than the uniform approach in most cases. Also, in most conditions these differences were statistically significant. It is therefore, the best method for practical operation in real hypermedia-based environments.

It could be said that the fusion strategy presented is a response to the effectiveness and efficiency issues and challenges that have been discussed in previous chapters. This chapter presented a fusion strategy which is original in its own right and contributes to the current state of knowledge about collection fusion strategies and distributed information retrieval in hypermedia digital libraries in general.

At this point, however, it is useful to recall the discussion which was made at the end of Chapter 3. The invention of the collection fusion strategy is one issue but the problem of how to integrate this strategy within an information seeking environment is another. This is a problem which can be viewed as a part of a larger problem of developing effective architectures and interfaces which allow the easy integration of information seeking methods in electronic environments.

In the following chapters, an agent-based OHS architecture will be presented. The open hypermedia system alongside a protocol and information seeking process model provide the framework to be used for integrating different tools, information seeking strategies and interfaces. This is the exact point which incarnates our work on distributed information retrieval and architectures and protocols for open hypermedia systems. In fact, this represents a more holistic view of the problems of information seeking environments. In the following chapters not only an extensible architecture of an open hypermedia system and a method for developing hypermedia digital libraries is presented, but it is also presented an example of how different information seeking strategies and tools can be integrated and can eventually become a part of the information seeking environment.

Chapter 6

An Agent-Based OHS Architecture

This chapter presents an agent-based distributed OHS architecture. The ultimate aim of the architecture is to aid the design of an OHS which can be used as an underlying platform for developing HDLs that address the architectural and information seeking issues discussed previously in this thesis. Three basic characteristics of the architecture are presented and critically discussed. More precisely, the three characteristics which are discussed are: the architecture for creating agents, the communication and coordination model and the agent communication language. The design of the distributed OHS architecture, the author believes, is novel and original. The proposed architecture utilises in an original form an arsenal of concepts and ideas that have been used in agent-based computer science disciplines and, it shapes and tailors them so they can be adopted in the design and development of an agent-based distributed OHS for HDLs. The agent-based OHS architecture deliberately emphasises interoperability. A new OHS protocol is presented in this chapter and further explored in Chapter 7 which discusses a prototype implementation of the OHS architecture. This new protocol offers an efficient and complete solution to the interoperability problem in OHSs.

6.1 Inspiration

The Open Hypermedia System (OHS) architecture presented in this chapter was mainly inspired by Cooperative Knowledge Based Systems (CKBS) and Multi-Agent Systems (MAS). It was also inspired by previous work on models and methods for incorporating elements of Cooperative Distributed Problem Solving (CDPS; Durfee, 1989) into the design of large distributed information systems (Longstaff et al, 1994).

A CKBS is a collection of autonomous and heterogeneous objects[13] which cooperate together in solving problems in a decentralised environment (Deen, 1990). Two objects may be said to cooperate if they exchange data, or an object undertakes a task on behalf of another object. Objects in these systems are called agents (Genesereth & Ketchpel, 1994). An agent could be a data/rule information server, or a program specialised to solve a problem (e.g. the collection fusion problem). MAS (Bond & Gasser, 1988) are similar to CKBS, but they emphasise on the intelligence that agents may have, while on the other hand, CKBS emphasise well-defined architectures and interfaces, as well as efficiency and reliability.

Based on the ideas presented above as an inspiration, in this Ph.D. work, Hypermedia Digital Libraries (HDLs) are considered as decentralised information environments in which different agents (e.g. data servers, link servers, IR tools) cooperate in order to solve an information problem. The process of solving the information problem proceeds in discrete stages and may involve elements such as user interactions, exchange of data (e.g. from a link server to a viewer), solving subtasks (e.g. searching a document collection) etc. This process is coordinated by the information seeker, and the agents which participate will share their knowledge and capabilities in order to achieve the ultimate goal (i.e. to change the state of knowledge of the information seeker).

The framework described above has two significant implications. First, it closely resembles Belkin's ASK information theory presented earlier in Chapter 1. Belkin's theory provides the theoretical framework in which to design highly interactive information seeking environments. Hence, this framework is suitable for HDLs which are highly dynamic and interactive information seeking environments. Second, CKBS, MAS and CDPS supply an arsenal of ideas, which are very useful in addressing the architectural issues identified earlier in this

[13] the term object should not be confused here with the same term as it is used in object orientation.

thesis (e.g. distribution, interoperability and heterogeneity). As it will be shown in Chapter 7, they also supply the necessary framework for addressing the integration problem of information seeking tools and strategies which has been discussed in section 3.7 and at the end of the last chapter.

6.2 Architecture Overview

The agent-based OHS architecture which will be discussed shortly in detail, is intended to be used as the framework and the underlying platform for developing HDLs (Figure 6.1). The conceptual architecture illustrated in that figure presents a HDL which is composed of three smaller distributed HDLs. A HDL based on this architecture is composed of several parts.

? *Information seekers*.

? *Hypermedia*[14] *Agents (HAs)*; HAs are software agents (Nwana, 1996) exchanging messages in the commonly agreed hypermedia agent communication language. The one and only criterion for agenthood is whether or not a program can communicate by exchanging messages using the commonly defined language, i.e. *any program* which can communicate in this language is a hypermedia agent.

? *Raw data*; i.e. the actual information objects which may change the state of knowledge of information seekers.

? *Virtual Hypermedia Agents (VHAs)*; VHAs are files storing meta-data for different purposes. The information which is stored in VHAs, first, allows HAs to instantiate raw data (e.g. preferred viewer to present data, presentation specifications). Second, some types of VHAs may capture information to structure effectively the information space (e.g. sets of other relevant VHAs, collection of VHAs which comprise an application). Third, VHAs store information to assist (after their instantiation from corresponding HAs) interoperation between HAs (e.g. protocols). Finally, VHAs can capture knowledge which may support information seekers in their searching activities (e.g. the keywords of a document which appear to be the most important).

---

[14] the adjectival complement "*hypermedia*" indicates that these software agents abide by the common rules defined by the agent-based OHS architecture. However, any software agent which is incorporated into the architecture is characterised as *a "hypermedia"* agent.

? *Meta-data*; i.e. the data which are produced manually or automatically over the raw data (e.g. linkbases, inverted indexes).

? *The Agent Communication Language*. This is the language used by HAs to communicate with each other and to exchange data and requests for services. The communication language decouples implementation from interface. As long as HAs abide by the details of the language, it does not matter how they are implemented.

? *Communication architecture*; i.e. the method used by HAs to handle their communication. Agents can communicate directly, or they can communicate indirectly using specialised agents (called facilitators).

? *Coordination and cooperation methods*; i.e. the model used by HAs to manage cooperation and coordination.

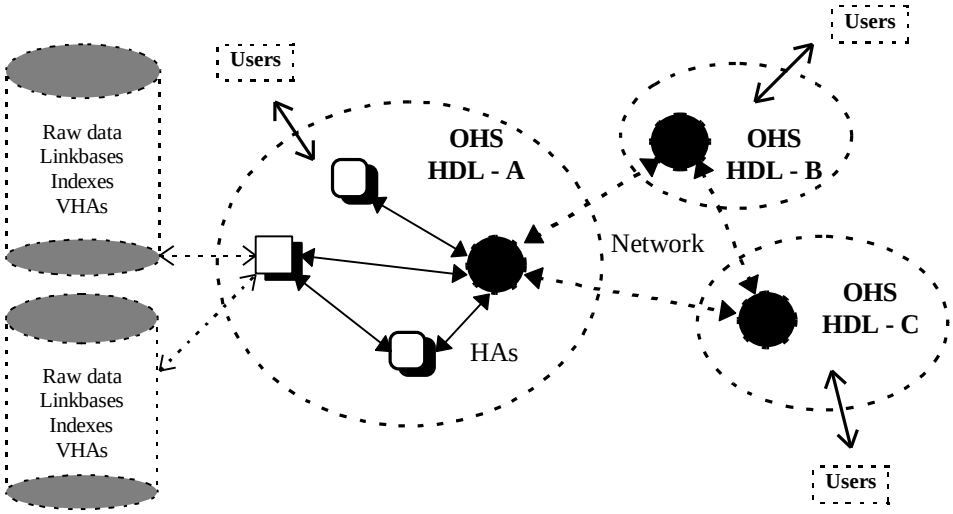


Figure 6.1: An overview of the agent-based OHS architecture for HDLs.

The conceptual architecture outlined above introduces a generalised framework which is independent of any particular hypermedia model or system. For instance, the Microcosm's filters or the HyperDisco's workspaces discussed in Chapter 3 can be easily mapped into this agent-based conceptual framework (e.g. Microcosm's filters as hypermedia agents).

The need, however, to become more specific in explaining this conceptual agent-based architecture and, also the need to develop and evaluate a prototype agent-based OHS and

HDL application, was the motivation behind using Dexter as the starting point for the design of a prototype agent-based OHS. The Dexter model was used in the identification and definition of VHAs and HAs. However, our agent-based OHS is a loose interpretation of the Dexter model, and the concepts, methods and approaches that are introduced by the conceptual architecture in this chapter do not hold explicitly to the Dexter model, but they can be generalised and applied equally to other hypermedia models.

The following sections discuss in detail the principles and the basic characteristics of the agent-based OHS which is based on the conceptual architecture outlined in this section. More precisely, the architecture, data model and the methods for creating hypermedia agents (HAs) and virtual hypermedia agents (VHAs) are discussed (6.3 & 6.4). Then, the communication and coordination architecture are presented (6.5). Finally, the agent communication language is discussed (6.6). The implementation of the prototype OHS and HDL is presented in Chapter 7.

6.3 Hypermedia Agents and Virtual Hypermedia Agents

There is an overuse of the word agent and there is in fact a growing and heterogeneous body of research being carried out under this banner (Wooldridge & Jennings, 1996). Therefore, in this section HAs are closer examined. Also their differences from VHAs are explained because this is a key point for the understanding of the architecture.

HAs are software agents, *i.e. executables, programs, processes*. Usually each hypermedia agent is designed and developed so it can perform one specialised task. The type of the task and the key decisions of how the task is actually performed are usually predetermined by the HAs designers and it is explicitly specified and hardwired into their implementations. *In that sense, HAs are not intelligent agents*. Taking any conventional computer program and changing it so it can exchange messages in the OHS architecture's *commonly defined language*, is adequate to convert this program into a hypermedia agent. Information seekers in HDLs will usually have at their disposal a limited number of HAs (i.e. programs).

It is helpful to think of a HA as the manager of a Virtual Knowledge Base (VKB). Communication between HAs or actions and tasks undertaken by HAs are usually in respect

of their VKB. So, HAs are basically useless without a VKB being "loaded[15]". Therefore, hypermedia agents must be able to "load" a VKB and, also must be able to undertake several actions in respect of or using their VKB. VHAs are exactly *the files which compose the VKB of a HA*. Figure 6.2 illustrates HAs and VHAs and the corresponding analogy with object orientation and the "Prolog interpreter" example.

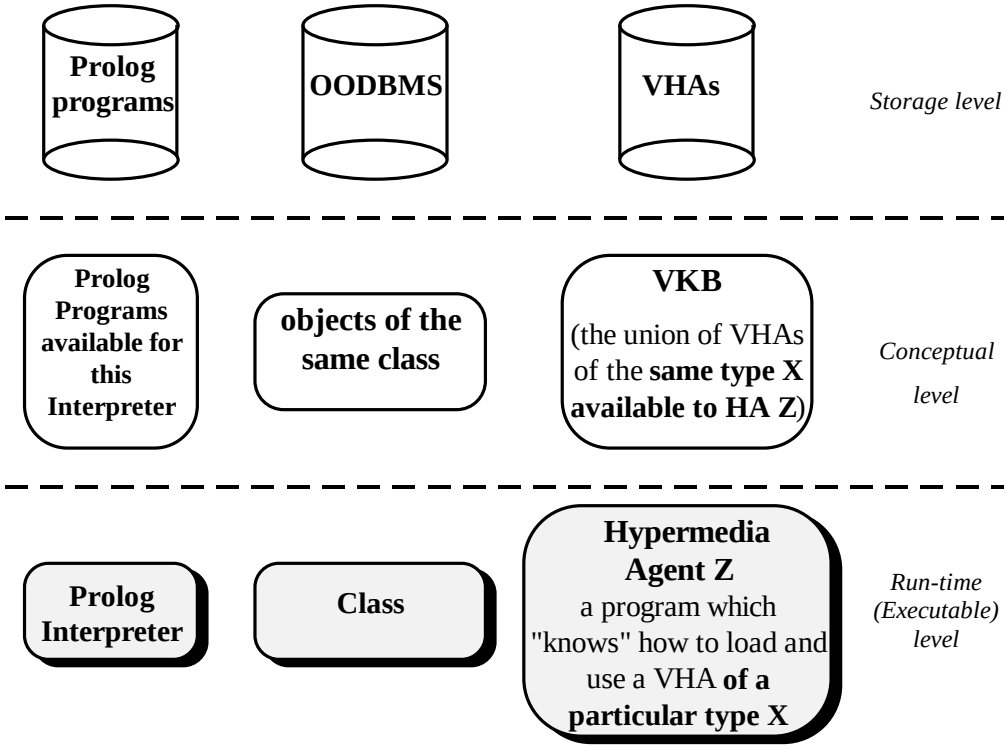


Figure 6.2: A model explaining HAs, VHAs, VKB and their dependencies.

It is important to note here that, although the VKB of a HA is usually composed of multiple VHAs, each particular time a *single* VHA is active. Most of the times communication and actions undertaken by the HA will be mostly in respect to this single, *currently instantiated*, VHA. Therefore, in describing communication sometimes the terms HAs and VHAs may be used interchangeably. Of course, at the physical level communication takes places between the HAs (i.e. the programs), but at the semantic level communication takes place between VHAs

[15] In the same way, for example, that a Prolog interpreter is useless before a Prolog program is being loaded.

(Figure 6.3). Note also that although HAs are relatively static (since they are programs and can change only through a redesign/recompilation process), VHAs are completely dynamic and they can be modified as a result of communication.

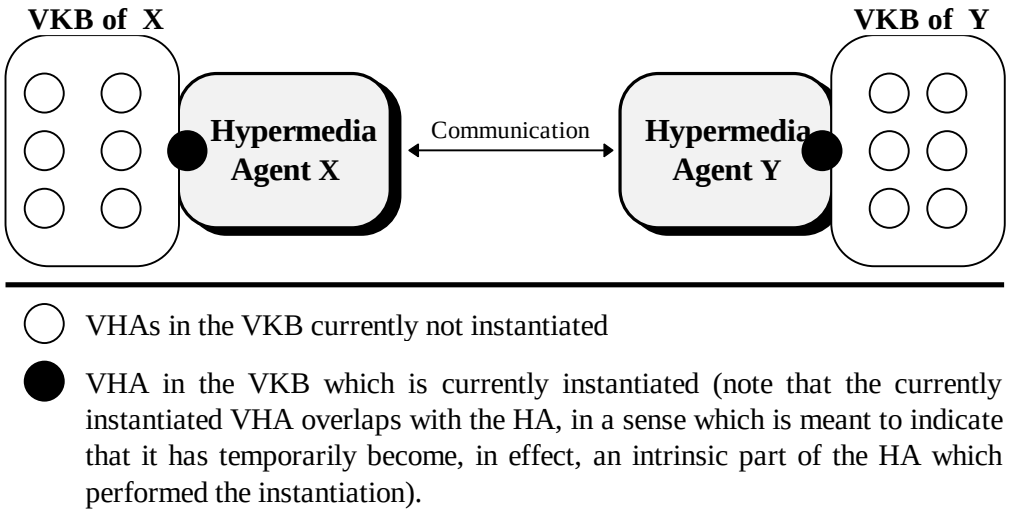


Figure 6.3: An overview of how communication between HAs/VHAs is conducted in the OHS architecture.

VHAs capture useful knowledge for different purposes. In this sense, a VHA file can be seen as a type of shadow file (Davis, 1995). Additionally, they capture this knowledge without changing the original data (e.g. without imposing any markup in the data).

The main purpose of VHAs is to capture information about and to "represent" raw data. So, a VHA in our architecture must exist to facilitate the instantiation of each information object. The first important consequence of this approach is that instantiation[16] of an information object in our agent-based OHS architecture happens as a result of the instantiation of the corresponding VHA (Figure 6.4).

[16] the term instantiation is used as it is defined in Dexter, i.e. the process which leads to the presentation of data to the users.

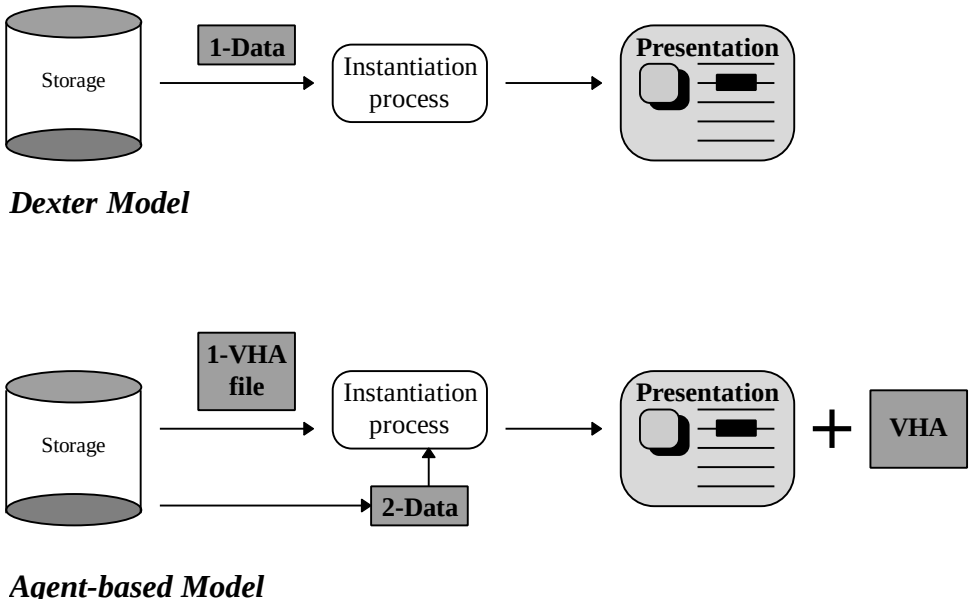


Figure 6.4: Instantiation of an information object in the Dexter and our agent-based OHS architecture.

The disadvantage of instantiating data through the instantiation of corresponding VHAs, is that a VHA file must be created for each information object file. Therefore, users should be provided with the necessary tools to perform the automatic development of VHAs. Ideally, the creation and management of VHAs should occur entirely in the background[17], as a result of using tools for managing, organising and processing raw data. Later in this thesis (Chapter 7, section 7.3.2), some examples are presented demonstrating how useful knowledge can be easily and automatically inserted into VHAs.

6.4 Architecture and Language for Creating VHAs

The model for creating VHAs aims to increase modularity and is illustrated in Figure 6.5. Every VHA is divided into two parts: an *agent head* and an *agent body*. The agent head information is independent of the information object that the VHA represents. In contrast to the agent head, the agent body stores information about the actual contents (e.g. the storage address of the data, its type etc.). In the rest of this section VHAs are described and their specifications are also given.

[17] in the same way, for example, that an operating system modifies in the background without notifying the user the 'last modify date' of a file that was opened, processed and finally saved.

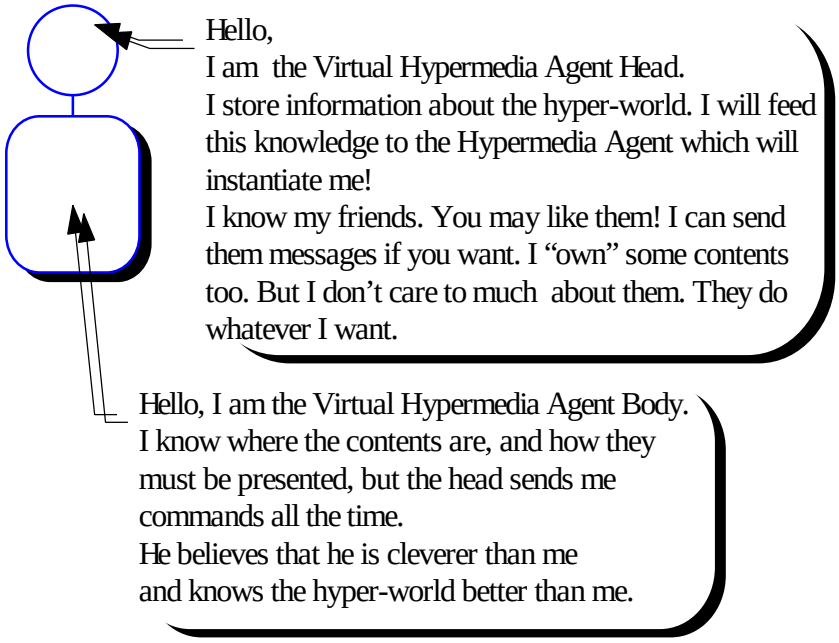


Figure 6.5: The modular architecture for creating VHAs.

Virtual Hypermedia Agent Head

Every VHA must have an ***VHA_ID*** which will allow it to be uniquely identified. The VHA_ID will be in the form <protocol://OHS-name/full-path-file-location>. Also every VHA should specify its type ***[VHA_TYPE]***. This information will be used to identify the most appropriate HA to instantiate the VHA. VHAs are not static but they are dynamically changed as a result of exchanging messages. We define the entity ***MESSAGE*** to represent a message in this agent communication message. Every VHA maintains a message list to keep messages from/to other agents. We call this list the ***MESSAGE_LIST*** of an agent. The following definitions can be now made (the entity KQML message will be later explained):

MESSAGE

Message: KQML Message

MESSAGE_LIST

AgentMessageList: seq. MESSAGE

VHAs maintain a list of other agents they are able to communicate with. Communication happens when services or data are required which are not available. For example, a message

will be sent from an HA specialised in distributed searching to another HA specialised in solving the collection fusion problem. We call this list the ***ACQUAINTANCE_CIRCLE*** of an agent. The entity ***FRIEND*** is defined to represent one member in this acquaintance list. Note that this is a dynamic list which will be automatically updated when for example new libraries/agents are introduced into a system.

FRIEND

FriendAddress: VHA_ID
FriendType: VHA_TYPE
FriendCoopValue: VALUE_OF_PAST_COOPERATION

ACQUAINTANCE_CIRCLE

FriendList: seq. FRIEND

Finally, each VHA has a list of agent specifications. We call this list the ***HEAD_SPECIFICATION_LIST***. We define the entity ***HEAD_SPECIFICATION*** to represent one member in the list. HEAD_SPECIFICATION takes values from the given set ***[HEAD_SPECIFICATIONS]***.

HEAD_SPECIFICATION

AgentSpecification: HEAD_SPECIFICATIONS

HEAD_SPECIFICATION_LIST

AgentSpecificationsList: seq. HEAD_SPECIFICATION

We can now define the head of a VHA as:

AGENT_HEAD

AgentID: VHA_ID
AgentType: VHA_TYPE
aMessageLIST: MESSAGE_LIST
aFriendLIST: ACQUAINTANCE_CIRCLE
aSpecificationLIST: HEAD_SPECIFICATION_LIST

Virtual Hypermedia Agent Body

The definition of the **AGENT_BODY** is basically equivalent to the definition of the Dexter's storage layer BASE_COMPONENT, as this is formally defined by Halasz & Schwartz

(1990). This is the first relation of our agent-based OHS to the Dexter model. However, the modular architecture of VHAs makes possible different definitions for the AGENT_BODY as a means to incorporate other hypermedia data models in the system. The following definition of the AGENT_BODY is based on the Dexter model and is the one used in the prototype system which will be described in the next chapter:

AGENT_BODY

ComponentID: UID
ComponentType: COMPONENT_TYPE
CompPresentationSpecs: COMPONENT_PRESENT_LIST
Anchors: seq. ANCHOR
Attributes: COMPONENT_SPEC_LIST

Finally, having defined the agent head and body each virtual hypermedia agent (VHA) in the architecture can be minimally[18] defined now as the combination of an agent head and an agent body:

VIRTUAL HYPERMEDIA AGENT

VHAgentHead: AGENT_HEAD
VHAgentBody: AGENT_BODY

Implementation of VHAs

Having specified VHAs it is useful to discuss how VHAs could actually be implemented.

- The first method is to implement each VHA as a separate file using a knowledge representation language (e.g. Prolog, AION-DS). This approach has the advantage that an inference engine can be used directly on the knowledge captured in VHAs. Obviously, this method is useful only if the HAs which instantiate VHAs can "load" and process files in the knowledge representation language used to create the VHAs.
- Another implementation method is using a relational database to store different elements of VHAs as attributes in tables. This method has the known advantages of relational databases for managing and accessing information.

[18] some types of VHAs, as will shown in the next section, may add to this baseline specification.

? Finally one could use a specialised language to create VHAs and store them in the file system as individual files. This approach is the simplest since it does not require any extra software. It could be also the most efficient if simple processing of VHAs is envisaged. On the other hand, if concurrent access to multiple VHAs is required (e.g. find all the documents having 'John' as author), this method is obviously the least efficient because it will require access to multiple files.

The third method from those outlined above was used in the implementation of VHAs in the prototype system which will be presented in the next chapter. A specialised language called Virtual Hypermedia Agent Markup Language (VHAML) was defined and used to create VHAs. Figure 6.6 illustrates an example of an VHA file written in VHAML. This VHA file "represents" an HTML document (\cacmlib\machine1\F_1_0125.htm, see the <BASEID> tag) and captures information which is useful to instantiate and display the document (e.g. Netscape is declared as the preferred viewer, maximise window when displaying this document to the user), facilitates communication with other hypermedia agents (e.g. addresses and protocols) and may support information seekers in their searching activities (e.g. the most important keywords of the HTML document and, the cluster in which this HTML document belongs).

```
<AGENT_HEAD>
      <AID>\\CACM-1\cacmlib\machine1\F_1_0125.tat</AID>
      <TYPE> TEXT </TYPE>
      <NAME>Polynomial Transformer (Algorithm 29) </NAME>
      <ACQUINTANCE>
            <FRIEND> \\CACM_2\LIBRARY</FRIEND>
            <FRIEND> \\CACM_4\LIBRARY</FRIEND>
      </ACQUINTANCE>
      <MESSAGE_LIST>
      </MESSAGE_LIST>
      <HEAD_SPEC_LIST>
            <AUTHOR> MIKE </AUTHOR>
            <PROTOCOL> NETBIOS </PROTOCOL>
            <PRIORITY> HIGH </PRIORITY>
      </HEAD_SPEC_LIST>
</AGENT_HEAD>

<AGENT_BASE>
      <BASEID>\cacmlib\machine1\F_1_0125.htm</BASEID>
      <BASE_TYPE> HTML </BASE_TYPE>

      <COMPONENT_SPECS>
            <SPEC><LIB>CACM_1.dxa</LIB></SPEC>
            <SPEC><CLUSTER>C1_0127.pri</CLUSTER></SPEC>
             <SPEC><KEYWORDS> transform polynomial algorithm
                  </KEYWORDS></SPEC>
      </COMPONENT_SPECS>
      <PRESENTATION_SPECS>
            <SPEC><WINDOW> MAXIMIZE </WINDOW></SPEC>
            <SPEC><VIEWER> NETSCAPE </VIEWER></SPEC>
     </PRESENTATION_SPECS>

</AGENT_BASE>
```

Figure 6.6: A VHA file written in VHAML.

6.5 Types of Hypermedia Agents and VHAs

The second influence of the Dexter model is on the definition of HAs. Figure 6.7 shows the basic types of HAs which are defined in our OHS. It also illustrates their derivation in respect to the three layered architecture of the Dexter model.

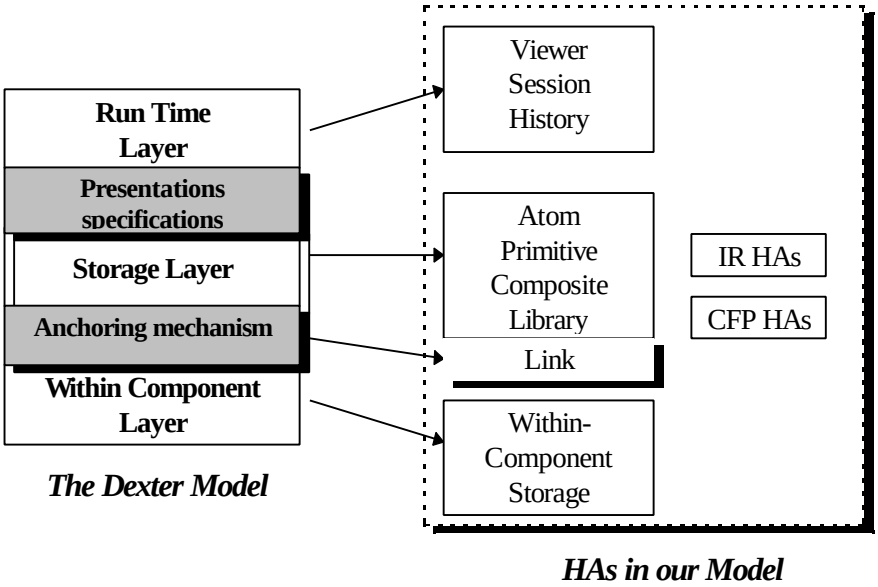


Figure 6.7: HAs in our OHS architecture and their inspiration in respect to the three layered Dexter architecture.

Each HA illustrated in Figure 6.7 has a specific role and provides some well defined services to the rest of the HAs. It is not obligatory to develop all the HAs before an OHS can be realised. For example, the IR agent, which is responsible for providing searching services using analytical methods, may not exist if an information seeker wants to use browsing strategies only.

Note also that Figure 6.7 illustrates the *classes* of HAs. An OHS may include more than one HA from each class. Also, HAs from the same class may not be implemented in the same way. The actual implementation of a HA is a characteristic which is deliberately kept outside the agent-based OHS architecture. However, all the agents from the same class must provide functionality according to the description which is given below.

? *Viewer* HAs should provide the services of data viewers to other HAs. In other words a viewer HA must be able to display data to the users.

? *Session* HAs track, manage and store information about the current hypermedia session and information about active instantiations. Also, session HAs play the special role of facilitators (section 6.6).

? *Atom* HAs must be able to instantiate atom VHAs. Atom VHAs "represent" and store information about raw data (i.e. text, graphics etc., see Figure 6.6 for an example of an atom VHA).

? *Primitive* HAs must be able to instantiate primitive VHAs. Primitive VHAs have information (i.e. pointers) which aggregate related atom VHAs in a single object (i.e. a set). In other words, primitive VHAs do not "represent" raw data, but they only have the necessary information to allow primitive HAs to present a cluster (set) to the information seekers. Figure 6.8 shows an example of a primitive VHA. In contrast to the VHA file in Figure 6.6 (which illustrates an atom VHA representing an HTML document), the primitive VHA file in Figure 6.8 does not "represent" raw data (note that the <BASEID> tag is missing), but rather defines a cluster which is composed of three atom VHAs (defined within the <ATOM_LIST> tag).

? *Composite* HAs must be able to instantiate composite VHAs. Similarly, to primitive VHAs, composite VHAs do not represent raw data[19]. Composite VHAs store information to organise other VHAs in a hierarchical manner.

? *Library* HAs must be able to instantiate library VHAs. Library VHAs maintain a list of all the VHAs members of the library.

? *Link* HAs manage the linkbase and generally provide what is described in OHSs as the link services. The link hypermedia agent is the agent which stores and manages the links between documents and resolves a link when an anchor of an information object is activated.

? *Storage* HAs provide storage services (e.g. a storage HA must be able to access, open, read a file from a given storage device and send its contents to other HAs).

? *Within-component* HAs interpret the internal structure of data (e.g. translate from an unknown data format to another which is recognized by viewer agents).

---

[19] a careful examination of the specifications (given on the end of this section) of the primitive, composite and library VHAs, shows that these types of virtual hypermedia agents also can "represent" raw data (because they include an AGENT_BODY). However, this is not their main purpose and therefore this capability is not emphasised. The main purpose of these types of VHAs is to organise and structure the hyperinformation space using sets, hierarchies and collections.

? *Information Retrieval* (IR) HAs provide analytical searching services.

? *Collection Fusion Problem* (CFP) HAs solve the collection fusion problem on behalf of other agents (e.g. an IR HA).

```
<AGENT_HEAD>
      <AID>\cacmlib\machine1\C1_0043.pri</AID>
      <TYPE> PRIMITIVE </TYPE>
      <NAME>C1_0043</NAME>
      <ACQUINTANCE>
            <FRIEND> \\CACM_2\LIBRARY</FRIEND>
      </ACQUINTANCE>
</AGENT_HEAD>

<AGENT_BASE>
      <COMPONENT_SPECS>
            <SPEC><LIB>CACM_1.dxa</LIB></SPEC>
            <SPEC><COMPOSITE>1_107.cmp</COMPOSITE></SPEC>
            <SPEC><KEYWORDS> pars decision tabl preced requir </KEYWORDS></SPEC>
   </COMPONENT_SPECS>
   <ATOM_LIST>
            <ATOM><CID>F_1_1548.tat</CID><CN>Parsing of  Decision Tables </CN>
            </ATOM>
            <ATOM><CID>F_1_2492.tat</CID><CN>The Development of Decision
                  Tables via  Parsing of Complex Decision Situations  </CN>
            </ATOM>
            <ATOM><CID>F_1_2982.tat</CID><CN>The Storage Requirement in
                        Precedence Parsing  </CN>
            </ATOM>
   </ATOM_LIST>
</AGENT_BASE>
```

Figure 6.8: A primitive VHA file.

Figure 6.9 presents an overview of the agent-based OHS architecture illustrating the basic communication paths between HAs. This figure shows how users can have at their disposal several HAs to interact with the OHS. It also illustrates that different viewers HAs (having initially different levels of awareness) may be incorporated into the architecture using a different software agent engineering method (this feature will be explained in detail in the next chapter). Note also that Figure 6.9 shows the "logical" communication paths between HAs. Actual communication may happen only through facilitators, if a particular communication architecture is adopted (see section 6.6).

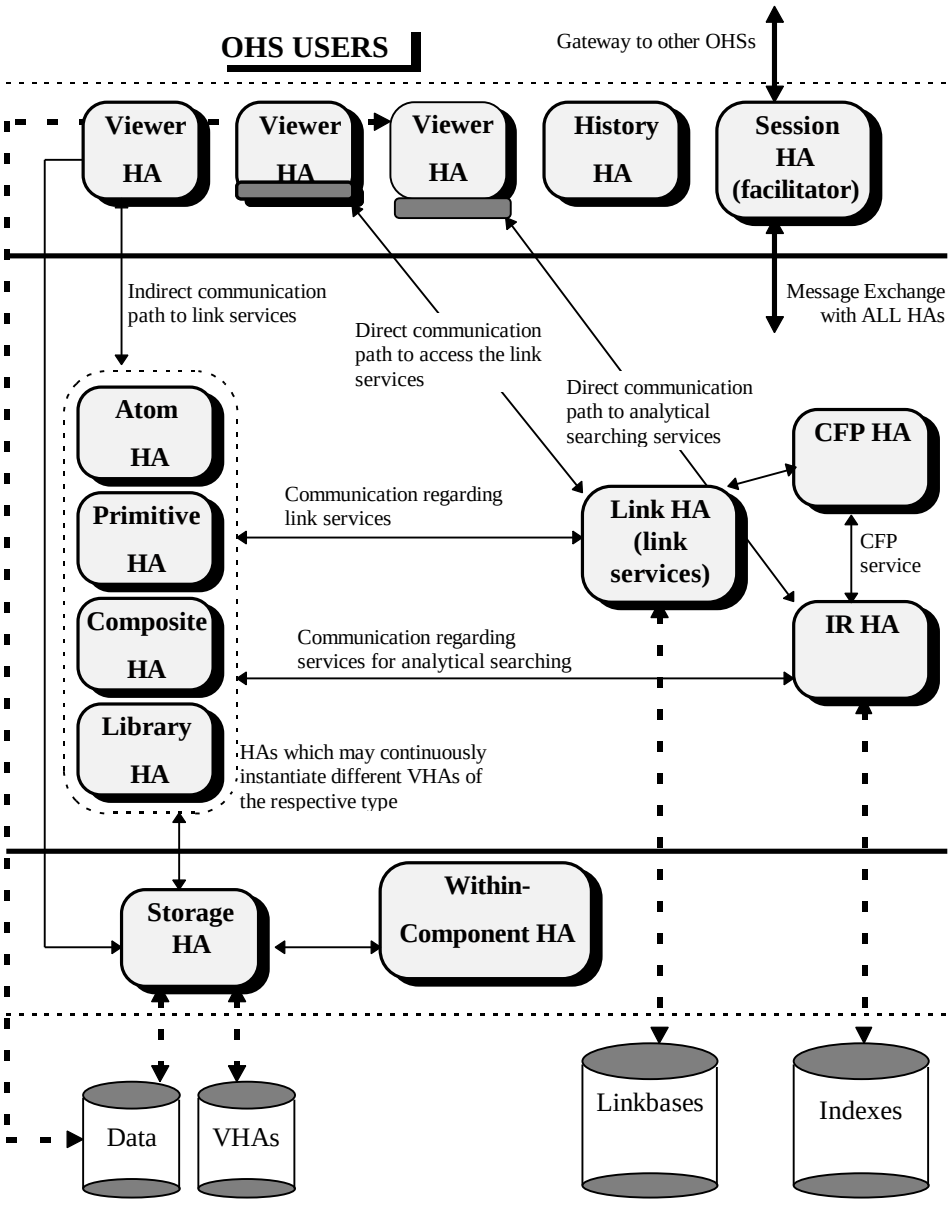


Figure 6.9: Overview of the agent-based OHS architecture.

Note that not all HAs are meant to continuously instantiate different VHAs. Instantiation of different VHAs happens only from HAs in the storage layer (i.e. atom, primitive, composite and library). HAs in the within-component and run-time layer and also the IR, CFP and Link HAs provide their services without instantiating VHAs (i.e. VHAs do not exist for these types

of HAs[20]). Based on the definitions given in the last section and the discussion above we can now define the **ATOM_VHA, PRIMITIVE_VHA, COMPOSITE_VHA, LIBRARY_VHA** as**:**

ATOM_VHA

AtomHead: AGENT_HEAD
AtomBody: AGENT_BODY

PRIMITIVE_VHA

PrimitiveHead : AGENT_HEAD
PrimitiveBody : AGENT_BODY
Atoms: seq. ATOM_VHA

COMPOSITE_VHA

CompositeHead : AGENT_HEAD
CompositeBody: AGENT_BODY
CompAtoms: seq. of ATOM_VHA
CompPrimitives: seq. of PRIMITIVE_VHA
CompComposites: seq. of COMPOSITE_VHA

LIBRARY_VHA

LibraryHead : AGENT_HEAD
LibraryBody: AGENT_BODY
LibraryAtoms: seq. of ATOM_VHA
LibraryPrimitives: seq. of PRIMITIVE_VHA
LibraryComposites: seq. of COMPOSITE_VHA
LibraryLibraries: seq. of LIBRARY_VHA

The definitions for atom, primitive, composite and library VHAs given above, in effect describe a hypertext data model which includes:

? raw objects (atom VHAs);

? sets (clusters) of raw objects (primitive VHAs);

? hierarchies of clusters and raw objects (composite VHAs);

[20] in reality a *single* VHA does exist for these HAs, but its instantiation takes place only once when these HAs are initialised. On the other hand, the atom, primitive, composite and library HAs continuously instantiate different VHAs as information seekers interact with the OHS.

? collections of raw objects, clusters and hierarchies (library VHAs).

This data model introduces a small change to the Dexter's data model. More precisely, the Dexter model defines in the storage layer three basic entities (i.e. *atoms, composites and links*). The agent-based architecture which is presented here adds two organisational entities at the storage layer (i.e. *primitives and libraries*). The reasons for introducing this change are outlined below.

? Composites are defined in the Dexter model as: (a) one level collection of atoms (b) as a hierarchical organisational entity. The definition of composites in the Dexter model is a powerful recursive definition which can be interpreted in different ways. However, the notion of composites is not very clear in the Dexter model both in terms of how it should be implemented and also for what purposes exactly it has been introduced. In our architecture, an attempt is made to clarify the role of composites by separating the two roles previously outlined between the primitive VHAs and composite VHAs respectively.

? The need to have a single entity, i.e. the library VHA, which can store a list of all the information objects which are members of the library.

6.6 The Communication and Coordination Architecture

The agent-based OHS architecture which is described in this chapter assumes the following model for message transport:

? HAs are connected by communication links that carry discrete messages;

? when an HA receives a message, it knows from which HA the message arrived;

? an HA can direct a message to a particular outgoing communication link;

? messages to a single destination arrive in the order being sent;

? message delivery is reliable.

At the implementation level this abstraction can be implemented using temporary TCP/IP connections over the Internet, e-mail messages etc.

There are also different ways in which communication between agents can be coordinated. One method is for HAs to connect directly to one another, handling their own communication. This communication architecture is the simplest but it requires each HA to be able to send and

receive messages from HAs in remote machines. Also, in applications where HAs have to communicate heavily, it might not be particularly efficient. Another way to handle communication is to communicate through specialised agents. The agents playing this special role are called facilitators (Genesereth & Ketchpel, 1994). Usually communication through facilitators will be more efficient because optimisation techniques (e.g. collecting and transmitting multiple messages in one connection) may be applied to achieve higher efficiency.

In our architecture session HAs play the role of facilitators for their "local" HDLs (Figure 6.10). The decision of which HA will play the role of facilitator has to do with the decision of which HA must be present in every HDL session.

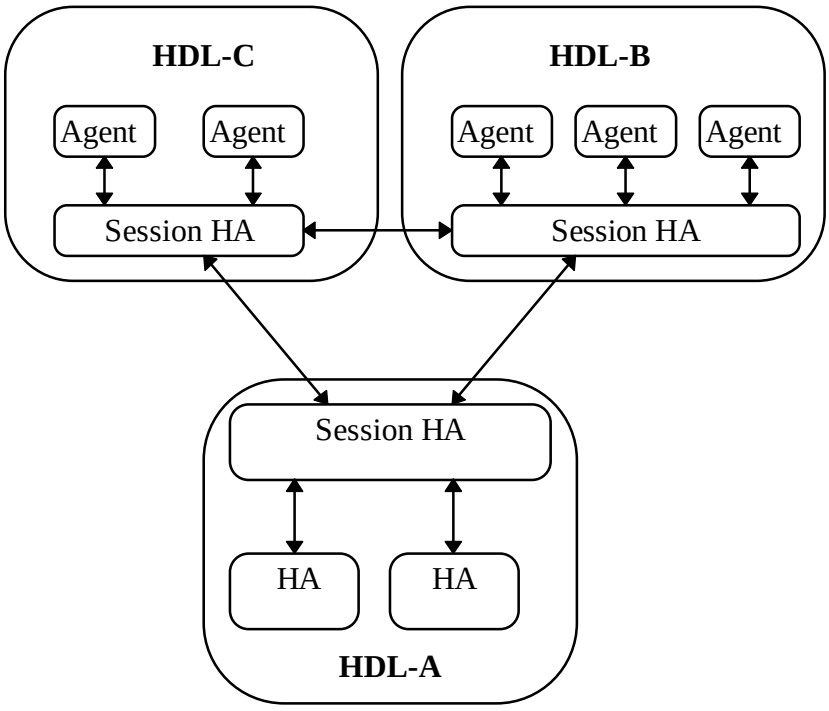


Figure 6.10: A federated communication architecture using session HAs as facilitators.

6.7 The Agent Communication Language

6.7.1 The KQML Language

KQML (Finin et al, 1993) is the language used as the agent communication language in our architecture. KQML offers a variety of message types (called performatives) that express an attitude regarding the content of the exchange. A performative in KQML is a string following the syntax shown in Figure 6.11. Every performative has a defined number of parameters. KQML reserves the names and meanings of some parameters essential to many performatives. Figure 6.12 shows two typical KQML messages and Table 6.1 explains the meanings of the reserved performative parameters.

<performative>	::=	(<word> {<whitespace> :<word> <whitespace> <expression>}*)
<expression>	::=	<word> | <quotation> | <string> | (<word> {<whitespace><expression>}*)
<word>	::=	<character><character>*
<character>	::=	<alphabetic> | <numeric> | <special>
<special>	::=	< | > | = | + | - | * | / | & | ~ | _ | - | % | : | . | ! | ?
<quotation>	::=	'<expr> | '<comma-expr>
<comma-expr>	::=	<word> | <quotation> | <string> | ,<comma-expr> | (<word> {<whitespace><comma-expr>}*)
<string>	::=	"<stringchar>*" | *<digit><digit>*"<ascii>*
<stringchar>	::=	\<ascii> | <ascii>-\-<double-quote>

Figure 6.11 Syntax of KQML in BNF.

Parameter	***Meaning***
:sender	the actual sender of the performative
:receiver	the actual receiver of the performative
:in-reply-to	the expected label in a response to a previous message
:reply-with	the expected label in a response to the current message
:language	the name of the representation language of the :content
:ontology	the name of the ontology (i.e. a set of formal specifications) assumed in the content parameter
:content	the actual information being communicated

Table 6.1: Reserved parameters of KQML messages.

```
( register      :sender  TEXT_ATOM
                :receiver LINK_AGENT
                :reply-with  tid1
                :in-reply-to lid1
                :language X
                :ontology ONT-Y
                :content  TEXT_ATOM_ADDRESS
)

( ask-all       :sender  TEXT_ATOM
                :receiver LINK_AGENT
                :reply-with  id0
                :language X
                :ontology ONT-Z
                :content Anchor( UID  )
)
```

Figure 6.12 Two examples of KQML messages.

6.7.2 KQML in our Architecture

General

The standard definition of KQML reserves a set of performatives. In using the KQML language in our architecture we tried to adhere to this defined set as much as possible. However, as will be shown later, in our model we extended KQML by adding three new performatives to address needs which are not addressed by the reserved set of KQML performatives.

What is crucial to understand about KQML is that it is indifferent to the format of the actual information itself (i.e. the value of the `:content` parameter). Thus, KQML messages will contain sub-expressions in other so-called “content languages”. In practice this means that when using KQML, messages in different languages (e.g. Prolog, OHP) can be communicated in the `:content` parameter. The `:language` and the `:ontology` parameters inform the receiver of a KQML message how it should interpret the `:content` of the message.

This feature of KQML has an important implication. Existing hypermedia protocols such as OHP can be used as the "content language" to exchange messages between HAs. Figure 6.13 illustrates how KQML can be used to sent an OHP message. What KQML offers to protocols such as OHP is a linguistic and a semantic level which are not captured by the OHP itself. For example, Anderson in his recent critique of OHP (1997) identifies some problems of the OHP (e.g. how a viewer introduces itself to a link server). KQML performatives are bounded with semantics (i.e. the meaning and expected action of the performative) and a linguistic level (i.e. the name of performative) which enhance the bandwidth and quality of communication.

```
( achieve       :sender  DCS_MICROCOSM
                :receiver VIEWER-X
                :reply-with  id1
                :in-reply-to  id0
                :language OHP
                :ontology OHS
                :content  \Subject LauchDocument
                          \DocumentName DocName.txt
                          \ReadOnly True
                          \DocumentType ASCII
                          \DataCallBack False       )
```

Figure 6.13: Using KQML to send an OHP message.

To explain better what KQML offers into content languages, consider the KQML message in Figure 6.13. KQML offers the following additional information to the OHP message which is being communicated in the `:content` parameter.

- At the semantic level, the meaning which is captured by the performative *achieve.* The meaning of this performative (*and this meaning is global for all the software agents in the world using KQML*), is that the Sender (S) wants the Receiver (R) to make something true of their environment, in this case to launch document "DocName.txt";
- It explicitly specifies which is the sender and which is the receiver of the message.
- It indicates how the message should be interpreted (i.e. using the language OHP and the ontology OHS). In OHP these two important aspects of communication are not addressed. OHP implicitly assumes that each viewer will accept *only* OHP messages and therefore "knows" how messages should be interpreted. However, a viewer (or general any application) may be able to receive messages in more than one languages and, therefore messages must define how they must be interpreted.
- The `:in-reply-to` and `:reply-with` parameters provide the mechanism to allow the S and the R to establish a series of messages if necessary. This allows agents to engage in a *conversation* instead of a single master/slave communication act. For instance, in the example shown in Figure 6.13 if the VIEWER-X has any problems in instantiating the document, or wants to inform back that the goal has

been achieved, it can send back a message to the LINK_SERVER_MICROCOSM using the label 'id1'.

It should be clear by now that different OHS protocols can be used with KQML. This feature of KQML suits also recent proposals for the development of additional OHS protocols to supplement OHP[21] (Goose et al, 1997; Gronb? k & Wiil, 1997).

In conducting our experiments with the prototype OHS and HDL application presented in the next chapter, a new protocol was introduced to facilitate interoperability between HAs. This protocol is called HAP (Hypermedia Agent Protocol) and it is primarily based on the use of a "content" language which has been introduced in this Ph.D. work, called HACL (Hypermedia Agent Content Language). The decision to introduce a new "content" language was driven by the following.

- OHP only covers the communication between a link server and a viewer in an OHS. Therefore, several types of interoperability (e.g. interoperability at the storage layer or interoperability requesting analytical searching services) are not addressed by OHP.
- the OHS community has consistently expressed the need for reconsidering OHP or producing new protocols.

Performatives used

The full specification of KQML defines 36 performatives. This rich set of performatives allows KQML to be used in a wide range of application areas. However, until KQML was used in this Ph.D. work, it had not been used in hypermedia systems or information seeking environments[22]. A contribution of this Ph.D. work is that it thoroughly investigated the use of KQML in application areas such as OHS and hypermedia digital libraries. In the following paragraphs the result of this investigation is summarised and, a set of KQML performatives is

---

[21] the author, however, disagrees with these proposed methods to develop *multiple* OHS protocols, *each covering a different aspect of OHSs interoperability* (e.g. one protocol for viewer/link server, another protocol for storage interoperability etc.). The author believes that multiple protocols may be used, but each single protocol must be complete (see Chapter 7, section 7.6).

[22] in fact, the use of KQML is very briefly mentioned in De Roure et al (1996), but the author is not aware of any report which presented, at least in a small depth, the use of KQML in OHSs.

presented which effectively addresses the needs of our model (and potentially the needs of OHSs in general). Simple examples of messages are also given detailing the needs addressed by particular performatives. Complete examples of how KQML was used in a prototype agent-based OHS will be given in the next chapter.

register, unregister

These messages are used to help HAs find other HAs. The performative register is sent by an HA to announce its presence to another HA. The `:content` parameter of this message will include the symbolic name and the address of the HA. The message unregister is used to cancel previous register messages. This message will be send when first running an HA (register), and when an HA is terminated (unregister). HAs will usually send these messages only to the facilitator agent. Note also, that a register message will usually follow one or more advertise messages (see next paragraph).

advertise, unadvertise

The advertise message is used when S (sender) wants R (receiver) to know that it can and will process a specific type of message. The unadvertise message is used when S wants R to know that it wishes to cancel a previous advertisement.

These messages are used when a HA wants to advertise or to cancel previous advertisements about the messages that it can and will process (i.e. its capabilities). For example a viewer HA may advertise that it is able to and will present files of a particular format. A link HA may advertise that it can provide link services to other HAs. Note that if the advertise message is sent to the local facilitator all the other local HAs are indirectly informed about the capabilities of the sender. Of course, HAs can send advertise messages eclectically to particular HAs. In that case, the commitment they make holds only for this particular HA.

ask-one, ask-all, stream-all

These three messages have similar meaning: S wants one (all in the case of the ask-all and stream-all messages) instantiations of the `:content` which are true(exist) in R.

These messages are used to request data. One use, for instance, is when a viewer HA (S) wants from a link HA (R) all the anchors for a document (ask-all). The stream-all message is identical to ask-all but it can accept multiple messages as a reply.

tell, untell

The meaning of the tell (untell) message is that the expression in the `:content` exists (not exists) in S.

These two messages are produced as a response to ask-one, ask-all and stream-all messages. Continuing the example introduced before a link HA can send a tell message to a viewer HA providing the anchors for the particular document. If no anchors exist for the particular document, an untell message will be sent. If the request was made using a stream-all and multiple anchors exist, the linker HA may send multiple tell messages to the viewer.

achieve, unachieve

The achieve message is used when S wants R to make something true in their environment. The unachieve message is sent when S wants R to reverse the act of a previous achieve message.

An example of using an achieve message is when an atom HA asks a viewer HA to present an information object to the user. Another example is when a session HA asks an atom HA to instantiate a text VHA (note that this in turn will cause the atom HA to ask from the viewer HA to instantiate the actual data).

error, sorry

These two messages are used to intervene with the normal course of a conversation. The error message will be send when S does not understand a message being sent by R. The sorry message will be send when S understands the message but it can not provide any other response to R.

A viewer HA may send an error message for example if receives anchor information in wrong format, or if an achieve message is syntactically incorrect. A viewer HA which can only display graphics in a particular format (e.g. BMP) and receives a message to instantiate a GIF graphic file from an atom HA, will send back a sorry message.

insert, uninsert, delete-one, delete-all, undelete

These messages are used when an HA wants to change the VKB of another HA.

broadcast, forward

These messages are used to facilitate communication. Using the forward message S can ask R to forward the message to another agent. The broadcast message is used when S wants R to send a message to all the agents listed in its acquaintance list.

evaluate

This is a new performative introduced by our agent-based OHS architecture to address a need which is not addressed by the set of KQML performatives in the official[23] KQML specification. The meaning of this message is that the S of the message asks from the R to evaluate an expression and return the results of the evaluation. During the evaluation process the receiver may use other HAs and whatever action it believes will produce the best results. Information about the methods and the action taken should be also returned together with the results.

In our architecture this message is used from HAs to ask the IR hypermedia agent to evaluate an expression, i.e. to apply IR techniques similar to those discussed in Chapter 5 in order to search for information in the HDL using analytical strategies.

get, set

These two performatives are also new and are introduced by our architecture. The purpose of these messages is to facilitate the communication of HAs with the storage HA.

The performatives that have been presented above are these which have been identified as necessary to adequately describe the OHS and HDL application area. Other, more specialised performatives exist if a particular need is not addressed. And most importantly, KQML can always be extended by adding new performatives.

Finally, in our architecture a new parameter is introduced to address a need which it is essential in interactive systems such as HDLs. Standard KQML does not address time issues in responding to a message. Therefore a new parameter `:ExpireTime` has been introduced in message exchange to indicate the time limits within the sender HA should receive a

[23] the official KQML specification is the one described in Finin et al (1993). However, there is a new proposal for a KQML specification (Lamprou & Finin, 1997). Our use of KQML is largely based on the new proposed KQML specification.

response from the receiver. Each KQML message produced by HAs can use this parameter to indicate that the reply, action, process etc. should take place in certain time limits. This parameter is essential in interactive systems where some messages need quick response (e.g. messages asking to display data).

HAP (Hypermedia Agent Protocol): A new protocol for OHSs interoperability

The key component of the HAP is its "content" language (HACL). The syntax of HACL in BNF is given in Figure 6.14. Note that this BNF definition assumes the definitions of <ascii>, <alphabetic>, <numeric>, <double-quote>, <backslash> and <whitespace>.

HACL message	::=	<subject>({<whitespace> :<indicator><parameter> <whitespace> <expression>*}<whitespace>)
subject	::=	<word>
parameter	::=	<word>
word	::=	<character><character>*
character	::=	<alphabetic> | <numeric> | <special>
special	::=	- | _ | = | + | - | * | / | .
indicator	::=	: | !
expression	::=	<word> | <string> | word {<whitespace><expression>}*
string	::=	"<stringchar>*"
stringchar	::=	<backslash><ascii> | <ascii>-<backslash>-<double-quote>

Figure 6.14: HACL message syntax in BNF.

As it is shown in Figure 6.14, a HACL message has two parts. The first part is *the subject* name which identifies the subject (theme) of the message. The second part is *the parameters* of the subject. Subjects can take as many parameters as they want. Parameters start with "::" or ":!". The second symbol indicates that the parameter is obligatory in order to perform the action correctly. This feature already addresses one important problem of the OHP protocol. The actual value of the parameter can be any expression.

A specific set of subjects has been introduced and their parameters have been defined to facilitate interoperability in our agent-based OHS model (Table 6.2). This set was used in the prototype application which is described in the next chapter. Of course, this set should not be

regarded as complete and, the set of subjects outlined in Table 6.2 can be extended if other OHS or applications have additional needs.

It is crucial to understand that the HACL messages in Table 6.2, should not be interpreted in isolation, but they must be interpreted in conjunction with the KQML performative that each time "carries" the HACL message in the `:content` parameter. The HACL message simply expresses and defines, first, the subject of the communication (e.g. anchors may be a subject) and, second, the parameters of the subject. But the subject alone is not sufficient to create a "meaningful conversation". The attitude or the action regarding the subject is expressed by the KQML message (performative) itself. The attitude or the action specified by KQML is *combined* with the subject and the parameters specified by HACL, to form a *single* message in the proposed OHS protocol.

The same HACL subject can be combined (sometimes using a different set of parameters) with different KQML performatives to construct messages with different meanings. For example, if the HACL subject *anchor* is combined with the ask-one performative and if the sender and receiver of the message is a Viewer and a Link HAs, that means that the details of anchor should are requested. This type of communication concerns the link services interoperability. If the same subject (i.e. anchor) is combined with the insert performative and the receiver of the message is the storage HA, a new anchor must be inserted into the linkbase. This type of communication is regarding storage interoperability. If it is combined with the untell message that means that an anchor does not exist.

Generally, there are probably tens or hundreds of combinations between HACL subjects and KQML performatives which can be created to satisfy a particular need. It is therefore impractical to create a complete list of all the possible messages. In the next chapter, different interoperability experiments which have been conducted are described. In this chapter detailed "conversations" between HAs are presented using the HACL.

Each HA in our prototype OHS must be able to process HACL messages, using an ontology named as DEXTER. The ontology specifies what concepts must be "known" to the HAs and essentially defines how the message should be interpreted. For instance, if a HACL message specifies "WORKSPACE" as a parameter, then this is an unknown concept in the DEXTER ontology and therefore the message can not be correctly interpreted. Of course, another ontology called e.g. HYPERDISCO may exist which can help an HA translate the message

(i.e. by mapping the concept "WORKSPACE" into a library VHA). In general the concept of an ontology can offer formality and great flexibility in an open environment.

There are two basic ways to implement ontologies. One simple way is by "embedding" the ontology specifications into the implementation of the HAs. That means that the interpretation of a HACL message is hardwired into the implementation of HAs. That was the method which was used in our prototype OHS, since it was an aim of this research work to investigate the use of different ontologies. The second way is by using a special and external "ontology component" which is able to interpret messages correctly using multiple ontologies. This method is more flexible and leads to a more elegant solution, but it is more costly.

Figure 6.15 shows the generalised framework which was introduced in this thesis to solve the interoperability problem in OHSs.

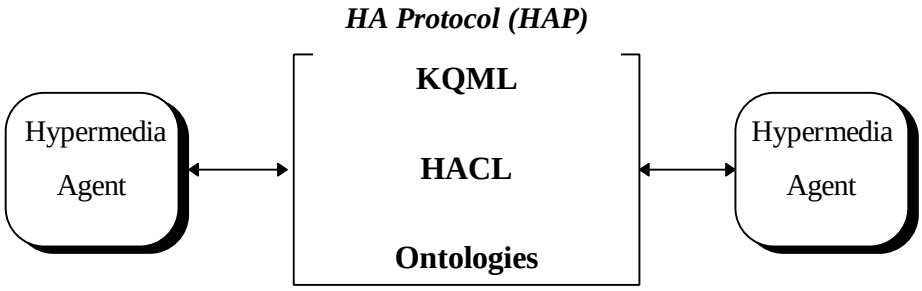


Figure 6.15: Generalised framework for interoperability in OHSs.

Subject	*Parameters*	Explanation
Anchor	*:!UID*	the ID of the component
	::AID	the ID of the anchor
	::Value	the value of the anchor
ResolveLink	*:!UID*	the ID of the component
	:!AID	the ID of the anchor
VHA	*:!ID*	the ID of the VHA
	::Type	the type of the VHA
	::FileContent	the actual contents of the VHA file
Data	*:!UID*	the ID of the component
	::Type	the type of the component
	::FileContent	the actual contents of the data file
InstantiateVHA	*:!ID*	the address of the VHA
	::Type	the type of the VHA
	::FileContent	this parameter can be used to directly exchange the contents of the VHA file
InstantiateData	*:!UID*	the address of the data file
	::Type	the type of the data file
	::FileContent	this parameter can be used to directly pass the contents of the information object
File	*:!Address*	the address of the file
	::Protocol	the type of the file
Search	*:!Query*	the expression to be evaluated
	::Library	libraries to search
	::CFP	method to select libraries
	::cut-off	parameter indicating the cut-off level
	::Result	parameter to communicate the results of a search process
CFP	*::Method*	the collection fusion method to be used
	::Sample	the collection to use for collecting linkage data
Activate	*::Name*	the name of the HA to activate
Profile	*:!Name*	the name of the HA which sends its profile
	:!Type	the type of the HA which sends its profile
	::expression	the expression/keywords describing the profile

Table 6.2: Subjects and parameters in HACL.

Figure 6.16 presents two examples of HAP messages. The OHP messages which are analogous to those examples are also presented. The first example shows how a link server (a Link HA in our model) informs a viewer about its ability to provide the anchors for documents. Note that this first advertise message has as the `:content` parameter another KQML message which essentially gives a "template" of the ask-all messages that the Link HA can and will process. The second example shows how the viewer after receiving the first message, responds by requesting the anchors for a particular document.

HAP MESSAGE

```
(advertise    :sender LINK_AGENT
              :receiver TEXT_VIEWER
              :reply-with id2
              :in reply-to id1
              :ontology DEXTER
              :content
                        (ask-all
                                  :language HACL
                                  :ontology DEXTER
                                  :content Anchor( :!UID ) )
)
```

OHP MESSAGE

```
\subject HeresServices
\Data {\Service HeresAnchorTable}
\Channel Z
```

CASE 1: Advertise services

HAP MESSAGE

```
( ask-all     :sender TEXT_VIEWER
              :receiver LINK_AGENT
              :reply-with id3
              :in-reply-to id2
              :language HACL
              :ontology DEXTER
              :content Anchor( :!UID X )
)
```

OHP MESSAGE

```
\Subject GetAnchorTable
\Channel Z
```

CASE 2: Asking anchors from the Link Server

Figure 6.16: Two examples of HAP (KQML/HACL) and OHP messages.

6.8 Addressing Architectural Issues

At the beginning of this chapter it was mentioned that one of the reasons to introduce the agent-based OHS architecture, is that it could potentially address the architectural issues

which arise in developing hypermedia digital libraries and which have been identified and discussed earlier in this thesis. This section examines in greater detail how the architecture may address these issues (in the rest of this section the term HDL refers to a hypothetical hypermedia digital library system which is based on the agent-based OHS).

Distribution

The agent-based OHS architecture addresses both data and services distribution.

- Service distribution is possible because HAs which actually implement the services of the HDL system can be distributed over a local or wide-area network. The architecture supports the necessary mechanisms to allow the distributed HAs to register and announce their presence to other HAs in remote machines (using the register message). It also makes possible for distributed HAs to advertise their capabilities, i.e. the services they can provide (using the advertise message), so other users or HAs in remote OHSs can make use of their services.

- Data distribution is addressed since HAs can access data from other HDLs. Note, that access to distributed data can be done using the "local" storage HA, but also using storage agents in remote machines (if they advertise their services).

Note, that the architecture not only addresses distribution, but also supports methods which can make this distribution happen efficiently. In particular, two aspects of the proposed architecture, the use of a facilitator agent, and the KQML messages for multistage data exchange (i.e. stream-all), provide the mechanisms for efficient handling of accessing data from remote sources.

Extensibility

The agent-based architecture supports extensibility in different ways.

- New services can be added to the HDL. The mechanism for introducing new services is by introducing new HAs. New HAs can be easily introduced into the system, if designers and developers of the HAs abide by the rules of the agent communication language. Note that the actual details of the HAs implementation are not relevant from an architectural point of view. For example, in Chapters 4 and 5 were discussed different ways for solving the collection fusion problem. Different HAs may exist in a HDL, each solving this problem using different collection fusion

strategies. The common characteristic of all these HAs should be their ability to register and advertise a CFP (collection fusion problem) service.

? New HDLs can register to and extend an already existing HDL. The mechanism to support this type of extensibility is by using the appropriate messages to register and advertise the services and the information of the new HDL.

? The communication language can be extended. This type of extensibility refers to the capability of KQML to be extended in two different ways. First, by introducing new performatives. In our architecture this feature of KQML is used to extend the KQML language with three performatives. Second, KQML can be used to communicate messages in different "content" languages. This feature allows different languages to be used and therefore increases the flexibility and efficiency of the system.

? Other hypertext data models can be incorporated into the architecture. The modular definition for VHAs suggests that it is possible to incorporate different hypertext data models into a HDL based on our architecture. Our decision to keep information relevant to the hypertext model (in the agent body) separate from information regarding the mechanics of the agent based architecture (in the agent head), allows different hypertext models to be introduced, if new agent bodies are defined, and HAs are correspondingly developed to handle them.

Heterogeneity

Our architecture supports heterogeneity in several ways.

? Forms of data (e.g. text, graphics etc.). As far as HAs exist to handle the instantiations of corresponding atom VHAs, multiple forms of atomic data can coexist.

? Implementation of services. As we already mentioned before, the implementation of the services remains completely outside from the architecture and therefore can happen in different ways. However, HAs must name and advertise the services using some commonly agreed names for services.

? Protocols (i.e. different OHS protocols may be used within KQML)

- interfaces (i.e. interfaces for searching). Again this aspect of heterogeneity refers to the ability to incorporate HAs providing the same services, but which are implemented in different ways.

Scalability

Whether the architecture will scale effectively remains an open question. However, the experiences from experiments that have been conducted to test the OHS and HDL application (see next chapter) and, experiences from other agent-based systems reported in the literature indicate that:

- decentralised architectures are conducive to the scalability of the developed systems;
- communication between agents can be handled efficiently, especially if federated architectures are used.

Interoperability

Any agent-based architecture relies heavily on interoperability. In fact, in most agent-based systems interoperability is not only an aim but an intrinsic feature of the system. Not surprisingly then, our OHS architecture supports interoperability at all three levels (in the next chapter three interoperability experiments are described for each interoperability level).

- First level interoperability is supported since HAs running in the same "local" OHS interoperate with other local HAs to exchange data and use other local services.
- Second level is also possible because HAs in our agent-based OHS can interoperate with HAs in remote OHSs based on the same architecture.
- Finally, third level interoperability is supported since the mechanism exists (i.e. KQML, use of different "content" languages, the concept of ontology) to support interoperability between OHSs based on different architectures.

Note that mainly for efficiency reasons, second and third level interoperability will usually happen through facilitators. However, direct interoperation is equally possible. This may happen when the final receiver of a message can be easily identified.

Personal Digital Libraries

At the beginning of this chapter, we illustrated a conceptual architecture for HDLs (Figure 6.1). In this architecture several users can access the local HDL and through the local HDL remote HDLs. In other words, many users can have concurrent access to the tools, data and meta-data of a HDL.

This architecture can be now considered at a lower level of granularity to address the issue of how Personal HDLs (PHDLs) can be developed. In this enhanced framework (Figure 6.17), a single user "owns" and has access to a particular set of tools, data and meta-data. Within this framework, the development of larger HDLs is possible through the aggregation of several PHDLs. Note, that communication between PHDLs can happen directly, but also PHDLs which belong to the same HDL, may use a common facilitator to forward messages in remote PHDLs.

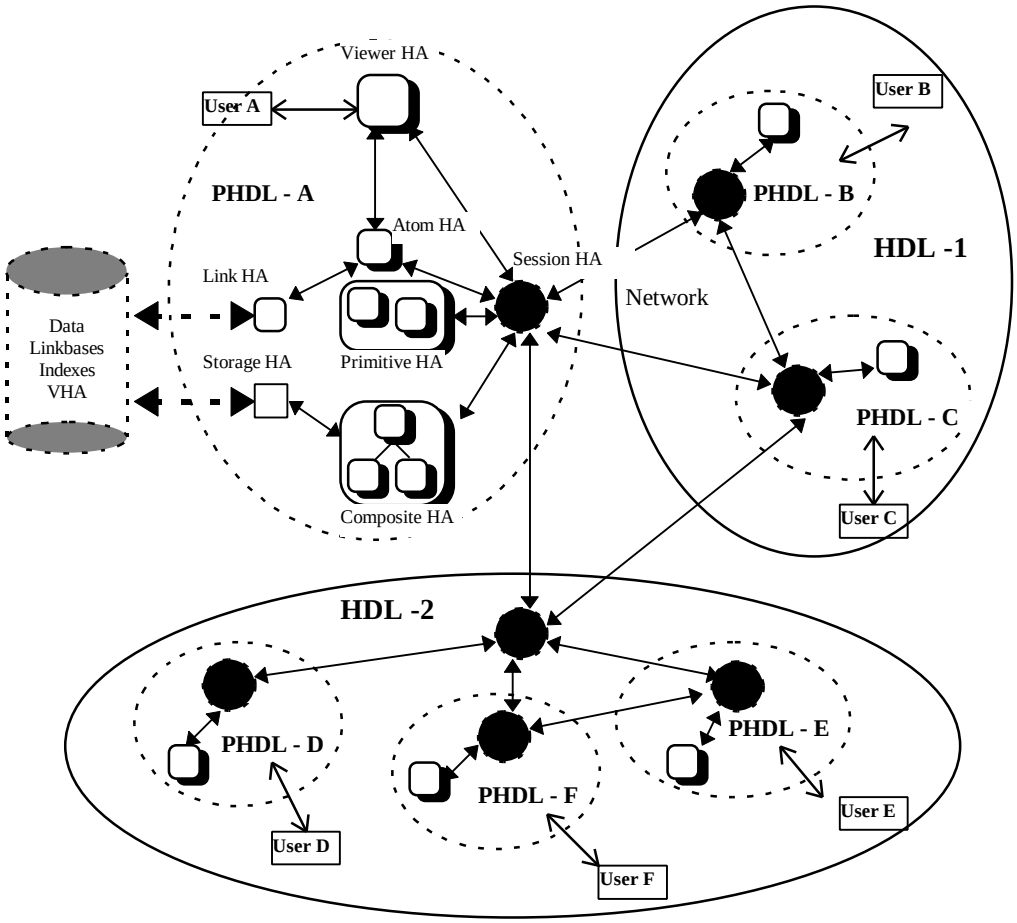


Figure 6.17: Conceptual framework for developing PHDLs.

6.9 Conclusions

An agent-based OHS architecture has been presented in this chapter. The novelty of the architecture can be summarised in the following points.

? Software agents and concepts from CKBS and MAS are used for the first time to design an OHS architecture. Undoubtedly, the problem of developing open hypermedia systems is, in a broader context, a problem of engineering open systems and applications. Agent-based software engineering is a software engineering method which has as the basic aim the development of open systems and applications. Quite surprisingly, although the problem of developing OHSs to a large extent is a problem of developing open systems, no OHS research effort has previously used software agents as the basis to design an OHS and hypermedia digital libraries.

? In this work, for the first time an agent-based interpretation of the widely used Dexter model is utilised to design an OHS. Dexter's data model is also slightly changed to make clear the organisational semantics of the composites in the original Dexter definition.

? The use of KQML is thoroughly examined for the first time, as the language which can serve interoperation in OHS and HDL application domains. KQML is additionally extended with three new performatives and a new parameter, to address needs with are not sufficiently addressed by the current "standard" KQML definition. As it was discussed, the use of KQML also delivers substantial benefits and advantages for OHS interoperation using existing protocols such as OHP.

? A new protocol, called HACL, is introduced for OHS interoperation. This protocol can be used as the "content language" in KQML messages and it can address interoperability in a more complete and efficient way than the OHP protocol. The new protocol in conjunction with the use of KQML as the hypermedia agent communication language addresses most of the weak point and the criticism which has been made about the OHP protocol (see Chapter 3).

? An entirely new approach which views "documents as agents" is introduced. VHAs in effect "represent" documents and together with the instantiation process which is introduced in our architecture, allows documents to be semantically enriched and viewed not simply as data, but as data which can be appropriately enriched to assist information management and information seeking activities.

? It will be shown in the next chapter, that our agent-based OHS architecture addresses the issue of interoperability at all three levels. This is an important goal of any architecture which aims to be used as an underlying platform for developing hypermedia digital libraries.

From a more qualitative perspective, another contribution of the architecture which is presented in this chapter, is that it purposefully regards and treats protocols as superior to architectures. In reviewing OHS research in Chapter 3, it has been identified that until now OHS research considers architectures superior to protocols. Most OHS research efforts try to differentiate by producing an architecture, a link service, a data model which differs in some points to other OHSs. In all these OHS research efforts proprietary protocols are internally used, and generally less emphasis is given to commonly defined protocols which can be used for different interoperability needs.

Our OHS architecture illustrates an approach which is fundamentally different from the OHS research efforts which concentrate on viewer integration, data models etc. In our agent-based architecture a rich and generalised OHS protocol is defined which can address the needs of OHSs and HDLs application areas. Also, both the communication language and the protocol can be extended if new needs must be addressed.

To give an example of what it really means to regard protocols as being superior to architectures, consider the following example. One issue which is a subject for debate in OHS research community, is whether anchors should be kept with the linkbase, or separate from the linking information. In an approach where protocol is superior to architecture, both approaches can coexist. In fact, there is not any good reason which can preclude these two different architectural decisions from coexisting within a single heterogeneous environment. If a protocol exists which can facilitate both approaches and, participating programs in an OHS "know" how to behave in each case (using the ontology parameter), then both approaches for storing anchors can be supported.

Chapter 7

A Prototype OHS System and HDL Application

This chapter discusses a prototype OHS and HDL application based on the agent-based architecture presented in the last chapter. The primary aim in developing the prototype OHS was to conduct an initial evaluation of the OHS architecture by investigating if the concepts and ideas presented in the last chapter are workable in practice. The prototype OHS and HDL application additionally served as the vehicle to conduct several interoperability experiments which are also reported in this chapter. These experiments aimed to demonstrate, examine and clarify the use of the Hypermedia Agent Protocol (HAP) in our OHS architecture. An attempt was made to demonstrate that the HAP can address the interoperability problem at different levels. The final aim in developing the prototypes was driven by the discussion made in Chapter 3 (section 3.7). This aim was to examine the proposed OHS architecture from an information seeking perspective.

7.1 NIKOS, a Prototype Agent-Based OHS

A prototype OHS, called NIKOS, was developed based on the agent-based OHS architecture described in the last chapter. The development of NIKOS was primarily driven by the need to experiment with the proposed OHS architecture and, particularly to answer the following questions:

1. is the idea of using Hypermedia Agents (HAs) and Virtual Hypermedia Agents (VHAs) workable in practice?
2. can we actually extend the prototype system by adding new HAs?
3. can the defined Hypermedia Agent Protocol (HAP) adequately facilitate communication and satisfy the interoperability needs of an OHS-based HDL?
4. will the prototype system operate efficiently and effectively as an information seeking environment?

The NIKOS OHS is composed of different programs which were engineered as HAs. In its simpler form the NIKOS OHS can be used as a personal open hypermedia information management and seeking environment. A user may interact with multiple HAs in order to manage, view and seek his/her personal information resources (i.e. data and meta-data). The "personal" HAs will usually collocated in a single machine. However, some of the HAs which do not directly interact with the user, may be equally distributed in remote machines. A NIKOS OHS which is used in the way described above, should be conceived logically as a Personal Hypermedia Digital Library (PHDL).

A more elaborate use happens when a NIKOS PHDL interoperates with other distributed NIKOS PHDLs. This will usually happen so a NIKOS user can access data which reside in remote NIKOS PHDLs, or, to access services not available by any HA in the "local" PHDL. Finally, as it will be shown later, a NIKOS PHDL may equally access remote hypermedia systems, if these can communicate using the Hypermedia Agent Protocol (HAP).

7.2 An HDL Application based on NIKOS

The need to experiment more realistically with NIKOS, was the motivation to develop an HDL application comprising several NIKOS PHDLs. The prototype HDL application was developed over the CACM collection which was presented earlier in Chapter 5. The CACM HDL application is composed of eight autonomous NIKOS PHDLs, each operating on one

CACM sub-collection. Sub-collections were extracted using the hierarchical clustering method discussed in Chapter 5. Figure 7.1 shows an overview of the CACM prototype HDL.

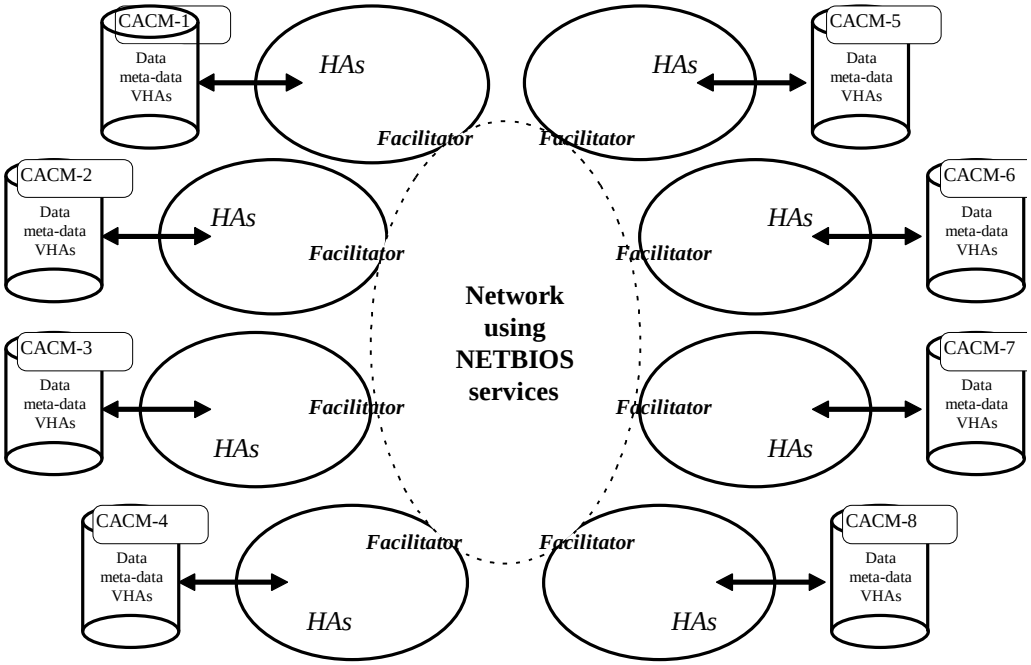


Figure 7.1: The CACM prototype HDL comprising eight autonomous, interoperating NIKOS PHDLs.

The CACM collection was chosen for the prototype application because it meets two basic requirements. First, it is reasonably large and richly interlinked, so it was possible to perform realistic experiments with the prototypes. Second, the CACM collection comes together with information retrieval tasks and associated relevance assessments suitable for evaluation. In fact, the CACM HDL application described here, was eventually used to conduct a user-centered evaluation. This experiment is reported in the next chapter.

The CACM HDL was realised by distributing eight NIKOS PHDLs over a local area network (LAN). In other words, NIKOS PHDLs are located in different Windows 95/NT machines connected through a LAN network. It is crucial to make clear, that HAs in different machines do not communicate and interoperate through the file system of the network server, but they solely communicate through message exchange using the NETBIOS services, an asynchronous communication protocol. This scheme, in effect, implements the message transport model which was outlined in section 6.6. Of course, NIKOS PHDLs could be equally distributed over the Internet, if message exchange was implemented using the protocol

of the Internet (i.e. TCP/IP). However, this is an implementation detail. The use of NETBIOS was dictated by the limitations in time and software resources. Nevertheless, it does not affect, in any aspect, the experiments using the prototype HDL which are reported in this chapter.

7.3 Implementation of HAs and VHAs

7.3.1 Hypermedia Agents

NIKOS' HAs were developed as Windows 95/NT programs. The HAs developed had the key decisions about their actions hardwired into their implementations, and in that respect should be characterised as unintelligent. In developing HAs for NIKOS the emphasis was on efficiency and reliability. However, it must be said here, that there is nothing in our agent-based OHS architecture that precludes "intelligent" HAs being developed.

Besides the usual issues which are involved in the development of any software, there are some additional issues which have to be addressed before the development of HAs. More precisely, there are some key decisions which must be taken, before a HA can be implemented.

1. Identify other HAs that the HA under implementation should register. This decision will determine how other HAs can discover and make use of the services that the HA provides. All the HAs in NIKOS register at least in the HA serving as the facilitator (i.e. the session HA).

2. Define the services the HA under implementation is going to advertise. For each service that it is advertised, a NIKOS HA should provide at least one internal function to handle corresponding requests. The handler functions must satisfy the requests in accordance to some generally agreed specifications.

3. Identify the messages the HA will accept and process. This decision is determined by the capabilities of the HA.

4. Specify how the HA is going to produce and send messages in response to different events. Events will be usually generated by the information seeker (e.g. activation of a link).

For the purposes of our prototype, several HAs (session, text viewer, library, composite, primitive, text atom, storage) were initially developed to deliver a minimum functionality in NIKOS. Table 7.1 summarises these HAs in terms of the key decisions outlined above.

HA	May send to	May receive from	Messages that it may send*	Messages it may receive**	Service(s) it provides
Session	All	All	All	All	facilitator, manages information about current session
Text Viewer	All	All	ask-one, ask-all, stream-all, get, set, achieve, evaluate	achieve, tell	viewer, capable to handle HTML, ASCII and RTF files
Atom	All	All	ask-one, ask-all, stream-all, get, set, achieve, evaluate	achieve, tell	instantiates atom VHAs
Primitive	All	All	ask-one, ask-all, stream-all, get, set, achieve, evaluate	achieve, tell	instantiates primitive VHAs
Composite	All	All	ask-one, ask-all, stream-all, get, set, achieve, evaluate	achieve, tell	instantiates composite VHAs
Library	All	All	ask-one, ask-all, stream-all, get, set, achieve, evaluate	achieve, tell	instantiates library VHAs
Link	All	All	register, advertise, tell, insert, delete,	ask-one, ask-all, stream-all	link services
Storage	All	All	tell	get, set	storage services

* All the HAs may send the messages register, unregister, advertise, unadvertise, error, sorry, broadcast, forward

** All the HAs may receive the messages register, unregister, advertise, unadvertise, error, sorry, broadcast, forward

Table 7.1: The basic characteristics of some HAs in NIKOS OHS.

Although NIKOS' HAs were implemented using Visual Basic, an object oriented design methodology was utilised to identify attributes and methods which can be reused in the development of different HAs. For example, some of the data structures (e.g. the message list) and methods (e.g. process message list) which are part of the implementation of the agent head, are common in all HAs. These components have been implemented as separate modules and eventually reused in the development of all HAs. Figure 7.2 illustrates an example of how functionality can be reused in different HAs (Figure 7.2 is based on the Coad and Yourdon, 1991 notation for object oriented analysis & design).

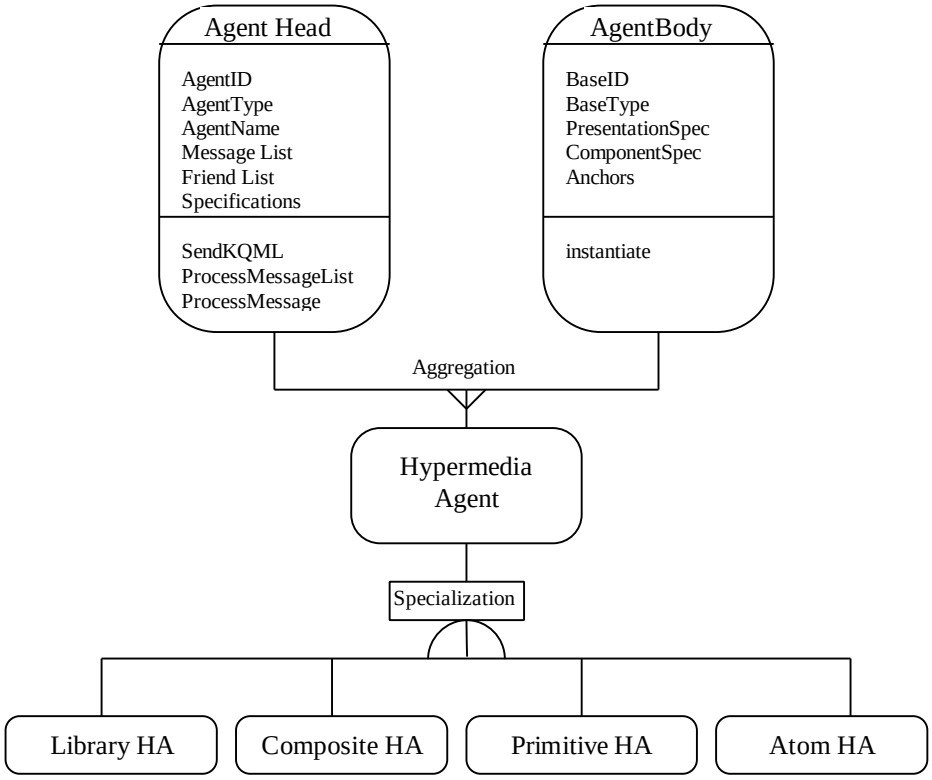


Figure 7.2: Object oriented design showing how functionality may be inherited and reused from different HAs.

7.3.2 Virtual Hypermedia Agents

The CACM collection originally comes as a set of ASCII files and a set of links between them (i.e. linkbase). Thus, each NIKOS PHDL in the prototype CACM HDL application initially composed of a set of interlinked ASCII files. For each NIKOS PHDL some ASCII files were randomly selected and converted to HTML format. This conversion was made to prepare a heterogeneous environments in terms of forms of text data. Besides the preparation of data files and linkbases, some simple automatic methods were additionally used to construct VHAs. In this section we briefly describe exactly how VHAs were automatically created.

The first step to create VHAs was to cluster documents in each sub-collection using the hierarchical clustering method mentioned earlier in Chapter 5. The result of applying a hierarchical clustering process to a set of documents is illustrated in Figure 7.3. As a result of the clustering process it was possible to automatically identify primitives VHAs (the leaf clusters) and composite VHAs (all the clusters in the cluster hierarchy which are not leaves). Finally, the root cluster was used to create the single library VHA for each PHDL.

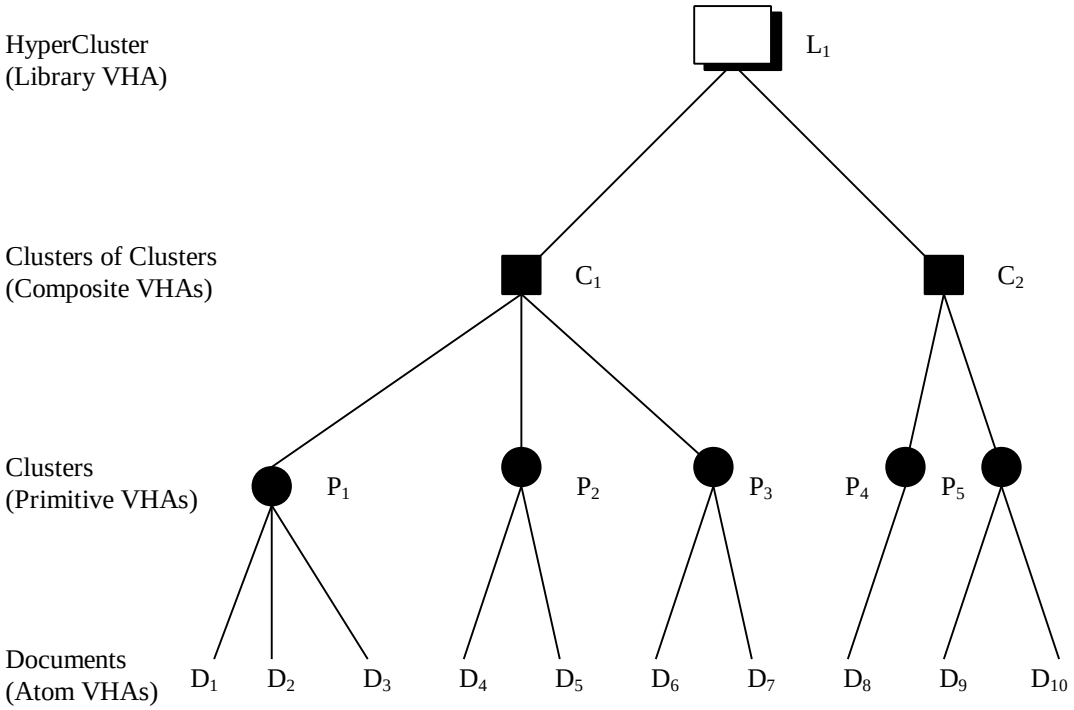


Figure 7.3: The result of a hierarchical clustering process in a flat collection of documents.

The organisation shown above can be applied to any flat collection of information objects, if a hierarchical clustering method is used. The result of this process is a well organised collection in which different information seeking strategies may be utilised (e.g. clustered browsing). Table 7.2 summarises the basic statistics of the eight CACM collections in terms of the documents, links, primitives, composites created in each library. Figures 7.4 and 7.5 illustrate an atom and a primitive VHA file which were fully automatically produced using the process described above.

PHDL	Documents	Atom VHAs	Primitive VHAs	Composite VHAs	Links	Links per document
CACM-1	284	284	49	10	3102	10.9
CACM-2	391	391	61	13	7603	19.4
CACM-3	472	472	87	18	8959	18.9
CACM-4	177	177	37	8	4116	23.2
CACM-5	128	128	25	5	1341	10.4
CACM-6	55	55	13	3	1683	30.6
CACM-7	158	158	32	7	3417	21.6
CACM-8	86	86	16	3	1.178	13.7

Table 7.2: Basic statistics of the eight NIKOS PHDLs.

```
<AGENT_HEAD>
      <AID>\\CACM-1\cacmlib\machine1\F_1_0245.tat</AID>
      <TYPE> TEXT </TYPE>
      <NAME>Polynomial Transformer (Algorithm 29) </NAME>
      <HEAD_SPEC_LIST>
            <AUTHOR> MIKE </AUTHOR>
            <PROTOCOL> NETBIOS </PROTOCOL>
      </HEAD_SPEC_LIST>
</AGENT_HEAD>

<AGENT_BASE>
      <BASEID>\cacmlib\machine1\F_1_0245.htm</BASEID>
      <BASE_TYPE> HTML </BASE_TYPE>

      <COMPONENT_SPECS>
            <SPEC><LIB>CACM_1.dxa</LIB></SPEC>
            <SPEC><CLUSTER>C1_0125.pri</CLUSTER></SPEC>
             <SPEC><KEYWORDS> herhon set polynomial algorithm
                  </KEYWORDS></SPEC>
      </COMPONENT_SPECS>
</AGENT_BASE>
```

Figure 7.4: An atom VHA file.

```
<AGENT_HEAD>
        <AID>\cacmlib\machine1\C1_0027.pri</AID>
        <TYPE> PRIMITIVE </TYPE>
        <NAME>C1_0027</NAME>
</AGENT_HEAD>

<AGENT_BASE>
        <COMPONENT_SPECS>
        <SPEC><LIB>CACM_1.dxa</LIB></SPEC>
         <SPEC><COMPOSITE>1_107.cmp</COMPOSITE></SPEC>
           <SPEC><KEYWORDS> sieve prime upper generate  </KEYWORDS></SPEC>
         </COMPONENT_SPECS>
         <ATOM_LIST>
                    <ATOM><CID>F_1_0377.tat</CID><CN>SIEVE (Algorithm 35)</CN>
                    </ATOM>
                    <ATOM><CID>F_1_2927.tat</CID><CN>Some New Upper Bounds on
                                the Generation of prime Numbers </CN>
                    </ATOM>
        </ATOM_LIST>
</AGENT_BASE>
```

Figure 7.5: A primitive VHA file.

7.4 Working with NIKOS

From a user-centered perspective, an information seeker initiates his/her NIKOS PHDL by executing the session HA. The default behaviour of the session HA is to locate other HAs in the PHDL system. The user selects the HAs which wishes to activate or, alternatively can instruct the session HA to activate all HAs available in a default directory.

HAs as they are activated, register their presence to the session HA and advertise their capabilities. Figure 7.6 shows examples of messages produced during this initialisation process. This figure shows the *register* and *advertise* messages that were sent by a text viewer HA which is capable of presenting ASCII, HTML and RTF files and, by a primitive HA which is capable of instantiating primitive VHAs.

```
( register      :sender  \\CACM-1\TEXT_VIEWER_NAME
                :receiver \\CACM-1\SESSION_NAME
                :reply-with  tvid0
                :language HACL
                :ontology DEXTER
                :content  Profile ( :!name \\CACM-1\TEXT_VIEWER_NAME  )
                                  :!Type VIEWER
)
( advertise     :sender  \\CACM-1\TEXT_VIEWER_NAME
                :receiver \\CACM-1\SESSION_NAME
                :reply-with  tvid1
                :language HACL
                :ontology DEXTER
                :content ( achieve  :language HACL
                                    :ontology DEXTER
                                    : InstantiateData(:!ID ::Type HTML ASCII
                                                           ::FileContent)
                         )
)


( register      :sender  \\CACM-1\PRIMITIVE _NAME
                :receiver \\CACM-1\SESSION_NAME
                :reply-with  pid0
                :language HACL
                :ontology DEXTER
                :content  Profile ( :!name \\CACM-1\PRIMITIVE_NAME  )
                                  :!Type PRIMITIVE
)
( advertise     :sender  \\CACM-1\PRIMITIVE_NAME
                :receiver \\CACM-1\SESSION_NAME
                :reply-with  pid1
                :language HACL
                :ontology DEXTER
                :content ( achieve    :language HACL
                                      :ontology DEXTER
                                      : InstantiateVHA( :!ID ::Type ::FileContent  )
                         )
)
```

Figure 7.6: Four messages showing how a text viewer and a primitive HA registering and advertising to their local facilitator (session HA).

In parallel to the registering and advertising process shown in Figure 7.6, the session HA of the “local” NIKOS PHDL may notify its presence to other PHDLs. To achieve this the session HA will send messages to register with other facilitators in remote PHDLs. It will also forward the advertise messages received from "local" HAs to remote facilitators. Using this process remote PHDLs can be informed about the presence and the capabilities of other libraries. Figure 7.7 shows an example of register and advertise messages sent by the session HA in the PHDL CACM-1 in order to:

- register with the session HA in CACM-2, that means to notify its presence. Note that the *register* message informs CACM-2 (using the ‘keywords’ parameter) about the contents of CACM-1;
- forward a message from the local storage agent to the session HA in CACM-2. Note that the *advertise* message notifies HAs in CACM-2 about the capability of the storage HA in CACM-1 to access files using the ftp protocol.

```
( register      :sender  \\CACM-1\SESSION_NAME
                :receiver \\CACM-2\SESSION_NAME
                :reply-with  sid0
                :language HACL
                :ontology DEXTER
                :content  Profile ( :!name \\CACM-1\SESSION_NAME
                        ::type SESSION
                        ::keywords “language programming computer” )
)
( advertise     :sender  \\CACM-1\SESSION_NAME
                :receiver \\CACM-2\SESSION_NAME
                :reply-with  sid1
                :language HACL
                :ontology DEXTER
                :content ( get          :language HACL
                                        :ontology DEXTER
                                        :File( :!address  ::protocol ftp )
                        )
)
```

Figure 7.7: Two messages illustrating how a session HA registers with and advertises capabilities to another remote session HA.

At the end of the registering and advertising processes[24] described above, all the HAs must be properly registered and advertised. Note that this process takes place in the background without the involvement of users. HAs which are finally activated will appear in the user's desktop (Figure 7.8) ready to be used by the information seeker.

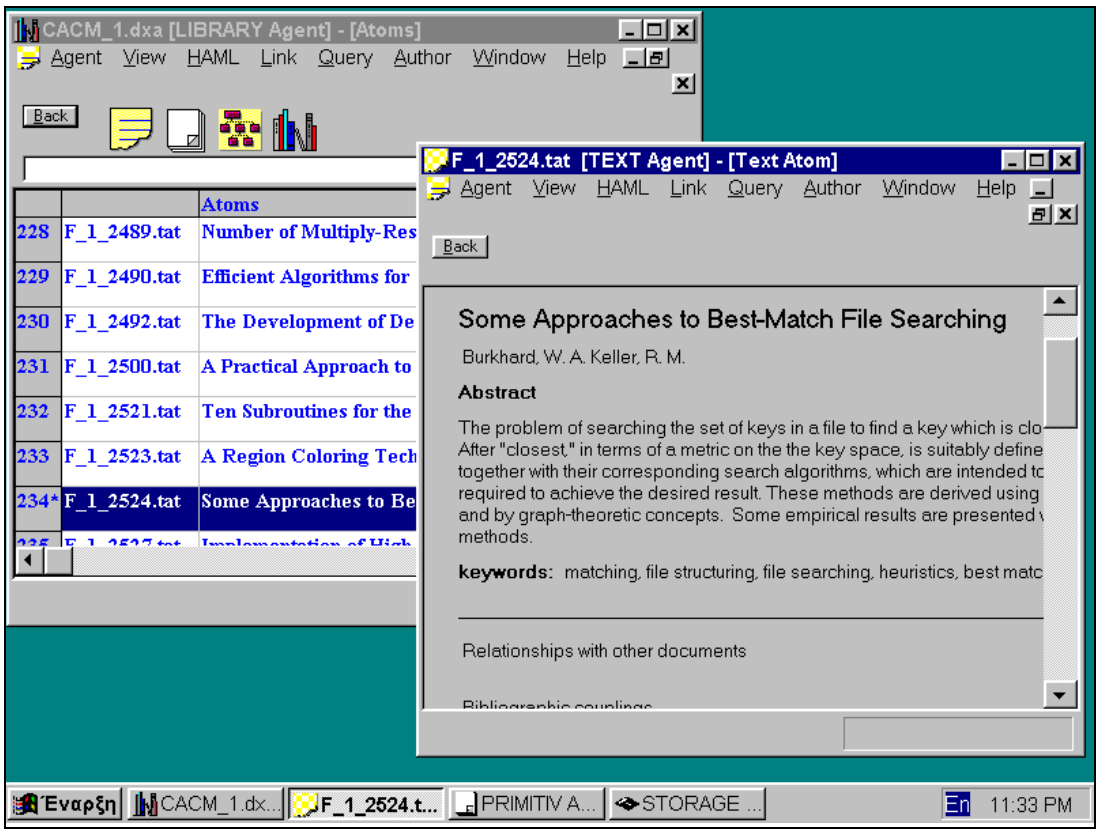


Figure 7.8: A snapshot of a NIKOS OHS system with several HAs running.

[24] in a real HDL application which may comprise tens or hundreds of participating PHDLs the process of registering and advertising to facilitators in remote PHDLs will be applied only occasionally. Normally, facilitators will store locally and recall information about remote HAs. In that case, HAs will have to communicate and inform their local or remote facilitators, only when something changes in their previously advertised specifications.

7.5 Extending NIKOS with New HAs

This section aims to give an answer to the second question presented at the beginning of this chapter. This question emerged after developing the first workable set of HAs, and was whether we could extend the system by adding new HAs which will deliver additional services to the prototype system. Therefore, we tried to add three new HAs to the NIKOS system.

- The Information Retrieval (IR) HA which performs single or parallel searching using indexes residing in one or multiple PHDLs respectively.
- The Collection Fusion Problem (CFP) HA which solves the collection fusion problem using the link-based fusion strategy discussed in Chapter 5. This HA may be used by the information seekers so they can receive suggestions about the most relevant PHDLs to their information needs, but it is also used by the IR HA in distributed parallel searching.
- The History HA which maintains a list of the user's movements during an information seeking process.

The incorporation of the history HA in the NIKOS system was quite straightforward. It required a small change to the HAs in the storage layer. More precisely, HAs which continuously instantiate VHAs had to be changed, so for each instantiation they make, they send a message to the history HA. The history HA receives the message and places the movement in its history list (Figure 7.9). Information seekers can later use this list to activate documents.

Incorporation of the IR agent required similar changes to other HAs which may use the services of the IR HA. The major change was the development of an interface which is used by information seekers to create a query and also select the PHDLs that will be searched (Figure 7.10). Another option for the information seekers is to ask from the CFP agent to suggest PHDLs for searching. Information seekers may fully accept the suggestions or may partially change them, before they initiate a distributed searching. Finally, information seekers have the option to ask the IR agent to make the distributed searching without specifying any PHDL. In that case, the IR HA will cooperate with the CFP HA to ask a solution about the source selection problem, before it will start the actual searching.

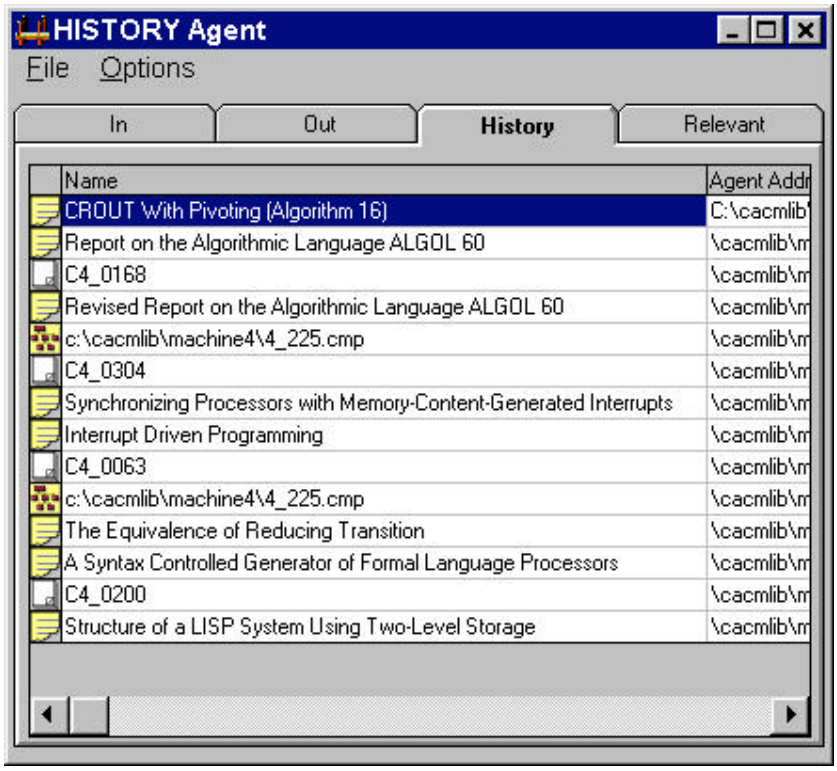


Figure 7.9: The history HA in NIKOS OHS.

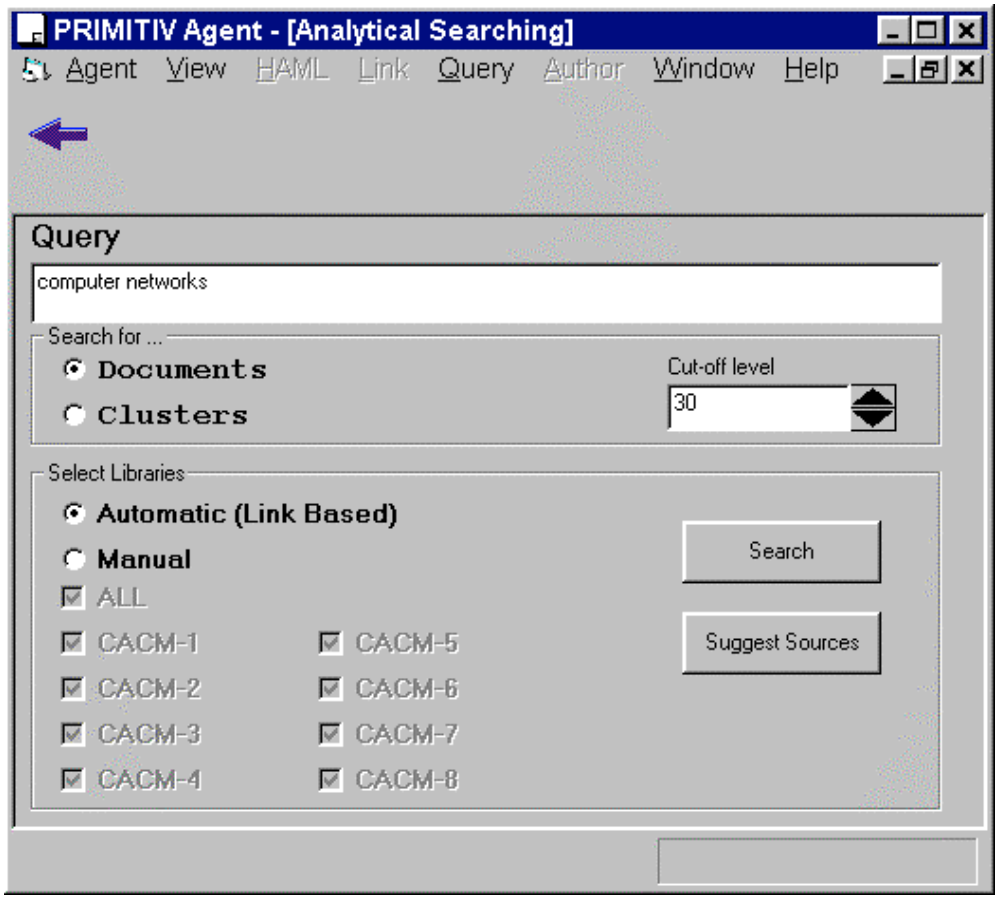


Figure 7.10: The interface added in HAs to use analytical searching strategies.

Generally speaking, the NIKOS system can be extended with new tools and information seeking strategies, if these tools or strategies can be implemented as HAs. Of course, in order to be able to include a wide range of different strategies the architecture and especially the protocol must posses a certain degree of expressiveness.

As it was mentioned in Chapter 6, however, the hypermedia agent communication language can be extended to address the potential needs of new tools, information seeking strategies etc. For instance, in our experiments in extending the first version of the NIKOS system with the IR and CFP agents, we realised that a new KQML performative (i.e. the evaluate performative) is required to address the particular needs of these two HAs.

7.6 Three Interoperability Studies

This section aims to give an answer to the third question presented at the beginning of this chapter. The three cases studies which will be presented in this section, examine in detail how different levels of interoperability are supported by the protocol of the agent-based OHS architecture. The first case study examines interoperability between HAs in a single NIKOS OHS by examining the integration of viewers to the NIKOS OHS. So, this first case study shows how first level interoperability is achieved in NIKOS (Figure 7.11), but it also demonstrates how the viewer integration problem is addressed in our OHS architecture.

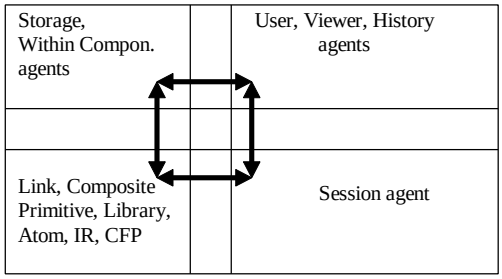


Figure 7.11: First level interoperability between HAs within the same NIKOS OHS.

The second case study examines interoperability between two different distributed NIKOS OHSs (Figure 7.12). The subject of interoperation is querying a remote index and the aim is to illustrate how users in a NIKOS system can access data and services from another remote NIKOS OHS.

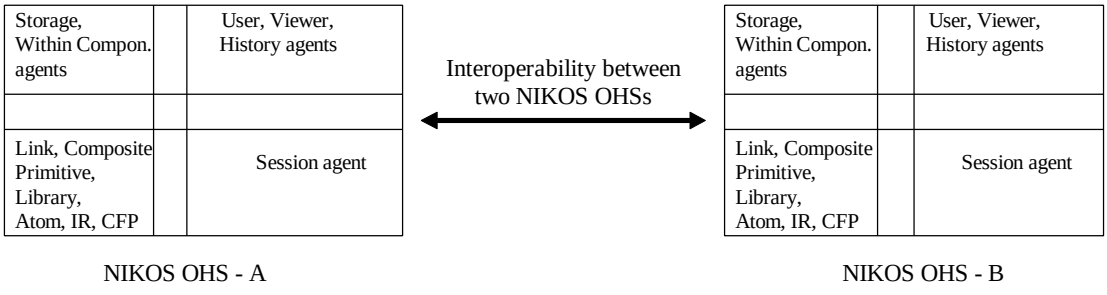


Figure 7.12: Second level interoperability between two distributed NIKOS OHSs.

Finally, the third case study discusses an interoperability experiment between two completely different hypermedia systems. The first system is the NIKOS OHS. The second system is a typical second generation hypermedia system called HyperTree (Salampasis & Tait, 1995). HyperTree was extended so it can communicate using KQML and HACL (Figure 7.13). In this case study the attempt is to demonstrate that the HAP (Hypermedia Agent Protocol) is flexible and sufficient so it can accommodate third-level interoperability between different hypermedia systems.

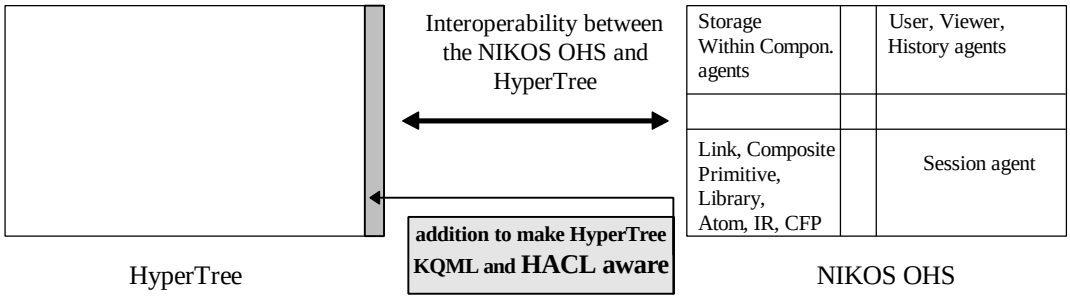


Figure 7.13: Third level interoperability between NIKOS OHSs and HyperTree.

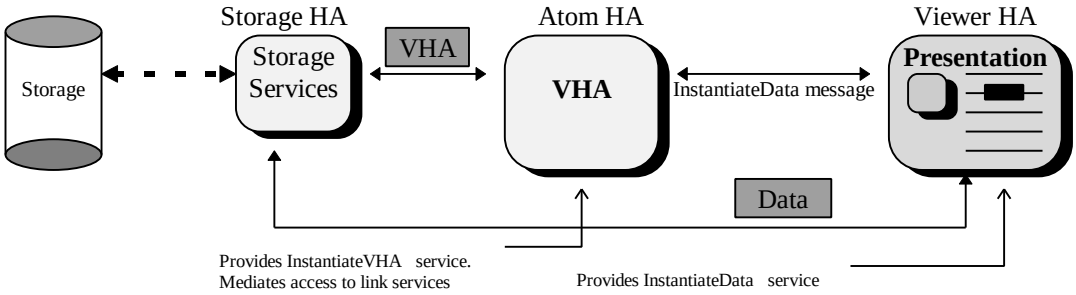
7.6.1 Viewer Integration (First Level Interoperability)

Whitehead (1997) has recently discussed four different integration methods for viewers: launch-only integration, wrapper integration, custom integration and, combination integration which combines two of the first three integration methods.

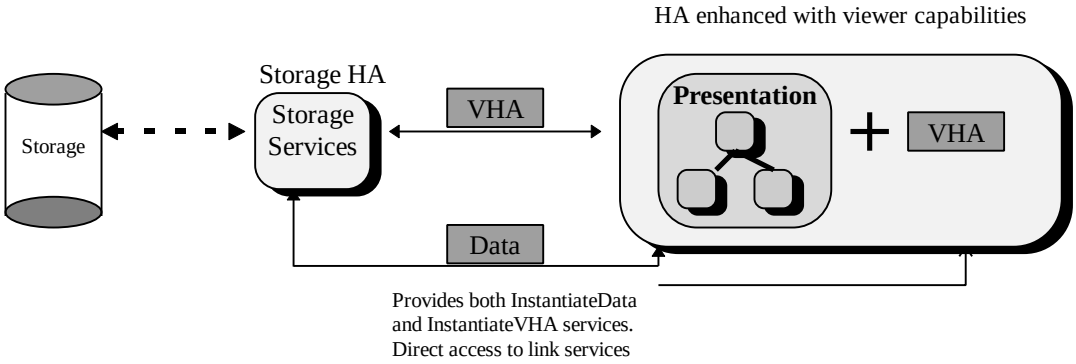
To understand how integration of viewers happens in our OHS architecture, it should be recalled that instantiation of a data object in our architecture, takes place through the instantiation of VHAs (see section 6.3). In other words, HAs in the storage layer of our architecture (e.g. atom HA), are always used as a kind of wrapper to facilitate the integration of viewers with the link HA providing the link services. From that point of view, our agent-based OHS architecture uses a combination integration method, with the HAs at the storage layer playing the role of a wrapper for viewers.

A variation of the integration method which is described in the above paragraph, happens when a viewer HA, is enhanced to provide InstantiateVHA services in addition to its "normal" InstantiateData service. In that case the viewer HA can handle both the VHA and corresponding data and therefore a HA in the storage layer is not required to mediate access to

the link services. The same scheme is inversely realised if HAs in the storage layer are enhanced to provide InstantiateData services in addition to their "normal" InstantiateVHA services. In fact, this method was partially used in NIKOS. More precisely, the primitive, composite and library HAs in NIKOS system were developed so they do not use external viewers to interact with users. Figure 7.14 illustrates the two instantiation approaches mentioned above (with and without using an external viewer).



1) Instantiation with Viewer HA (VHA and data handled by different HAs)



2) Instantiation using only a HA in the storage layer (both VHA and data handled by the storage layer HA, i.e. storage layer HAs play the role of a viewer)

Figure 7.14: Instantiation using 1) both an atom HA and a viewer HA and, 2) using only a (composite) HA enhanced with viewer capabilities.

To explain in detail how the instantiation process takes place, consider the following scenario: an information seeker currently interacts with a primitive HA and examines the members of the cluster. Suppose now, that the information seeker wants to instantiate a particular text file,

for closer examination. The following steps will follow in order to achieve the instantiation of the text file:

1. the primitive HA sends a message to the atom HA requesting to instantiate the VHA file;
2. the atom HA interoperates with the storage HA to get the contents of the VHA file;
3. the atom HA instantiates the atom VHA and interoperates with the viewer HA requesting to instantiate the corresponding data file;
4. the viewer HA requests data from the storage VHA;
5. concurrently with step 4, the viewer HA requests[25] anchors from the atom HA which has just instantiated the corresponding VHA in step 3.

Note that steps 2 and 4 are not necessary, if viewer and atom HAs can directly access data from the storage device without using the storage HA. Figure 7.15 shows the series of messages exchanged during this process.

[25] anchors can be also received from the link server, if the hypermedia model used stores anchors in the link server (e.g. Microcosm).

Step 1: Primitive HA requests atom HA to instantiate the VHA having VHA_Address (event generated by information seeker)

```
( achieve       :sender  \\CACM-1\PRIMITIVE_NAME
                :receiver \\CACM-1\ATOM_NAME
                :reply-with  pid0
                :content  InstantiateVHA ( :!ID VHA_Address ::Type TEXT  )
)
```

Step 2: ATOM_NAME requests storage HA to get the VHA file

```
( get           :sender  \\CACM-1\ATOM_NAME
                :receiver \\CACM-1\STORAGE_NAME
                :reply-with  aid0
                :content  File( :!address VHA_Address  )
)
```

Step 3: ATOM_NAME HA instantiates VHA_Address VHA and requests TEXT_VIEWER_NAME to instantiate the actual text filer

```
( achieve       :sender  \\CACM-1\ATOM_NAME
                :receiver \\CACM-1\TEXT_VIEWER_NAME
                :reply-with  aid1
                :content  InstantiateData ( :!address TEXT_FILE_Address ::Type TEXT  )
)
```

Step 4: text viewer requests storage HA to get the text data file to be presented to the user.

```
( get           :sender  \\CACM-1\TEXT_VIEWER_NAME
                :receiver \\CACM-1\STORAGE_NAME
                :reply-with  tvid0
                :content  File( :!address TEXT_FILE_Address  )
)
```

Step 5: text viewer requests from atom(link) HA all the anchors for the text file to be instantiated

```
( ask-all       :sender  \\CACM-1\TEXT_VIEWER_NAME
                :receiver \\CACM-1\ATOM_NAME
                :reply-with  tvid1
                :content  Anchors( :!UID TEXT_FILE_Address  )
)
```

Figure 7.15: Messages illustrating the process of instantiating a text file.

The type of integration which has been discussed above, is a combination integration which requires the viewer to fully communicate using KQML and HACL. This type of integration is like the integration of fully aware viewers in Microcosm (Hall et al, 1996).

Integration of other viewers which are unable to communicate using KQML and HACL, can take place as launch only integration. In that case the cooperation process outlined in Figure 7.15 ends at step 3. In this step the atom HA instantiates the atom VHA and launches the external application (Figure 7.16). Note, that it may not be possible to activate links from the launch-only viewer (e.g. if links are not embedded in the data as it happens for example in HTML files). However, the atom HA which has instantiated the VHA will contain the anchor list for the information object displayed in the "unaware" viewer. Also the atom HA can still communicate with the link HA and access the link services. This means the atom HA can compensate for features that are missing from the external viewer because of the weak integration.

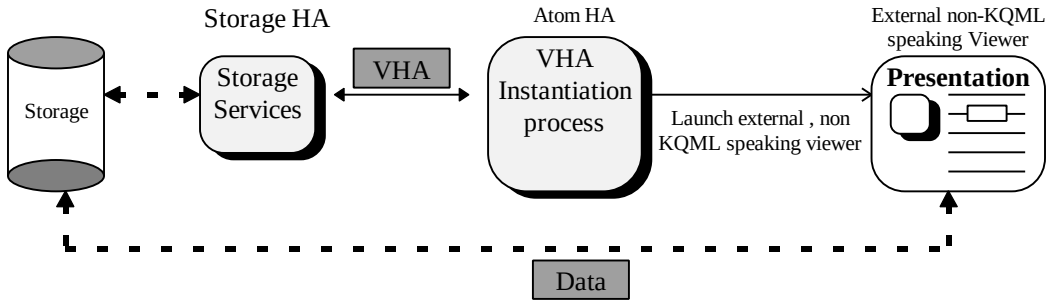


Figure 7.16: Integration of external, non-KQML speaking viewer.

Finally, another type of integration is possible if an external viewer can be customised using a internal language (e.g. Microsoft Word which can be customised using an interpreted version of Visual Basic) (Figure 7.17).

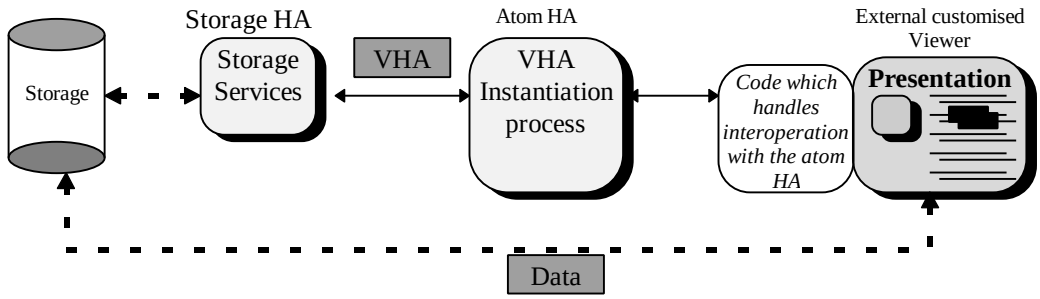


Figure 7.17: Integration of external customised viewer.

7.6.2 Querying a Remote PHDL (Second Level Interoperability)

A higher level of interoperability is required, if services or data from a remote PHDL must be accessed. For example, this type of interoperability is required in the prototype HDL if an information seeker wants to query a remote NIKOS PHDL. Suppose that an information seeker in CACM-1 PHDL currently interacts with a primitive HA. Using the interface that it is illustrated in Figure 7.10, submits a query (i.e. "ALGOL COBOL") and specifies the CACM-2 as the collection against which the query should run.

Upon submission of the query the following steps will take place.

1. The primitive HA recognises that this is a request that should be handled by a remote PHDL. It sends a message to the local facilitator requesting to forward the evaluate message to the PHDL CACM-2.
2. The local facilitator forwards the message to the facilitator in CACM-2.
3. The facilitator in CACM-2 receives the message and identifies the HA in CACM-2 which is able to satisfy the request. The identification of the correct HA is done through the examination of the capabilities that HAs have previously advertised.
4. The IR HA in CACM-2 searches the local collection and produces the results of the query (i.e. file F_2_1206.htm and F_2_3204.htm). Then it returns back the results to its local facilitator.
5. The facilitator sends the results to the facilitator which originally requested the search.
6. The facilitator in CACM-1 returns the results back to the primitive HA which will eventually presented them to the user.

Note that like the examples of interoperability discussed in the last section, steps 2, 3, 5 could be eliminated, if the primitive HA and the DIR HA could exchange messages directly. In that case the development of HAs becomes more complicated, but probably in some cases cooperation could be faster. Figure 7.18 illustrates the messages related to the steps outlined above.

Step 1:

```
( achieve     :sender  \\CACM-1\PRIMITIVE_NAME
              :receiver \\CACM-1\SESSION_NAME
              :reply-with  pid0
              :content  Search( :!Library CACM-2 :!query "ALGOL COBOL" ::cut-off 2  )
)
```

Step 2:

```
( achieve     :sender  \\CACM-1\SESSION_NAME
              :receiver \\CACM-2\SESSION_NAME
              :reply-with  sid0
              :content Search( :!Library CACM-2 :!query "ALGOL COBOL" ::cut-off 2  )
)
```

Step 3:

```
( achieve     :sender  \\CACM-2\SESSION_NAME
              :receiver \\CACM-2\IR_NAME
              :reply-with  sid0
              :content Search( :!Library CACM-2 :!query "ALGOL COBOL" ::cut-off 2  )
)
```

Step 4:

```
( tell        :sender  \\CACM-2\DIR_NAME
              :receiver \\CACM-2\SESSION_NAME
              :in-reply-to  sid0
              :content  Search( ::result "F_2_1206.htm F_2_3204.htm"  )
)
```

Step 5:

```
( tell        :sender  \\CACM-2\SESSION_NAME
              :receiver \\CACM-1\SESSION_NAME
              :in-reply-to  sid0
              :content  Search( ::result "F_2_1206.htm F_2_3024" )
)
```

Step 6:

```
( tell        :sender  \\CACM-1\SESSION_NAME
              :receiver \\CACM-1\PRIMITIVE_NAME
              :in-reply-to  pid0
              :content  Search( ::result "F_2_1206.htm F_2_3024" )
)
```

Figure 7.18: Messages illustrating the process of querying a remote PHDL.

7.6.3 Accessing Other Hypermedia Systems (Third level Interoperability)

Another experiment has been conducted to study the issues and problems in interoperation between different hypermedia systems. The experiment described in this section aimed to study these issues by conducting an interoperability experiment between the NIKOS OHS and the HyperTree hypermedia system.

The HyperTree system is a typical frame-based second generation hypermedia system (Salampasis et al, in press). HyperTree supports text, graphics and sound data which are internally stored in a relational database. There are two different types of nodes in HyperTree: organisational or concept nodes and information nodes. The main purpose of the organisational nodes is to organise the hypermedia information network using hierarchical structure. Information nodes represent the actual multimedia contents (e.g. text, graphics). HyperTree organises and structures the hypermedia database in two different structures: a graph and a hierarchical structure. There are also two different types of links which are stored separate from data. Figure 7.19 shows the data model of the HyperTree system.

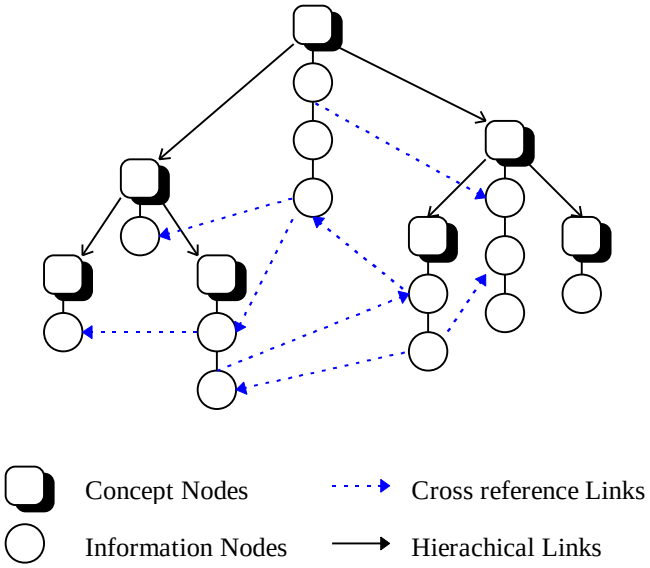


Figure 7.19: The data model of the HyperTree system.

Despite the fact that HyperTree stores links separate from data, it can not be characterised as an OHS. HyperTree tools are tightly bounded within a single application framework.

However, HyperTree it was the only hypermedia system which was available for the experiment and had some quite well defined system interfaces.

The goals of this interoperability experiment was to enable PHDLs based on our agent-based architecture to interoperate with the HyperTree system. Interoperation should include agent-based PHDLs accessing data and links from a HyperTree system while HyperTree users could work with HyperTree at the same time. The actual architectural setting of the experiment is illustrated in Figure 7.20.

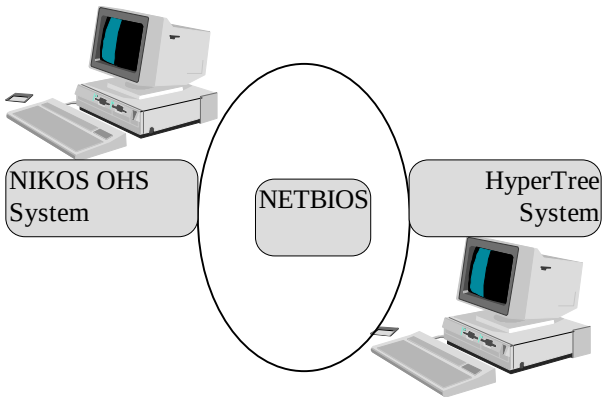


Figure 7.20: Architectural setting for interoperation between NIKOS OHS and HyperTree.

The following issues and problems had to be resolved to establish interoperation.

? *Extensions to HyperTree system to become KQML aware.*

The first change was to extend HyperTree, so it can exchange messages using KQML and HACL. That was made by adding to the existing implementation of HyperTree, a new module which provides the required functionality to enable HyperTree interoperate with other agent-based PHDLs. This new module acts a wrapper and basically implements an agent head over the old HyperTree system (Figure 7.21).

A set of services had to be determined which HyperTree can provide to other NIKOS-based PHDLs. This set of services include access to data in HyperTree (i.e. services normally provided by the storage HA in our architecture) and access to HyperTree's

links database (i.e. services provided by the link HA). HyperTree's wrapper registers and advertises these services to other PHDLs.

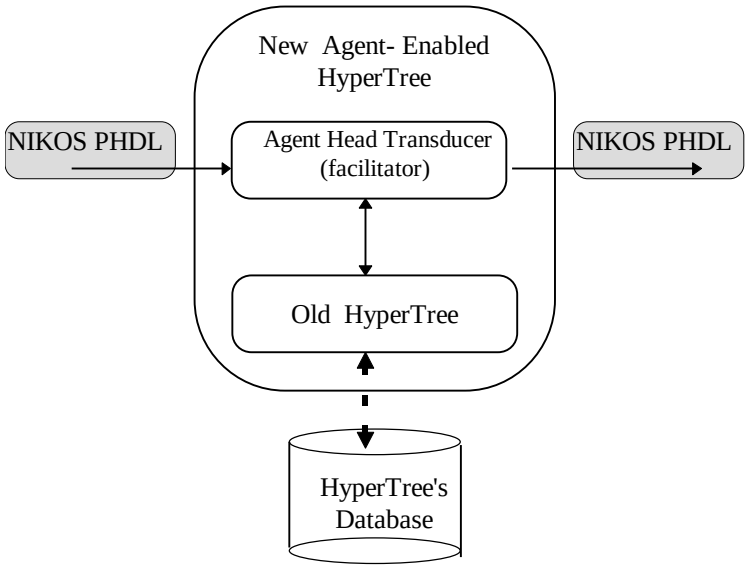
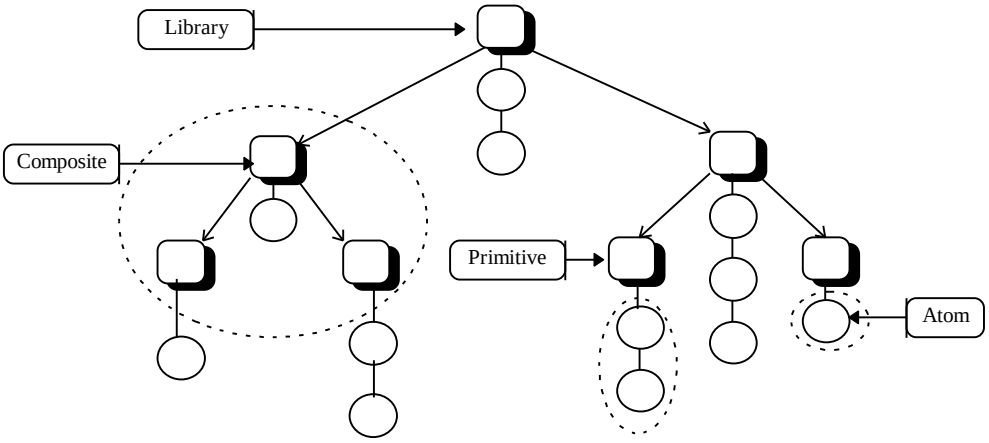


Figure 7.21: HyperTree's architecture after extensions.

? *Issues regarding the data models*

There was a good matching between the data models used in HyperTree and the NIKOS OHS. HyperTree's support for hierarchical structure and its support for organisational nodes in addition to information nodes matches quite good with the NIKOS' data model which implements similar organisational models using atoms, primitives and composites (Figure 7.3). However, some conventions had to be introduced to facilitate communication regarding hypermedia components in HyperTree. Figure 7.22 shows the data model mappings used in storage interoperation between NIKOS and HyperTree.



Arrows show what is passed as a parameter from NIKOS systems. Dashed ellipses indicate what is returned back from HyperTree

Data model transformations

Requests from NIKOS PHDLs	HyperTree's reply
1. request for atom VHA (code of information node is the parameter)	?? returns the details of a single information node
2. request for primitive VHA (code of concept node is the parameter)	?? returns the list of information nodes associated with a single concept node
3. request for composite VHA (code of concept node is the parameter)	?? returns the list of Information Nodes associated with a single concept node ?? returns the concept nodes in the upper and lower level.
4. request for library VHA (code of concept node is the parameter)	?? returns all the concept nodes in lower level up to the leafs (composites) ?? Returns the leaf concepts (primitives) ?? for each concept node in the above lists return the associated information nodes (atoms)

Figure 7.22: Mappings between NIKOS' and HyperTree's data models.

? *Problems with missing features.*

Another type of problems arise from the fact that HyperTree does not support some features; for example VHAs. The wrapper has to compensate for these missing features in order to enable HyperTree interoperate. Suppose for example that HyperTree receives a get message which requires a composite VHA to be returned back to the sender. The wrapper must access at the run-time all the necessary information and using the transformations depicted in Figure 7.22, to construct on-the-fly the VHA file and, eventually return it to the original sender of the message.

7.7 NIKOS as an Information Seeking Environment

The defining feature of the NIKOS OHS as an information seeking environment is that it emphasises and advocates the mixed use of different information seeking strategies. Information seekers in NIKOS can interact in a *parallel, coordinated fashion* with multiple HAs. This interaction model is possible because each HA in NIKOS system implements, in effect, a different information seeking strategy:

1. simple across document browsing supported by atom and viewer HAs;
2. clustered browsing; this type of browsing is supported by primitive HAs which do not display raw data, but display clusters of raw information objects;
3. hierarchical browsing; hierarchical browsing is supported by composite HAs which display hierarchies of other composites and clusters;
4. browsable table of contents (library HAs);
5. single index searching (IR HAs);
6. multi-collection distributed searching (IR and CFP HAs);
7. clustered searching (IR HAs).

The HAs which represent the different strategies can cooperate by exchanging messages. Hence, they support the information seeker in moving from one tool (i.e. a HA) which implements a particular search technique, to another tool which supports a different strategy. Of course, the information seeker must have control of this process and should be able to coordinate HAs. Thus, *to seek information in NIKOS is to coordinate and manage HAs*.

From the users point of view, NIKOS as an information seeking environment is seen as a set of autonomous cooperating tools which are available to assist searching (Figure 7.23). Each tool can assist information seekers in a different way during the information seeking process. The framework for information seeking discussed in section 4.1 (see Figure 4.1), shows how multiple HAs may be used to support different tasks during an information seeking process. For example, the CFP HA assists users in the source selection process and the IR HA in the execution of queries. Other HAs in the storage and the run-time layer assist information seekers in examining search results using different views of the information space under searching (e.g. simple networked, clustered, hierarchical).

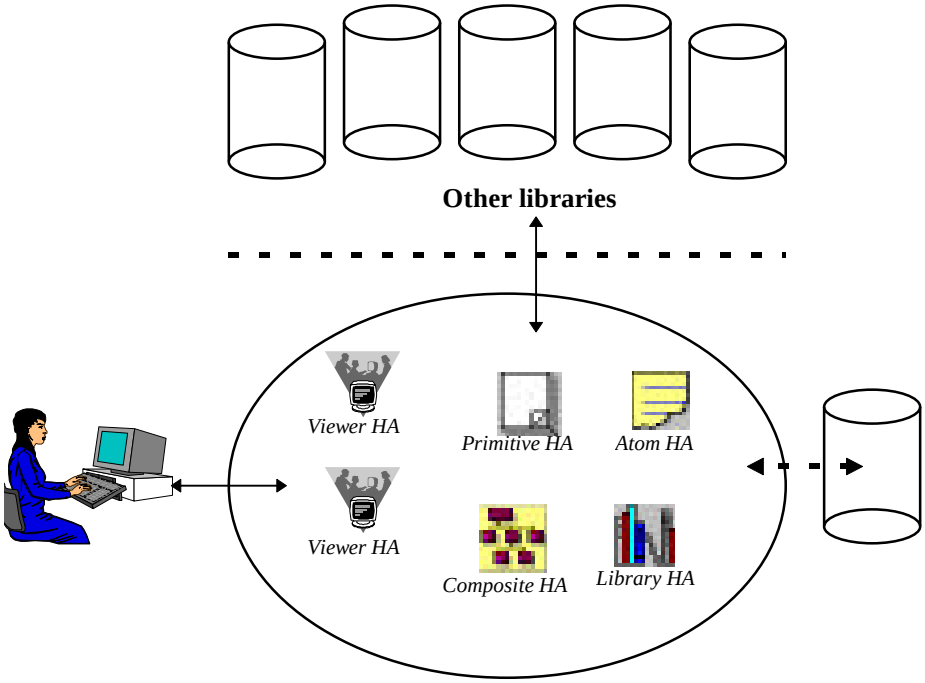


Figure 7.23: Overview of NIKOS as an information seeking environment.

Figure 7.24 illustrates the actual manifestation of the sketched view shown in Figure 7.23. It presents a snapshot of an information seeking process with several HAs activated each presenting a different view of the information space under searching. In this figure, different HAs are active and offer to the user the opportunity to inspect different, but possibly related, views of the information space. At the bottom of the screen the information seeker can view other HAs which are available (e.g. a library HA and a composite HA) and, which s/he can activate if their assistance is required.

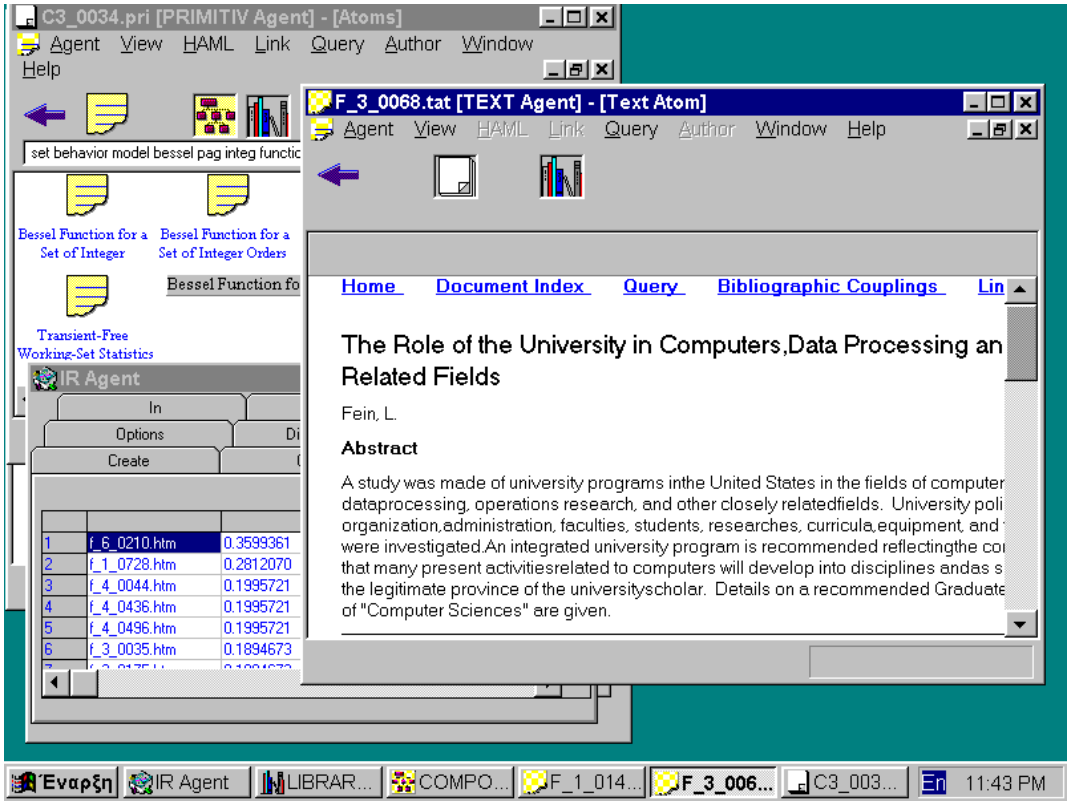


Figure 7.24: A screen snapshot showing the parallel use of several HAs to search information in NIKOS.

To compare our prototype CACM HDL based on the NIKOS system, an analogous HDL was built using the WWW. The WWW HDL was enhanced with local and distributed searching capabilities similar to those provided by the IR HA in NIKOS system. However, the information seeking environment which is available in NIKOS is substantially different from the one provided by the WWW version of the same library.

In the WWW both browsing and analytical strategies must be accessed through the same interface, i.e. a single WWW client. Of course, multiple clients can be opened but they can not be synchronised in the same way that *HAs can be synchronised* by exchanging messages. For example, the atom and the primitive HAs can be synchronised so instantiation of a text file additionally induces the instantiation of the corresponding cluster in which this text document belongs. This may happen if the atom HA is programmed so after the instantiation of an atom VHA, sends a message to the primitive HA requesting the instantiation of the corresponding cluster. This type of synchronisation is not possible in the WWW, because its support for interoperability is limited to client/server.

Additionally, in NIKOS the process of information seeking is not tightly "pre-engineered" into the system as it is in the WWW. Consider for example a typical information seeker using the WWW. The user will click on a link and s/he will be presented with a new search artefact that should be examined in isolation. Based on the examination of the search results (i.e. a WWW page), s/he will click on another link to produce another search artefact that must be examined in isolation and so on. Hendry and Harper (1996) characterise such information seeking environments as "over-determined" and Vickery and Vickery (1993) as "over-engineered". Usually "over-determined" information seeking environments are inflexible and ineffective, especially for opportunistic information seekers.

In NIKOS different views of the information space can coexist and examined in parallel. In NIKOS the information seeking environment is not completely pre-engineered since it can be customised by activating or deactivating different HAs. If it is also considered that HAs can be developed in different ways (if they abide by the rules of the agent-based OHS architecture) the user interfaces used for searching can be also altered or customised. This is another useful option for the adaptation of the information seeking environment.

For managing the HAs and coordinating the information seeking process, information seekers must be aware of the services and capabilities of the available HAs. Having this basic knowledge they can initiate the information seeking process by activating only the HAs which they believe are more suitable to produce the best results. This selective utilisation offers to information seekers the opportunity to customise their information seeking environment and adapt it for the information problem at hand. Of course, during the information seeking process, as the information problem changes or is better defined, or as the users become more familiar with the environment, they have the opportunity to activate more HAs or deactivate others which are not useful anymore.

7.8 Discussion

In the last six sections an attempt was made to give some answers to the four questions that have been presented at the beginning of this chapter.

The development of the prototype OHS system and HDL application, together with the experiments and studies that have been earlier presented, demonstrates that the idea of HAs and VHAs is workable in practice (sections 7.2 to 7.4). It is relatively easy to develop a HA given that much of the functionality is similar between HAs and, once implemented it can be reused in the development of different HAs. Additionally communication between HAs does

not append any unacceptable overhead in the operation of the system. Especially, if direct communication between HAs is permitted without intervention of the facilitator, the communication overhead is minimal and almost unnoticeable. It was also shown, that the development of VHAs can happen completely automatically using tools which can impose hierarchical and clustered organisations to any flat collection of information objects.

The second question which we tried to investigate and give an answer if possible, was the issue of extending the NIKOS system with new HAs (section 7.5). Generally, it was quite easy and straightforward to add new HAs to the NIKOS OHS system. From a software-engineering point of view, extension required small changes to the implementation of existing HAs or the addition of interfaces that can provide access to the new HAs. On the other hand, the Hypermedia Agent Protocol (HAP) was proved to be expressive and flexible enough so it could satisfy new communication needs which emerged as a result of extending the NIKOS with new HAs.

The interoperability case studies presented in section 7.6 aimed to demonstrate that the HAP can successfully facilitate the communication and interoperability needs of HAs at different levels. More precisely, it was demonstrated that the HAP can address all the three levels of interoperability as these have been identified in Chapter 1. Additionally, it could well address different types of interoperability (e.g. storage interoperability, interoperability about link and analytical search services). It must be recalled, that the Open Hypermedia Protocol (OHP) presented in Chapter 3, does not address all these different types of interoperability.

Section 7.7 examined the NIKOS from an information seeking perspective. The NIKOS OHS advocates an interaction model for information seeking which is substantially different from the WWW. The most defining feature of NIKOS is that it advocates the parallel, interleaved use of multiple strategies, through the parallel coordinated use of multiple HAs. The author believes, that this interaction model is more effective that the “over-engineered” interaction model which is found in most hypermedia systems. In the next chapter, a user-centered evaluation is presented which provides more evidence about this claim.

Chapter 8

User-Centered Evaluation

This chapter presents a user-centered evaluation which aims to assess the effect of distributed parallel searching strategies to information seeking performance. It also aims to compare the link-based and the uniform fusion strategies in a realistic environment. Another important aim of the experiment is to evaluate the NIKOS OHS as an information seeking environment and to compare it with other HDLs based on the WWW.

8.1 The problem of Evaluation

Finding methods to evaluate the performance of IR systems is a major problem which traditionally attracted much interest (e.g. Sparck Jones, 1981; van Rijsbergen, 1979; Belkin, 1981; Sarajevic, 1995; Dunlop, 1997). Currently, the effectiveness of IR systems is mainly measured by two metrics: recall (R) and precision (P).

Most of the evaluations using R and P are system-centered; that is they do not involve users directly. The measures of R and P are taken as a result of ad hoc runs of standardised queries. The evaluation of the collection fusion strategies presented in Chapter 5, is an example of a system-centered evaluation.

The nature of highly interactive information seeking environments such as HDLs, however, appeals to more user-centered approaches for several reasons (Salampasis et al, in press). Perhaps, the most appealing of them is that information seekers in HDLs search using browsing strategies. Because of their highly interactive nature, browsing strategies are more dependent on the physical, emotive and cognitive abilities of information seekers than are analytical strategies. Hence, it is not realistic to artificially simulate browsing in the same manner as ad-hoc, stateless, non-interactive runs of standardised queries simulate operational query-based environments in P and R evaluations. This fact has been recognised and several calls have been made for user-centered evaluations (e.g. Salampasis et al, in press; Hersh et al, 1995; Wildemuth et al 1995).

In this chapter a user-centered experiment is presented to evaluate the performance of information seekers using different hypermedia digital libraries (HDLs). The calculation of R and P was based on a series of data produced during a sequence of interactions that users had with the systems under evaluation. This is the approach which the interactive track of the TREC-4 conference (Harman, 1996) suggests for evaluating interactive information retrieval systems (e.g. Charoenkitkarn et al, 1996), in response to some scepticism regarding the appropriateness of past TREC conferences which treated evaluations in batch mode (Sparck Jones, 1995).

8.2 Materials and Systems Used in the Experiment

Four different HDLs have been implemented for the experiment. These four HDLs were developed over exactly the same CACM distributed collection of hypermedia documents. The CACM collection composed of eight sub-libraries produced using a hierarchical clustering

process and, it was previously described in chapters 5 and 7. Three of the HDLs used in the experiment were based on the WWW. Therefore the CACM documents had to be converted into HTML. For this purpose the corresponding links were appropriately embedded into the CACM documents together with the titles of the destination documents to give a prospective view of the destination document and therefore to facilitate selection of a link. Figure 8.1 shows a CACM document in HTML format as it was used in our experiment. All the CACM documents had a similar layout and organised in the same way.

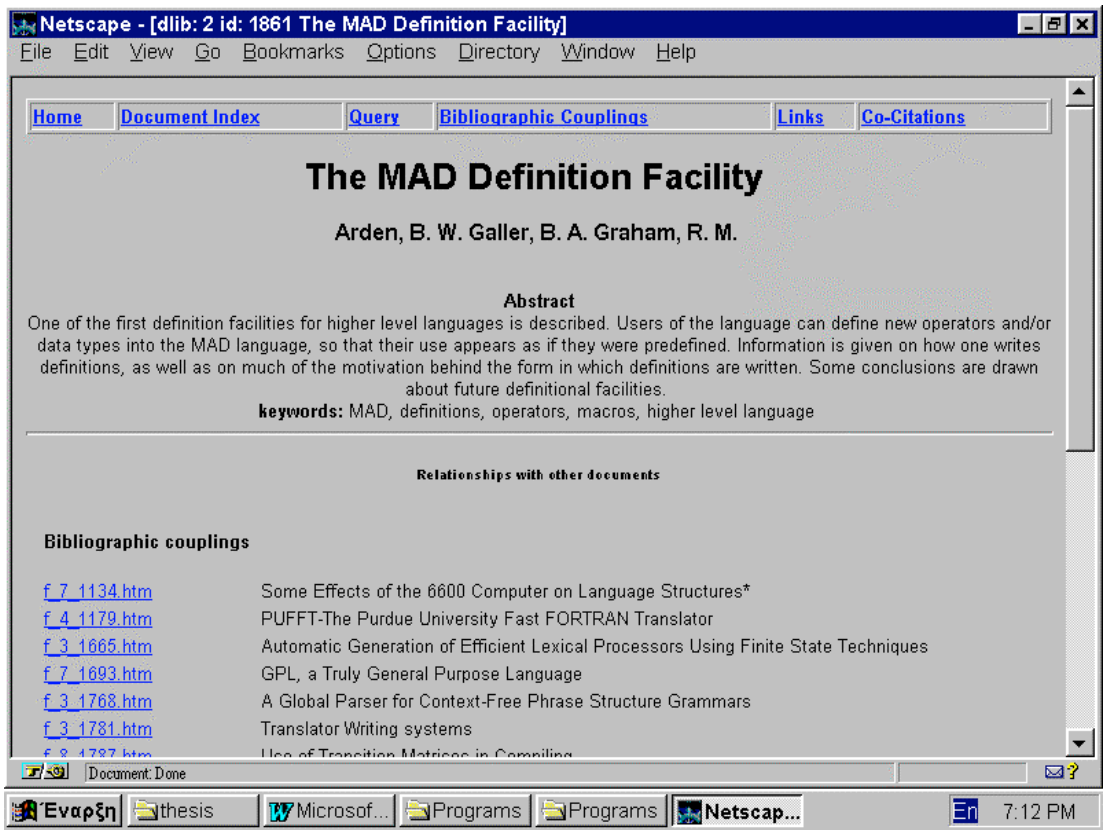


Figure 8.1: An example of a CACM document as it was used in our experiment

Although the three WWW-based HDLs were based on exactly the same raw material, each version had some differences with the others in terms of how users could search for information.

1. Subjects in the first WWW HDL could search for relevant documents using simple across document browsing, or "local" searching strategies. Using "local" searching subjects could search only one sub-library (from the eight available) at a time. So, if they wanted to search all sub-libraries for a given query, they had to repeat the query

for each sub-library and examine the results separately. Finally, the subjects could also use as a navigational aid, a browsable table of contents which had a link for each document member of a sub-collection.

2. The second WWW HDL supported exactly the same information seeking strategies as the first, but additionally subjects could perform distributed, parallel searches over the eight sub-libraries. That practically meant, that subjects could submit just one query to search all the sub-libraries in parallel and could examine one single merged result. In this second WWW-based HDL the uniform collection fusion strategy was used to solve the collection fusion problem.

3. the third WWW-based HDL was identical to the second but the link-based collection fusion strategy was used for solving the collection fusion problem in distributed searches.

The fourth CACM HDL which was used in the experiment was based on the NIKOS OHS. This NIKOS-based HDL was the one which is described in chapter 7. Several hypermedia agents were available to the information seekers to assist them in their information seeking process during the experiment. Subjects using the NIKOS-based HDL, like the second and the third WWW HDLs, had also the capability to perform distributed searches using the link-based fusion strategy. Table 8.1 summarises the basic characteristics of the HDL systems (i.e. conditions) that have been tested in our experiment.

	Browsable Table of Contents	Across-document Browsing	Clustered Browsing	"Local" Searching	Distributed Searching
WWW - 1 (Condition 1)	Yes	Yes	No	Yes	No
WWW - 2 (Condition 2)	Yes	Yes	No	Yes	Yes uniform strategy
WWW - 3 (Condition 3)	Yes	Yes	No	Yes	Yes Link-based strategy
NIKOS (Condition 4)	Yes	Yes	Yes	Yes	Yes Link-based strategy

Table 8.1: The basic characteristics of the four HDLs (conditions) tested

8.3 Experiment

The experiment was divided into two parts and followed a between subjects design. In the first part only the three WWW-based HDLs were used. This first part had two basic aims. The first aim was to assess the effects of using parallel searching in information seeking environments. The second aim was to compare the uniform and the link-based fusion strategies in a realistic, user-centered environment. In the second part of the experiment searches have been conducted using the NIKOS-based HDL and the aim was to compare the information seeking performance of the NIKOS-based HDL with the WWW-based HDLs.

8.3.1 First Part of the Experiment

Aims

Previously, throughout this thesis (e.g. sections 2.3, 4.2) arguments have been made to support the claim that a distributed parallel searching strategy, may increase the performance of information seekers. The first aim of the first part of this experiment was to attempt to evaluate these claims and arguments in a realistic information seeking environment.

The second aim was to compare the uniform and the link-based fusion strategies in a user-centered experiment. In Chapter 5 the link-based fusion strategy has been evaluated using the "classical", system-centered methodology. That evaluation showed that the proposed link-based strategy performs better than the uniform strategy. Now, by conducting this user-centered experiment the aim was to investigate if the same observations would be confirmed in realistic information seeking environments. The need to compare the fusion strategies in a user-centered evaluation was driven by the author's belief that more complete and accurate conclusions can be drawn if the results from both the system-centered and user-centered experiments are considered.

Method

Thirty six subjects voluntarily participated in the first part of the experiment. All the subjects were computing science undergraduate students in the final year of their degree (9), or postgraduate students studying for M.Sc. (8) or research degrees (19). An nearly equal proportion of students from each category was allocated to each group in order to preserve the homogeneity of the groups[26]. All the participants had past experience using the WWW on a daily basis.

Subjects were tested individually. A written description about the WWW-based HDLs was given to the subjects before the tests, to help them gain an overview of the system to be used (Appendix B). Also, a brief fifteen minutes training session was conducted with each subject before testing, to ensure that s/he could search the HDL and also s/he understood the nature of the task that s/he will be asked to perform. No formal training in information seeking strategies was given to the subjects.

Subjects were divided into three groups. Each group used one of the WWW-based HDLs described in last section (i.e. conditions 1, 2 & 3). Each subject in the group was informed about the available information seeking strategies (Table 8.1). Then, each subject was given an information problem (i.e. a query) and asked to find as many relevant documents in 30 minutes using their preferred strategy or combination of strategies from those which were

[26] Besides that, it was very unlikely that the larger knowledge that postgraduate students may have in some fields of computer science could be a serious reason for increased performance, because the CACM collection covers rather old computer science material about which none of the participants was likely to have any specialised knowledge.

available. To motivate their searching the number of relevant documents was given to the subjects. If a subject found all the relevant documents in less than 30 minutes the search session was finished earlier than this duration limit. The subjects were also asked to write in a special prepared form the numbers (i.e. the id's) of the documents that they viewed and they judged as being relevant to their query. This list of documents is called the "judgement list" of a subject.

Each subject performed two 30 minutes search sessions. So in total in the first part of the experiment 72 thirty-minutes search sessions were performed. The first session for each subject used documents like the one shown in Figure 8.1. In the second session that each subject undertook, the same CACM documents were used but with one small difference. For each relevant document the queries for which it was relevant were clearly specified in the document. We call this second group "assisted" sessions. Figure 8.2 shows the same document which is illustrated in Figure 8.1, changed to be used in the "assisted" sessions. So, in this second session the subjects could directly identify if the current viewed document is relevant to their query or not. One reason for giving this type of assistance in each subject's second session was to measure the effects that high knowledge on a subject area might have to the performance of a searcher. But, the main reason was to differentiate between the effects of distributed parallel searching for "knowledgeable" information seekers and for "non-knowledgeable" users.

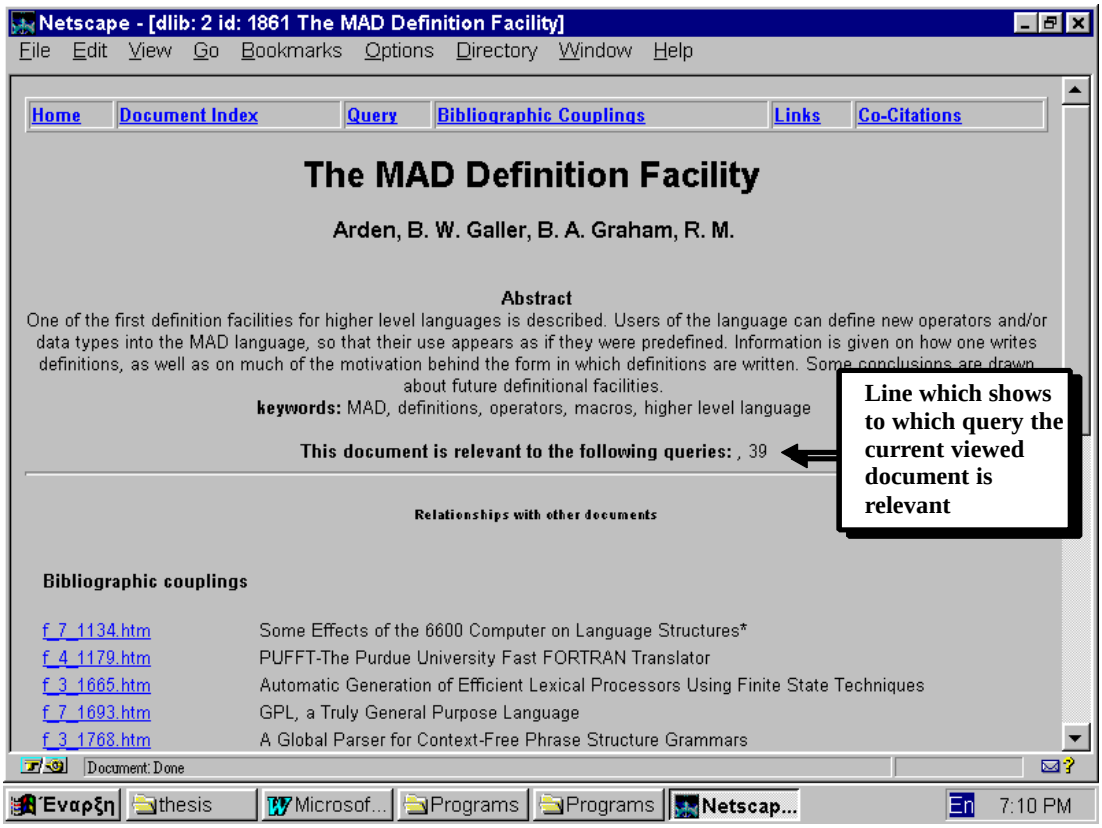


Figure 8.2: An example of a CACM document as it is used in the "assisted" search sessions.

Each search session was logged and the data were analysed. The searches of three subjects were corrupted during the logging process by the WWW server. For the remaining 33 search session values for the following measures were calculated:

1. *Minutes*, the time in minutes that the session actually lasted;
2. *First found*, the minute in which the first relevant document was found;
3. *JR*, the judged recall at the end of the session;
4. *JP*, the judged precision at the end of the session;
5. *States*, the total number of different states (movements) that a searcher move through during the search session.

Note that the metrics JR and JP refer to "judged" recall and precision. In other words, these are the values for R and P based on the "judgement list" that the subjects produced during the experiment. Of course, it is possible that subjects have viewed (i.e. open in the browser) a relevant document without recognising/judging that it is relevant (obviously, that was possible

only in the unassisted searches), so they didn't write this document in the "judgement list". So, we also measure the *viewed recall* in thirty minutes (VR). Finally, it was possible that relevant documents have been retrieved from an analytical search, but never viewed by the subjects. Therefore, we also measure the *retrieved recall* in thirty minutes (RR). Table 8.2 defines JR, VR and RR.

$$JR = \frac{\text{Relevant documents being written in the judgment list}}{\text{Relevant documents in the whole collection}}$$

$$VR = \frac{\text{Relevant documents viewed}}{\text{Relevant documents in the whole collection}}$$

$$RR = \frac{\text{Relevant documents retrieved}}{\text{Relevant documents in the whole collection}}$$

Table 8.2: Definitions for JR, VR and RR

Given that a document must be retrieved before can be viewed, and must be also viewed before it is written in the "judgement" list, it can be inferred that the following equation should always hold:

retrieved recall >= viewed recall >= judged recall

The combined use of the retrieved, viewed and judged recall can give a more complete picture of the performance of an information seeking environment (Charoenkitkarn et al, 1996). Therefore all these metrics are used in the experiment.

Results of unassisted searches

Tables 8.3 and 8.4 illustrate the averaged results of the unassisted search sessions (mean and standard deviation). The two columns at the right of Table 8.3 indicate the proportion of "analytical" states ("local" and distributed searches) and the proportion of browsing states.

	Minutes	**First found**	**States**	**Querying**	**Browsing**
Condition 1	26.17 4.06	9.00 *5.12*	155.75 57.96	32%	68%
Condition 2	26.64 5.95	8.00 *6.26*	126.64 49.80	28%	72%
Condition 3	25.60 6.59	6.10 *7.13*	129.20 51.67	28%	72%

Table 8.3: Basic statistics of the unassisted search sessions.

	JR	**JP**	**VR**	**RR**
Condition 1	0.20 *0.17*	0.33 *0.30*	0.42 *0.26*	0.53 *0.31*
Condition 2	0.18 *0.12*	0.37 *0.32*	0.37 *0.23*	0.42 *0.23*
Condition 3	0.40 *0.28*	0.48 *0.26*	0.64 *0.28*	0.68 *0.26*

Table 8.4: Performance results of the unassisted search sessions.

In respect to the first aim of this first part of the experiment (i.e. to assess the effect of distributed searching in the information seeking performance), the results shown in Tables 8.3. and 8.4 confirm the hopes that HDLs which support link-based parallel and distributed searching (i.e. condition 3) are more effective than information seeking environments that support only single searching strategy (i.e. condition 1) (Figure 8.3). More precisely, the results for R and P for the third condition are significantly better in all cases than the results obtained in condition 1.

On the other hand, the R and P results of the second condition (parallel searching using the uniform strategy) are worse in most of the measures calculated (JR, VR, RR) and only in terms of judged precision (JP) the results are slightly better. Generally, in comparison to the large differences that have been observed between the third (link-based parallel searching) and the first condition (single searching), the differences between the second condition (uniform parallel searching) and the first condition are small.

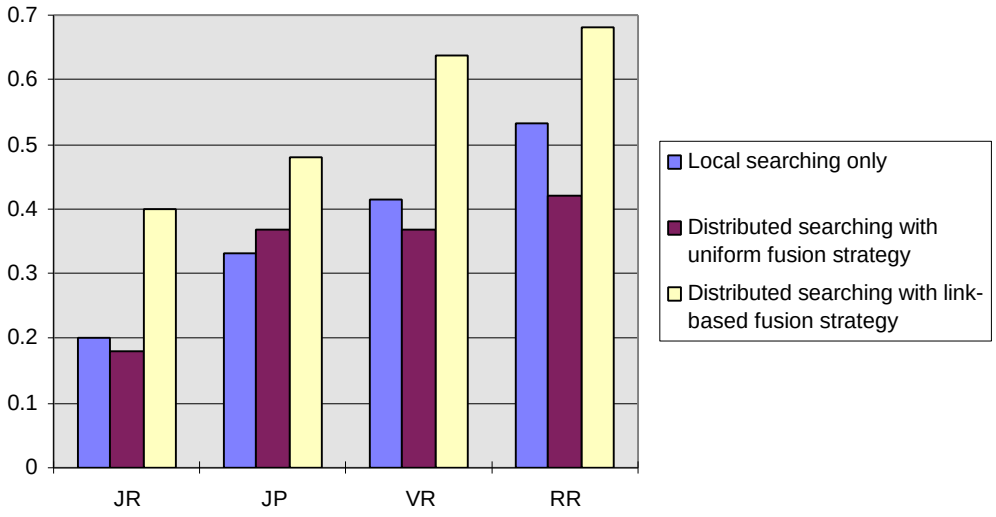


Figure 8.3: Judged R and P, Viewed and Retrieved R for unassisted searches.

These results are in line with the discussion made in chapter 4 where different fusion strategies have been presented. The results clearly illustrate that parallel searching using the link-based fusion strategy is consistently more effective than a single searching strategy. On the other

hand, the results also indicate that some parallel searching strategies will not always perform better that the single searching strategy. For instance, multi-database searching which treats all the sub-collections the same (i.e. the uniform approach) may not be always more effective than single searching.

The results in Table 8.3 also show that information seekers in the distributed parallel searching conditions find the first relevant document to their information problem earlier. Finally, they illustrate that the users in the parallel distributed searching condition go through less states (i.e. make less transitions) during the information seeking process than the users in the single search condition (Figure 8.4). Finding the first relevant document sooner is useful in information tasks where only one or a few relevant documents can satisfy the information need (precision oriented problems). Also the fact that users in the distributed condition move through fewer states may indicate that these users may develop less cognitive overhead during the searching process than the users in the single searching condition.

From the results outlined above we can generally conclude that parallel distributed searching has a positive effect on an information seeking environment. It can also be concluded that the positive effect is significant in the case of using the link-based collection fusion strategy. The positive effect applies both to metrics such as R and P which generally reflect the effectiveness of a system and, to other measurements such as the ability to find sooner the first relevant document or to produce less states during an information seeking process, which are some of the factors which determine the efficiency of an information seeking environment. On the other hand, distributed searching using the uniform approach has not a significant effect in comparison to single search.

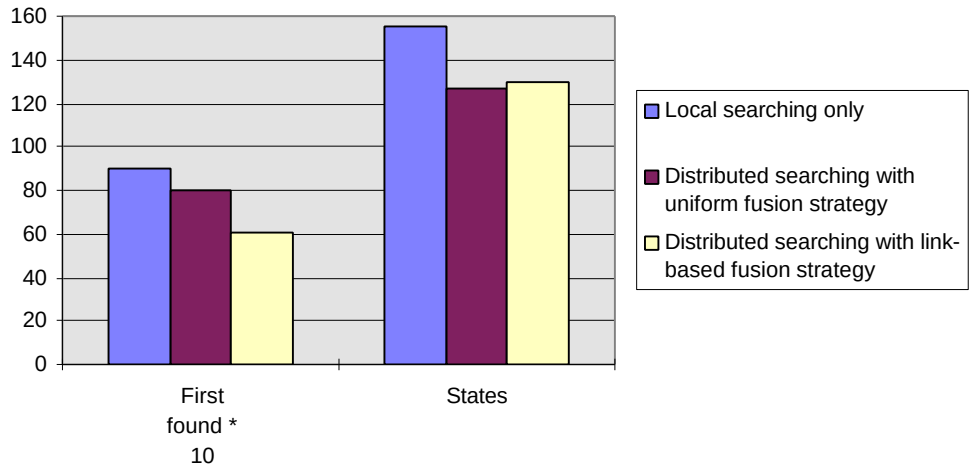


Figure 8.4: Minute in which first document was found and states produced for unassisted searches (for presentation reasons the variable "First found" has been multiplied by 10).

In respect to the second aim of this first part of our experiment (i.e. to compare the uniform and the link-based strategies in user-centered evaluations), the results illustrated in the tables and figures above show that the link-based fusion strategy performs significantly better than the uniform fusion strategy. More precisely, in all the cases the link-based strategy has produced better R and P results and users in this condition have generally found earlier their first relevant document earlier.

These results are even more encouraging if we compare the results of the user-centered evaluation with the respective system-centered results obtained in chapter 5 using the CACM-8 collection (Table 5.2 in Chapter 5). Although the results in chapter 5 using the CACM-8 collection (which is exactly the same as the one used in this user-centered experiment) do not show a difference between the uniform and the link-based strategies, in contrast the user-centered evaluation shows some significant differences in their performance.

The author believes that this can be explained by the additional benefits that the link-based strategy may deliver to information seekers. The link-based fusion strategy gives an indication about which sub-libraries are likely to contain more relevant documents. This indication is not given by the uniform strategy which retrieves documents uniformly from all the participating collections. Therefore, although both strategies may have equal performance in ad hoc runs of standardised queries, in realistic environments, users exploit the feature of the link-based

strategy to focus into a limited number of libraries and, therefore are able to produce better effectiveness and efficiency results than the users in systems with a uniform strategy.

Results of assisted searches

Table 8.5 illustrates the results for the assisted search sessions.

	Minutes	First	JR	States	Querying	Browsing
Condition 1	19.50	5.75	0.75	224.50	31%	69%
	8.59	*3.08*	*0.24*	*86.50*		
Condition 2	20.09	3.55	0.89	259.73	23%	77%
	10.63	*2.25*	*0.15*	*155.71*		
Condition 3	16.00	3.90	0.91	159.00	28%	72%
	9.06	*3.28*	*0.16*	*81.65*		

Table 8.5: Search results for the "assisted" search sessions.

The results shown in the table above are generally in line with the results which are obtained from the unassisted searches. Subjects in the link-based parallel searching strategy (i.e. condition 3) have produced better results than subjects in the single searching condition (i.e. condition 1). The results were significantly better both in terms of effectiveness (Figure 8.5) and also in terms of efficiency (Figure 8.6). Combining these results with the results obtained in the unassisted searches we can generally conclude that the link-based parallel searching strategy consistently performs better than the single searching strategy, regardless of how knowledgeable are the users conducting the searches.

The comparison between the link-based (condition 3) and the uniform fusion strategies (condition 2), demonstrates again that the link-based has performed significantly better than the uniform strategy in terms of efficiency (Figure 8.6), and slightly better in terms of effectiveness (Figure 8.5). Again, combining the results obtained from the unassisted and assisted searches, we can generally conclude that the link-based fusion strategy consistently

performs better than the uniform fusion strategy, regardless of how knowledgeable are the users conducting the searches.

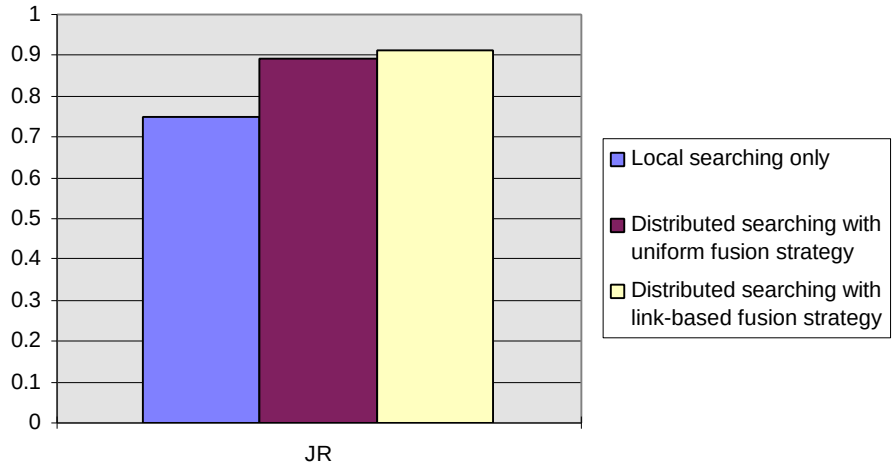


Figure 8.5: Results of JR for assisted searches.

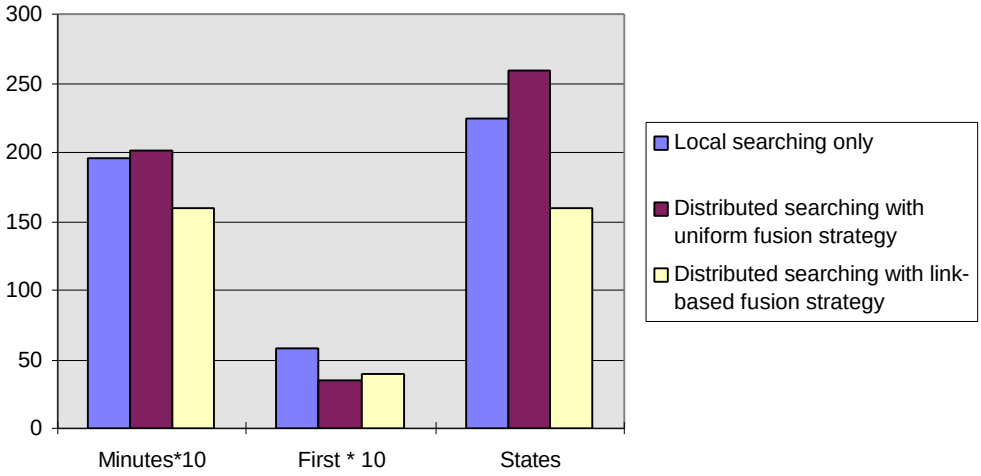


Figure 8.6: Minute in which first document was found and states produced for assisted searches (for presentation reasons the variable "First found" has been multiplied by 10).

8.3.2 Second Part of the Experiment

Aims

In the last chapter a claim was made that NIKOS is a rich information seeking environment which is flexible, extensible and customisable and, therefore it will potentially be an effective information seeking environment. The second part of the experiment aimed to investigate this claim by evaluating the NIKOS OHS as an information seeking environment. It also aimed to compare information seeking performance using the NIKOS-based HDL with the performance of equivalent WWW HDLs. NIKOS from an information seeking perspective is dissimilar to the WWW in several ways which have been discussed in the end of the last chapter. Now, this experiment aims to investigate if these dissimilarities are able to produce any significant differences in the performance of information seekers.

Method

Twelve subjects participated in the second part of the experiment. The subjects were randomly selected from the thirty six subjects who participated in the first part of the experiment. The second part of the experiment using the NIKOS-based CACM hypermedia digital library took place five months after the first part. Therefore the first experience that subjects had with the WWW-based CACM library, was very unlikely to give any advantage to subjects in this second part of the experiment. Additionally, no subject performed the same query that s/he performed in the first part of the experiment.

Subjects were tested again individually. None of the participants had any past experience using the NIKOS OHS. A fifteen minutes presentation which outlined the NIKOS OHS was given to the subjects before the test, to help them gain an overview of the system to be used. Again, no formal training in information seeking strategies was given to the subjects.

Each subject was given an information problem (i.e. a query) and asked to find as many relevant documents in 30 minutes using their preferred strategy (i.e. hypermedia agent) or combination of strategies from those which were available. Again, to motivate their searching the number of relevant documents was given to the subjects. If a subject found all the relevant documents in less than 30 minutes the search session was finished earlier than this time limit.

Each subject performed one 30 minutes unassisted search session. So, in total for this fourth condition 12 thirty minutes search sessions were performed. The subjects were asked again to write in a special prepared form the numbers (i.e. the id's) of the documents that they believed

were relevant to their problem. At the end of the search session each subject was given a questionnaire having 17 questions, half of them positive and half negative (Appendix C). The questions were divided into five sections. The nature of each section and of each question was orally explained to the subjects, who afterwards anonymously gave their responses. Each question could be answered using a five scale answer list where the middle response was neutral. If a subject did not experience the subject of a question, s/he could choose the sixth "No Opinion" response.

The states of the subject's search sessions were logged by the history hypermedia agent. For each search session values for the measures used in the first part of experiment were calculated.

Results of the second part of the experiment - Questionnaire

Figures 8.7 through 8.23 present the results of the questionnaire that subjects answered after using the NIKOS OHS. Each figure presents the question and the two marginal responses in the scale (i.e. the most positive has code 1 and the most negative has code 5). Neutral responses have the code 3 and N/O (no opinion) responses are presented with code 6.

System speed

Figure 8.7 shows the responses to the first question which is regarding the speed of the NIKOS system. All the responses were positive and this fact illustrates that despite the message exchange between hypermedia agents (about 500 messages are exchanged in each thirty minutes search session), the speed of the NIKOS OHS was acceptable.

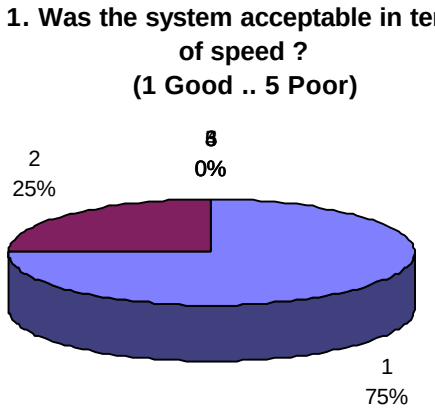


Figure 8.7: Responses to question 1.

System Comprehension

The four questions in this section of the questionnaire aimed to explore if it was easy or difficult to understand the NIKOS OHS during the presentation and training session. This section basically refers to the comprehension of the NIKOS OHS *before* it was used by the subjects. One could say that the NIKOS system is more complicated than the WWW and, therefore users might be confused or they might find it difficult to use the system. However, the practical experience gained from the experiment shows that a fifteen minutes presentation was enough so subjects could start using the system.

The second question (Figure 8.8) explores if subjects had any difficulty in understanding the data model of NIKOS OHS. The responses shown in Figure 8.8 indicate that the comprehension of the data model was not a problem for most subjects.

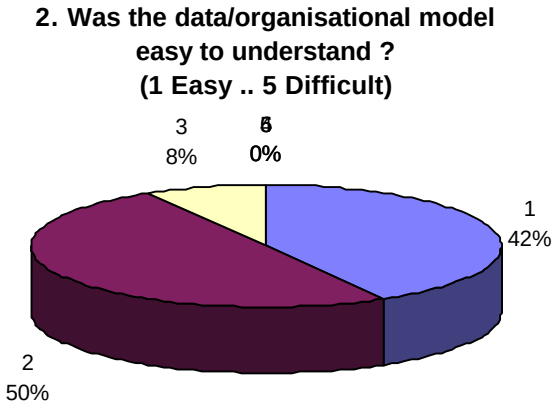


Figure 8.8: Responses to question 2.

The next three questions (Figures 8.9, 8.10 & 8.11) examine if the process model that NIKOS OHS suggests for information seeking (e.g. information seekers can open and use concurrently multiple hypermedia agents, they can "switch" from one hypermedia agent/strategy to another), was difficult for subjects to understand. Note that the fifth question (Figure 8.11) refers to the interaction method with the NIKOS OHS as a *whole system* (i.e. having multiple hypermedia agents open on the desktop and managing these agents during information seeking) and not to particular hypermedia agents.

The responses in these questions show that almost none subject felt that it was difficult to understand the information seeking process and interaction model promoted by NIKOS OHS.

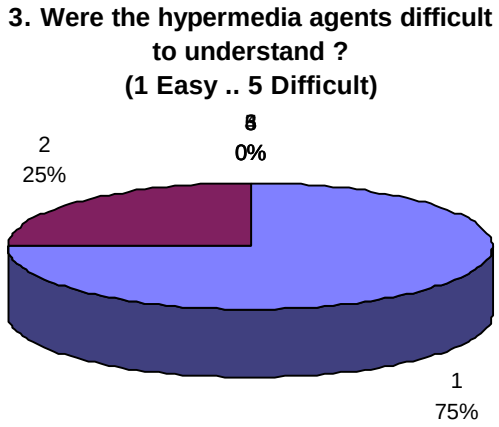


Figure 8.9: Responses to question 3.

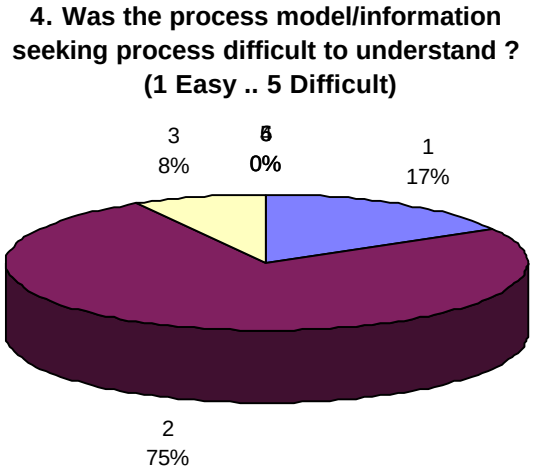


Figure 8.10: Responses to question 4.

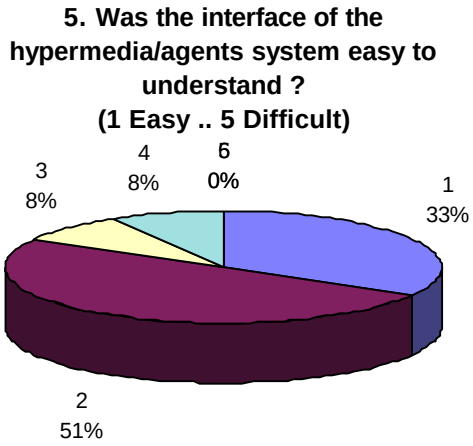


Figure 8.11: Responses to question 5.

User interface

The two questions in this section (Figures 8.12 and 8.13) of the questionnaire aimed to examine the interface of hypermedia agents.

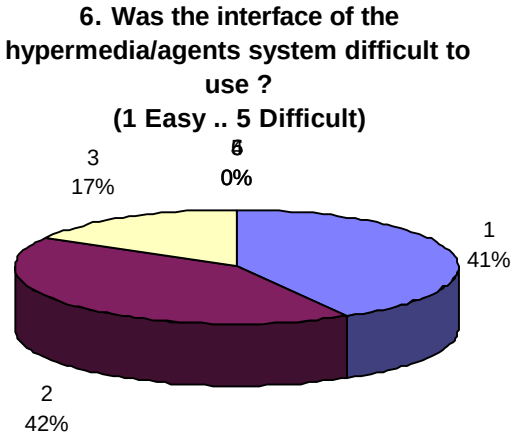


Figure 8.12: Responses to question 6.

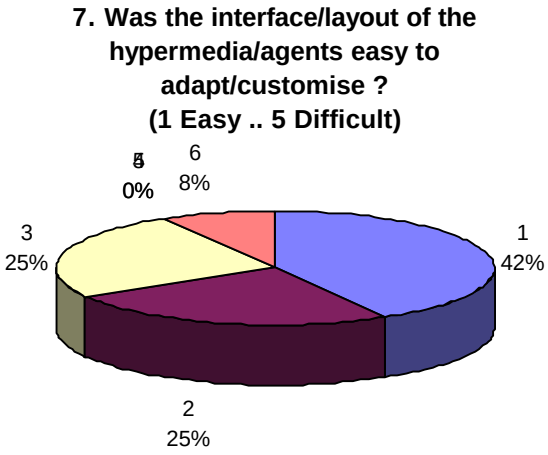


Figure 8.13: Responses to question 7.

Usability

The five questions in this section aimed to examine how much easy or difficult it was for subjects to use the NIKOS OHS. This section refers to the experiences that subjects had *during* their search sessions using the NIKOS OHS. Question number 8 (Figure 8.14) aimed to examine the general impression of the NIKOS' usability. The next question (Figure 8.15) refers to the difficulty that subjects could have to coordinate different programs (i.e. hypermedia agents) during their information seeking activities. The next question (Figure 8.16) explores NIKOS in terms of interactivity. Finally, the next two questions (Figures 8.17 and 8.18) aimed to assess the usability of the NIKOS OHS in conducting simple and parallel analytical searches.

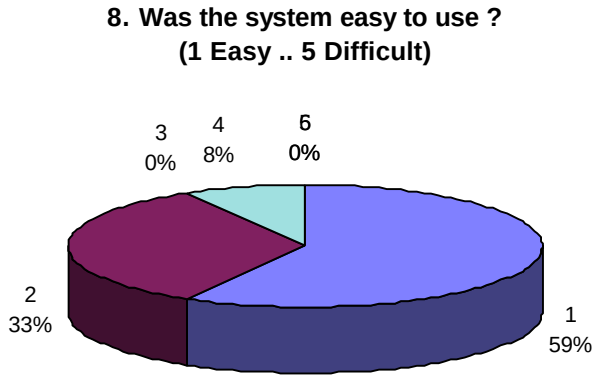


Figure 8.14: Responses to question 8.

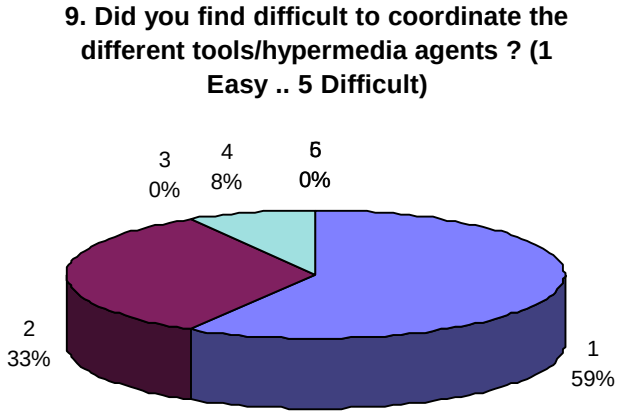


Figure 8.15: Responses to question 9.

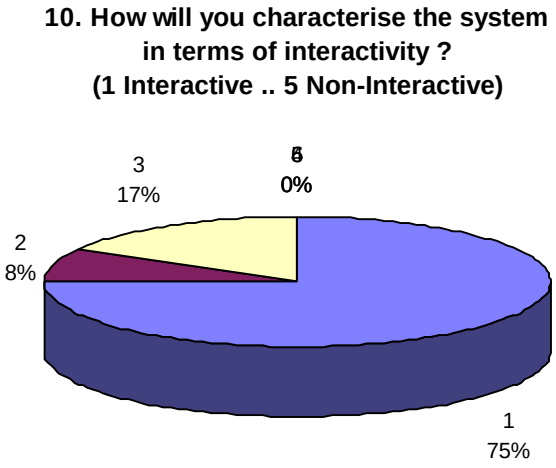


Figure 8.16: Responses to question 10.

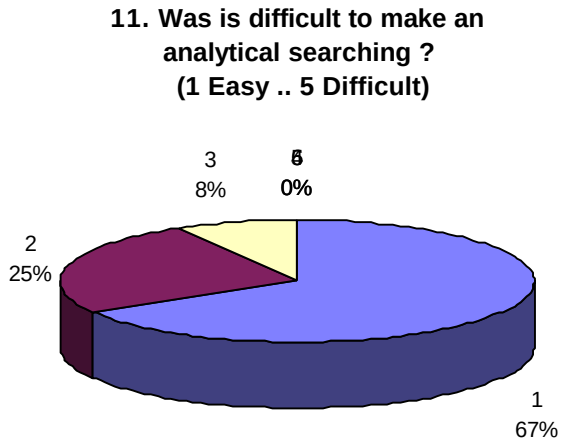


Figure 8.17: Responses to question 11.

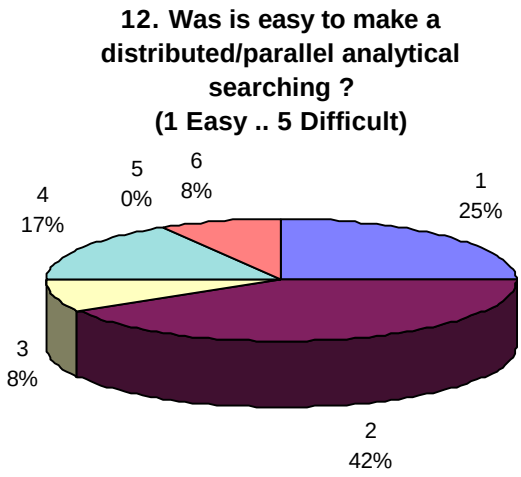


Figure 8.18: Responses to question 12.

Information seeking issues

The five questions in this last section aimed to examine NIKOS OHS from an information seeking perspective. The question shown in Figure 8.19 aimed to examine the opinion of the subjects about the usefulness of parallel distributed searching in NIKOS. As it is illustrated in this figure most of the subjects had a positive opinion about the usefulness of this searching strategy.

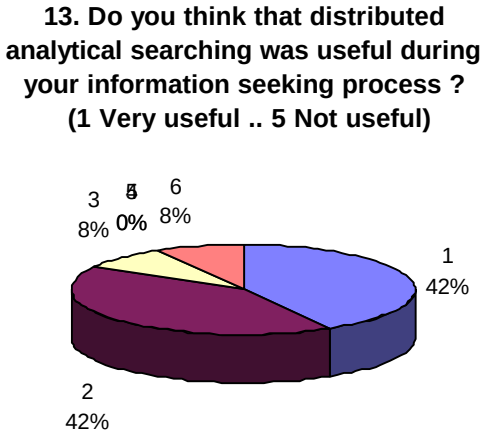


Figure 8.19: Responses to question 13.

The next question shown in Figure 8.20 examines the usefulness of the source selection and suggestion which is implemented in NIKOS using the collection fusion hypermedia agent. In this question half of the subjects had a positive opinion about having this feature. However, one third of the subjects had a neutral opinion about the usefulness of this feature. Subjects also had unanimously expressed a positive opinion about the usefulness of clustered browsing (Figure 8.21).

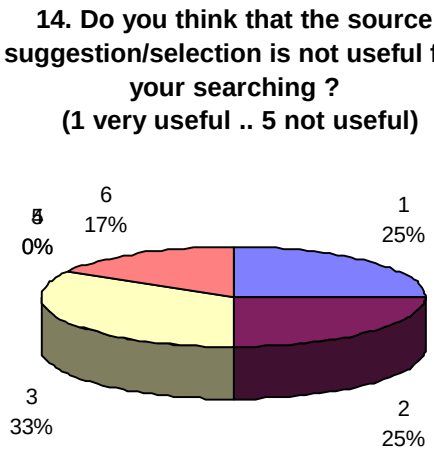


Figure 8.20: Responses to question 14.

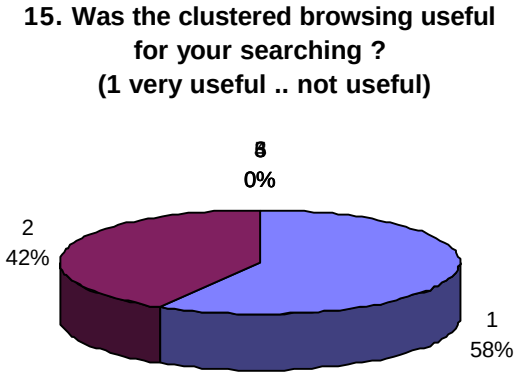


Figure 8.21: Responses to question 15.

The next question shown in Figure 8.22 aimed to explore the difficulty that subjects had to use and "switch" from one hypermedia agent to another. Each hypermedia agent represents a

different information seeking strategy and therefore information seekers might find mentally or cognitively demanding an information seeking environment which promotes the use of multiple strategies and the switch from one strategy to another. Subjects unanimously expressed their opinion that it was easy to "switch" and use multiple strategies. Also most of them (66%) had a positive opinion about the combined used of multiple strategies in comparison to using a single strategy (Figure 8.23).

The mostly positive opinions that subjects have expressed in these two questions are very encouraging for NIKOS as an information seeking environments. As it was already discussed in chapter 7, NIKOS as an information seeking environment is based on a philosophy which advocates the concurrent use of multiple strategies. The responses shown in figures 8.22 and 8.23 show that subjects generally agree with this approach.

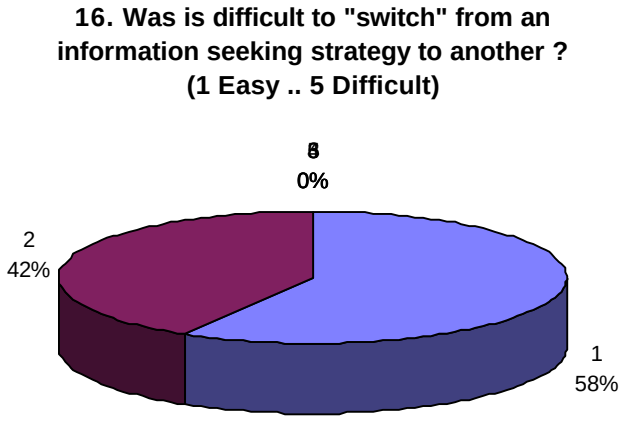


Figure 8.22: Responses to question 16.

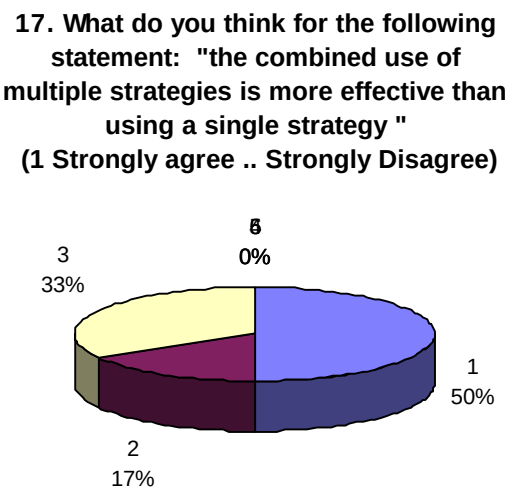


Figure 8.23: Responses to question 17.

R and P results of the second part of the experiment

Table 8.6 illustrates the R and P results obtained using the NIKOS-based HDL (condition 4) and the two WWW HDLs (condition 2 and 3) which are directly comparable with the NIKOS HDL, because they both support parallel distributed searching.

	JR	**JP**	**VR**	**RR**
Condition 2	0.18	0.37	0.37	0.42
(WWW-2)	*0.12*	*0.32*	*0.23*	*0.23*
Condition 3	0.40	0.48	0.64	0.68
(WWW-3)	*0.28*	*0.26*	*0.28*	*0.26*
Condition 4	0.46	0.70	0.62	0.75
(NIKOS)	*0.38*	*0.36*	*0.35*	*0.24*

Table 8.6: Performance of subjects using the NIKOS-based HDL (condition 4) in thirty minutes.

This table shows that subjects using the NIKOS OHS performed significantly more effective than subjects using the WWW almost in all the cases examined. More precisely, the subjects using the NIKOS OHS have produced significantly better results than the subjects using the WWW in condition 2 in all the cases. Also, subjects using the NIKOS OHS have produced significantly better results for JR, JP and RR than subjects using the WWW in condition 3 (Figure 8.24). Only, the viewed recall (VR) in condition 4 was marginally smaller than in condition 3 (0.62 and 0.64 respectively).

The results generally are very encouraging given that the WWW-based HDLs in conditions 2 and 3, were enhanced with parallel searching.

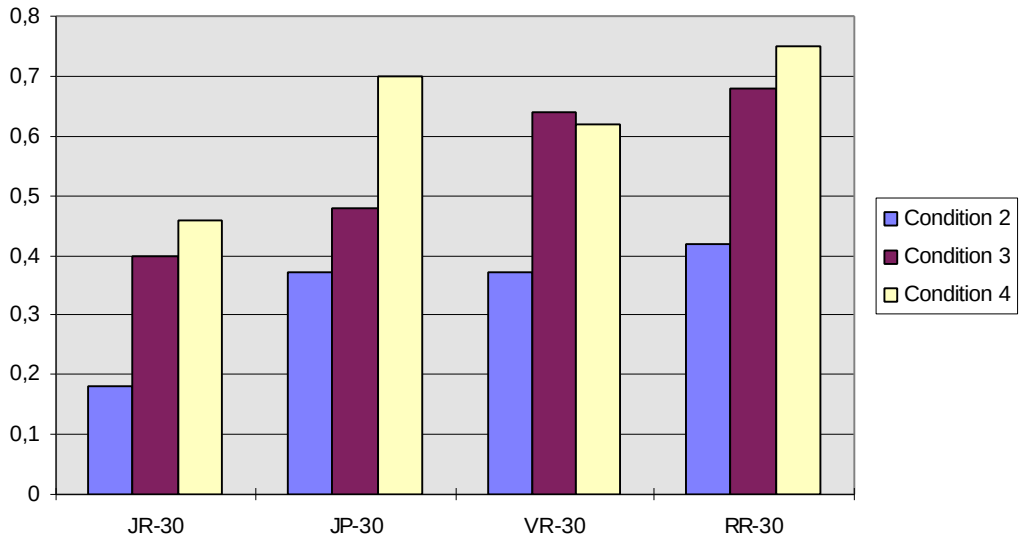


Figure 8.24: Comparison of performance between NIKOS-based HDL and two WWW-based HDLs.

8.4 Limitations of the Experiment

Our experiment, as any other experiment, is bounded by some limitations. The first limitation is regarding the artificial separation made in the experiment between "knowledgeable" and "non-knowledgeable" searches. A knowledgeable searcher in a particular subject is not characterised only by her/his ability to instantly "recognise" relevant documents, as it is taken to be in our experiment. There are additional abilities such as the ability to distinguish important from unimportant terms and to use these terms for searching. However, this limitation can affect only one aim of our experiment. This is the aim to study if the effect of distributed searching for “knowledgeable” information seekers will be different from the effect to “non-knowledgeable”. All the other aims and findings (e.g. comparison between the link-based and uniform strategy) are totally independent of this separation and, therefore not affected.

The other limitation has to do with the artificiality of methods that have been used to create the HDLs, and more precisely to create a distributed HDL from the single CACM collection. This issue has been already discussed in Chapter 5.

Despite these limitations, however, the author believes that the experiment presented in this chapter is useful, because it involved a relatively large number of users and minimised (as

much as possible) all the external factors which could affect the soundness of the experiment. In fact, to the best of the author's knowledge this experiment is the first user-centered experiment ever conducted which evaluated the effect of distributed parallel searching on information seeking performance. It also is the first user-centered experiment which aimed to compare two different collection fusion strategies in a realistic environment.

8.5 Conclusions

Given the limitations outlined in the last section, the experiment which has been described in this chapter suggests the following.

- Distributed searching strategies which search in parallel multiple collections and produce a single merged result, have a positive effect to information seeking performance. Information seekers using parallel distributed searching strategies performed significantly more effectively than information seekers which can only do "local" single library searching. They also performed more efficiently as some measures indicate (e.g. the time in which the first document is found and the less states which are produced during the seeking process).

- The link-based collection fusion strategy performs significantly better than the uniform fusion strategy both in terms of effectiveness and efficiency. Given that similar conclusions have been drawn based on the results obtained from the system-centered evaluations in chapter 5, it could be said that the claim which is made in this thesis (i.e. that the link-based fusion strategy performs better than the uniform approach) is now supported by a significant amount of evidence.

- Distributed parallel searching is generally useful and increase the performance of both "knowledgeable" and "non knowledgeable" information seekers. However, in “knowledgeable” information seekers the relative benefit of using parallel searching strategies is smaller than when these strategies are used by “non-knowledgeable” information seekers.

- The NIKOS OHS is more effective information seeking environment than the WWW. The comparison of the results illustrated that users using the NIKOS-based HDL performed better with users using WWW-based HDLs. Note that these two WWW-based HDLs were appropriately enhanced so they could support single and distributed parallel analytical searching. If these results are combined with the responses of twelve to a set of questions examining the NIKOS OHS, it could be

said that there is a significant amount of evidence which supports the claim that the NIKOS OHS is an effective and efficient information seeking environment.

Chapter 9

Conclusions

This chapter summarises this Ph.D. thesis and revisits the hypotheses that have been presented in Chapter 1. It also discusses the novelty of the research work which has been carried out and the original contribution to knowledge. Finally, it outlines further work which can be undertaken, as the implications of the work and the results that have been produced in this Ph.D. research programme.

9.1 Conclusions

The first aim of this Ph.D. work was to design, implement and test a collection fusion strategy, which would solve the collection fusion problem in hypermedia digital libraries using linkage information. The second aim was to design, implement and evaluate a distributed Open Hypermedia System (OHS) which would be engineered according to the principles of agent-based software engineering. It aimed to use this agent-based OHS as an underlying framework for realising hypermedia digital libraries which can address the architectural challenges identified in this thesis. It was also hoped that the resulting HDL will be flexible enough so different information seeking strategies, for example the link-based collection fusion strategy, could be integrated. The objective was to illustrate the potential of OHSs, not only as information integration and management environments, but their potential to become rich and effective information seeking environments.

The preceding chapters have shown in detail the methods which are used to achieve these aims, the motivation and rationale of these methods and, the methodical system-centered and user-centered evaluation of the results which have been obtained.

The link-based collection fusion strategy which was presented in Chapter 5 has two important characteristics. First, it solves the collection fusion problem solely using linkage information extracted from local linkbases at run-time. Second, it does not require any learning phase before it can be utilised. Other collection fusion methods reported in the literature use additional information from remote databases or require a learning phase. Most of the published papers discussing these collection fusion strategies, show a relative ignorance that both approaches are inconvenient when applied to large and dynamic environments such as hypermedia digital libraries.

The author believes that this is an important problem and therefore should not be ignored. It may be true that developing fusion strategies which do not require excessive amount of information from remote libraries nor require a learning phase, is a difficult road to follow. However, the results produced from the link-based fusion strategy (Chapters 5 & 8) indicate that it is possible to develop fusion strategies which are isolated and do not require a learning phase and, which perform better than the "obvious" uniform and random approaches.

Another important element of the research which is presented in Chapter 5 is the emphasis which is given to studying the efficiency of collection fusion strategies. Many researchers who published work on collection fusion strategies study the effectiveness issues (i.e. R and P), but

they show a limited interest about efficiency. A significant part of the research which was carried out in this Ph.D. programme and reported in Chapter 5, has been the work on measuring the efficiency of fusion strategies by using the total number of sub-libraries which are involved in a distributed retrieval run.

Of course, the collection fusion problem and other information retrieval problems which have recently emerged, are just an aspect of developing hypermedia digital libraries which is basically algorithmic. The development of hypermedia digital libraries, however, involves also crucial architectural issues. The author believes that efforts to develop digital libraries will be successful, if both issues are properly considered.

The design of the agent-based OHS architecture and prototype system which was presented respectively in Chapters 6 and 7 was driven by this idea. The proposed architecture utilises an arsenal of concepts and ideas that have been used in other computer science disciplines such as Software Agents, Cooperative Knowledge Based Systems (CKBS) and Multi Agent Systems (MAS). These ideas have been appropriately shaped and tailored so they can be adopted in the design and development of an OHS-based hypermedia digital library. The result is an open and extensible framework which takes into account *both* the architectural issues and information seeking issues arising in digital libraries.

From an architectural point of view, the key aspect of the agent-based OHS architecture is its deliberate emphasis on interoperability and superiority of OHS protocols to OHS architectures. The author believes this is a key issue and design decision in the development of hypermedia systems which will be really open in many different aspects.

From, an information seeking perspective, the agent-based OHS architecture implements an informal, open and adaptable information seeking environment. In this environment information seekers can amend their information workplace and make it more suitable to their specific needs. Additionally, the framework is extensible so a wide range of search techniques can be incorporated into the system. To summarise, the author believes that because the NIKOS OHS is based on the framework outlined above, it has the following characteristics.

? *It can incorporate and integrate many information seeking strategies*. This should be a primary goal for any information system since it allows the development of rich information workplaces.

- *It allows parallel, interleaved use of different information seeking strategies.* Autonomy of Hypermedia Agents (HAs) which represent different information seeking strategies, allows users to use in a parallel and coordinated fashion different strategies to search for information. Users may also like to switch to different HAs for reasons such as slow response of a particular HA or to monitor the progress of other searching processes.

- *It supports incremental learning and use of the electronic environment.* An information seeker can start searching by using the HAs which implement the simplest strategies. When the information seeker acquires more experience about the environment and the information space under searching, s/he can incrementally use additional HAs to amplify the capabilities of her/his information seeking environment.

- *It allows synchronisation between different information seeking strategies.* The HAs in our architecture can use message exchange to synchronise their views of the information space that provide to the information seeker.

- *It supports integrated management of results.* An electronic environment based on NIKOS gives the opportunity to the information seekers to easily manage their information after retrieval. This opportunity derives from the capability of OHSs to integrate different applications, usually viewers, to present or process data.

- *It facilitates iterative query refinement.* Users are often unable to immediately construct a query which will effectively describe their information need. Also, even if they are able to accurately describe their information need, often this information need will change as a result of the information seeking process. Usually users must reformulate their queries to describe better their information need.

9.2 Hypothesis Revisited

As it was hypothesised in the first chapter of this thesis (section 1.5.1), it was possible to use linkage information, in order to solve the collection fusion problem in hypermedia digital libraries. This new and novel algorithm and procedure was presented in Chapter 5, together with a formal system-centered evaluation which illustrated the effectiveness and efficiency of the link-based fusion strategy in comparison to other fusion strategies that can be applied under the same conditions. Also, as it was additionally hypothesised, information seekers can significantly benefit from this fusion algorithm. More precisely, the user-centered experiment

that has been presented in chapter 8, shows that information seekers can benefit both by increasing the effectiveness and the efficiency of their information seeking activities.

As the other hypothesis outlined in chapter 1 predicted, it was possible to design and develop an Open Hypermedia System (OHS) based on an agent-based software engineering approach. The development of a prototype hypermedia digital library application based on the agent-based OHS illustrated that it was also possible to use the OHS as an underlying platform for developing a hypermedia digital library. A series of controlled experiments have been presented in chapter 7 which illustrated how the interoperability issue can be addressed in this hypermedia digital library. Also, the integration of single and parallel analytical searching techniques through the integration of an IR and a CFP HA showed that the resulting HDL is extensible and flexible so it can integrate different methods which support information seeking activities.

9.3 Contribution to Knowledge

The Ph.D. research programme presented in this thesis is original, the author believes, in several aspects which can be summarised as follows.

Distributed Information Retrieval

1. The link-based collection fusion strategy for solving the collection fusion problem discussed in Chapter 5 represents the major original contribution to the area of Distributed Information Retrieval (DIR). The method to use linkage information to approximate the distribution of relevant documents in a hypermedia digital library is an original approach which has not been used before to solve the collection fusion problem.

2. Another contribution of this Ph.D. work is that it explicitly stated and investigated the link-hypothesis. This hypothesis has been implicitly used in a number of research works on hypermedia information retrieval, but the author is not aware of any work which explicitly stated and investigated the link-hypothesis.

3. The results which are produced from the user-centered experiments, represent an original and unique contribution to the debate of the usefulness of distributed, parallel searching and fusion strategies. Until now, the problem of parallel searching multiple libraries (i.e. fusion strategies) has been evaluated only in the laboratory. The user-centered experiment reported in Chapter 8, is to the best of the author's

knowledge, the only user-centered experiment which compared different distributed parallel searching strategies with single searching strategies. The results demonstrate that information seekers using parallel searching strategies are more effective and efficient that those using only single strategies and, therefore may influence the development of information seeking environments in the future.

Open Hypermedia Systems

1. The agent-based OHS architecture and the agent-communication language which are presented in Chapters 6 and 7, represent an original and novel approach to Open Hypermedia Systems and Hypermedia Digital Libraries. Although, the use of KQML is briefly mentioned in few published works on OHSs, the author is not aware of any published work which gives the breath and depth of using an OHS "content" language, KQML and software agents to solve the interoperability problem, as it is given in the work presented in this thesis. This work, the author believes, shows a path towards addressing interoperability issues in OHSs.

2. The agent-based interpretation of the Dexter model to design an OHS for HDLs is new and, the author believes, it is an approach illustrating novelty and originality. This widely used high level reference model was modified and extended with ideas and concepts that have been used before in other computer science disciplines such as software agents, CKBS, MAS. The result of the modifications and extensions was a Dexter-based distributed OHS which provides an enabling framework for developing hypermedia digital libraries.

9.4 Further Work

As a result of the research which has been carried out, a number of areas has been identified which would be interest to undertake further work.

As it was said in Chapter 7, due to limitations in resources the communication framework which is used, the distribution of hypermedia agents was confined to a local area network. It would be useful to extend this communication framework so hypermedia agents could be distributed over the Internet. A special software component called App-Link^{TM} was used in the prototype system to achieve communication between hypermedia agents. Unfortunately, in the time of developing the prototype OHS App-Link^{TM} could provide data exchange only in networks supporting NETBIOS. Plans have been reported, however, for a new version of

App-Link™ which will support TCP/IP. The author believes, that using this new version hypermedia agents can be easily distributed over the Internet.

Another area which requires further work is the examination of the proposed communication protocol using a wider range of OHSs and in a wider range of applications (e.g. environments for collaborative work). The application of the protocol in more OHSs and application areas will help to further develop the protocol and possibly to identify problems and interoperability needs which are not fully addressed in the current state.

Due to the limited number of available test collections which have linkage information, the link-based collection fusion strategy was evaluated using only two test collections. It would be extremely useful before any final conclusions could be drawn, to test the link-based fusion strategy in a wider range of test collections. Also, it would be useful to conduct further user-centered experiments. A point which was clear to the author during this Ph.D. programme, was the large amount of time and other resources which are required to conduct user-centered experiments. However, this type of experiment gives insights which are not available from system-centered experiments.

The last paragraph reveals a more general issue which the author believes needs work to be done by the research community: the development of test collections and evaluations methodologies specifically for hypermedia systems. Some reports have been published in the literature which suggest automatic methods to construct hypermedia test collections, but until now the author is not aware of any collection has been developed specifically for hypermedia. According to the author's opinion, the problem of lacking rigorous evaluation methodologies which mainly characterises hypermedia research, can be better addressed if standard test collections and methodologies are produced.

The author as a result of his desire to evaluate his work on hypermedia and, as a sideline of his research work, has proposed a novel approach to evaluate information seeking performance in hypermedia based on the structural analysis of hypermedia networks (Salampasis et al, in press). However, this is an entirely new evaluation methodology and therefore needs further work which will test the validity of the evaluation methodology itself and then its usefulness and expressiveness.

In the work that has been reported in Chapters 6 and 7 two hypermedia agents were integrated to the agent-based OHS: one for analytical searching and one for the collection fusion problem. Another area which the author wishes to undertake further work is to investigate the

integration of more hypermedia agents which implement other information seeking strategies such as assisted browsing or information filtering.

The collection fusion method is already applied in a prototype digital library for the extension training of Greek beekeepers (Batzios et al, 1997). The author aims to undertake further work in some of the areas which are outlined in this section and to publish the results in the future.

This section concludes the thesis. Perhaps, the most important result produced out of this Ph.D. work, is that there are substantial benefits by considering digital libraries both from an architectural and an information seeking perspective. Digital libraries are a research area where the results from multiple areas must be properly combined and synthesised in a probably unique way, in order to develop systems which effectively provide *full services*. Within this framework, the author believes, OHSs have an important role to play. They represent a philosophy for the development of large information systems which is suitable for the development of highly dynamic, interactive and heterogeneous electronic environments such as digital libraries.

Reference list

Akscyn R., McCracken D. and Yoder E. KMS: A Distributed Hypermedia system for managing knowledge in organisations. *Communications of the ACM*, 31, 7, pp. 820-835, 1988.

Allan J. Automatic hypertext construction. *Cornell University, department of Computer Science technical report TR95-1414*, February 1995.

Allan, J., Ballesteros, L., Callan, J., Croft, W.B. and Lu, Z. Recent Experiments with INQUERY. In *Proceedings of the Fourth TREC Retrieval Conference (TREC-4)*, 1996.

Anderson K., Taylor R. and Whitehead E. Chimera: Hypertext for heterogeneous software environments. In *proceedings of the second European Conference on Hypertext Technology (ECHT94*), Edinbrough, Scotland, pp. 94-107, 1994.

Anderson K. A Critique of the Open Hypermedia Protocol. In *proceedings of the 3rd Workshop on Open Hypermedia Systems*, Hypertext '97, Southampton, England, pp. 11-18, 1997a.

Anderson K. Integrating Open Hypermedia Systems with the World Wide Web. In *proceedings of the eight ACM conference on Hypertext*, Southampton, England, pp. 157-166, 1997b.

Armstrong R., Freitag D., Joachims T. and Mitchell T. Webwatcher: A Learning Apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Sources (SS-95-08)*, Stanford, CA, pp. 6-13, 1995.

Atkins D., Birmingham W., Durfee E., Glover E., et al. Toward Inquiry-Based Education Through Interacting Software Agents. *IEEE Computer*, 29, 6, pp. 69-78, 1996.

Balabanovic M. and Shoham Y. Learning Information Retrieval Agents: Experiments with Automated Web Browsing. In *AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Sources (SS-95-08)*, Stanford, CA, pp. 13-19, 1995.

Balasubramanian V. A Hypermedia Approach to Digital Libraries: Review of Research Issues. *SIGLINK Newsletter*, 4, 2, pp. 26-28, 1995.

Balasubramanian V., Bashian A. and Porcher D. A Large Scale Hypermedia Application using Document Management and Web Technologies. In *proceedings of the 8th ACM Conference on Hypertext*, Southampton, England, pp. 134-145, 1997.

Bapat A., Wasch J., Aberer K. and Haake J. HyperStorM: An extensible object-oriented hypermedia engine. In *proceedings of the seventh ACM conference on Hypertext*, Washington D.C, USA, pp. 203-214, 1996.

Barker P., Beacham N., Hudson S. and Meng C. Document Handling in an Electronic Oasis. *The New Review of Document and Text Management*, 1, 1, pp. 1-17, 1995.

Bates M. The design of browsing and berrypicking techniques for the on line search interface. *Online review*, 13, 5, 407-424, 1989.

Batzios C. Designing of Experimental procedure - Experimental Protocol. In *proceedings of the Educational Seminar on the protection of experimental animals - the use in the clinic research and diagnosis*, Thessaloniki, Greece, pp. 395-417, 1995.

Batzios C., Salampasis M., Liakos V., Tait J and Androulidakis S. A Hypermedia Digital Library for the Education and Extension Training of Greek Beekeepers. In *proceedings of the First European Conference for Information technology in Agriculture*, Copenhagen, Denmark, pp. 159-162, 1997.

Belkin N. Information Concepts for Information Science. *Journal of Documentation*, 34, 1, pp. 55-85, 1978.

Belkin N. Anomalous states of knowledge as a basis for information retrieval. *Canadian journal of Information Science*, 5, 1, pp. 133-143, 1980.

Belkin N. Ineffable concepts in information retrieval. In *Karen Sparck Jones (ed.) Information retrieval experiment*, Butterworths, London, 1981.

Berners-Lee T., Cailliau R., Groff J. and Pollermann B. World Wide Web: The information universe. *Electronic networking: Research, Applications and Police*, 2, 1, pp. 51-58, 1992.

Berners-Lee T., Cailliau R., Luotonen A., Nielsen H., et al. The World Wide Web. *Communications of the ACM*, 37, 8, pp. 76-82, 1994.

Bernstein M. An Apprentice That Discovers Hypertext Links. In *proceedings of the first European Conference on Hypertext (ECHT90)*, pp. 212-223, 1990.

Birmingham W. An Agent-Based Architecture for Digital Libraries. *D-Lib magazine*, July 1995.

Birmingham W., Durfee E., Mullen T. and Wellman M. The Distributed Agent Architecture of the University of Michigan Digital Library. In *AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Sources (SS-95-08)*, Stanford, CA, pp. 19-29, 1995.

Bobak A. *Distributed and multi-database systems*. Artech House, London, 1996.

Bond A. and Gasser L. An Analysis of problems and researchers in DAI. In *Bond A. and Gasser L. (ed.) Readings in Distributed Artificial Intelligence*, Morgan Kaufman, New York, pp. 3-35, 1988.

Bowman M., Danzig P., Manber U. and Schwartz M. Scalable Internet Resource Discovery: Research Problems and Approaches. *Communications of the ACM*, 37, 8, pp. 98-107, 1994.

Bruza P. Stratified Information Disclosure, a Synthesis between Hypermedia and Information Retrieval. Ph.D. thesis, University of Nijmegen, March 1993.

Bush V. As We May Think. *Atlantic Monthly*, pp. 101-108, July 1945.

Callan J., Lu Z. and Croft B. Searching Distributed Collections With Inference Networks. In *proceedings of the 18th annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Seattle, USA, pp. 21-28, 1995.

Carr L., Hollom R., Hall W. and Davis H. The Distributed Link Service: A Tool for Publishers, Authors and Readers. In *proceedings of the fourth international World Wide Web conference*, Boston, USA, pp. 647-665, 1995.

Charoenkitkarn N., Chingell M. and Golovchinsky G. Is Recall Relevant? An Analysis of How User Interfaces Conditions Affect Strategies and Performance in Large Scale Text Retrieval. In *proceedings of the fourth Text REtrrieval Conference (TREC-4)*, NIST 500-236, pp. 211-233, 1996.

Chiamarrela Y. and Kheirbek A. An integrated Model for Hypermedia and Information Retrieval. In *Agosti M. and Smeaton A. (ed.) Information Retrieval and Hypertext*, pp. 139-176, 1996.

Christel M., Kanade T., Mauldin M., Reddy R., et al. Informedia Digital Video Library. *Communications of the ACM*, 38, 4, pp. 57-58, 1995.

Cleary C. and Bareiss R. Practical Methods for Automating Linking in Structured Hypermedia Systems. In *proceedings of the Seventh ACM Conference on Hypertext*, Washington DC, USA, 1996.

Coad P. and Yourdon E. *Object Oriented Analysis*. Yourdon press, second edition, New York, 1991.

Conklin J. Hypertext: An Introduction and Survey. *IEEE Computer* 20, 9, pp. 17-41, 1987.

Crane G. Building a Digital Library: the Perseus Project as a Case Study in the Humanities. In *proceedings of the first ACM Int. conference on Digital Libraries*, Washington DC, USA, pp. 3-11, 1996.

Croft B. and Turtle H. A retrieval model for incorporating hypertext links. In *proceedings of the second ACM hypertext conference*, Pittsburgh, USA, pp. 213-224, 1989.

Croft B. Hypertext and Information retrieval: What are the Fundamental Concepts? In *First European Conference on Hypertext Technology (ECHT 90)*, INRIA, France, pp. 362-365, 1990.

Croft B. NFS Center for Intelligent Information Retrieval. *Communications of the ACM*, 38, 4, pp. 42-43, 1995.

Cutting D., Karger D., Pedersen J. and Tukey J. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *proceedings of the 15th Annual International ACM/SIGIR Conference*, Copenhagen, 1992.

Davies J., Weeks R. and Revett P. An Information Agent for the WWW. In *proceedings of the fourth International conference on World Wide Web*, Boston, USA, 1995.

Davies J., Weeks R., McGrath M., McGrath R., et al. In *proceedings of the 18th BCS IRSG Annual Colloquium on Information Retrieval Research*, Manchester, England, pp. 27-46, 1996.

Davis H., Hall W., Heath I., Hill G. and Wilkins R. Towards an Integrated Information Environment with Open Hypermedia Systems. In *proceedings of the second European Conference on Hypertext Technology (ECHT 92)*, Milano, Italy, pp. 181-190, 1992.

Davis H., Hutchings G. and Hall W. A Framework for Delivering Large-Scale Hypermedia Learning Material. In *Educational Multimedia and Hypermedia Annual, 1993: Proceedings of ED-MEDIA '93-World Conference on Educational Multimedia and Hypermedia*, Orlando, Florida, USA, pp. 115-122, 1993.

Davis H., Knight S. and Hall W. Light Hypermedia Link Services: A Study in Third Party Application Integration. In *proceedings of third European Conference on Hypertext (ECHT94*), Edinbrough, Scotland, pp. 41-50, 1994.

Davis H. *Data Integrity Problems in an Open Hypermedia Link Service*. Ph.D. thesis, Faculty of Engineering and Applied Sciences, Department of Electronic and Computer Science, University of Southampton, November 1995.

Davis H., Lewis A. and Rizk A. OHP: A Draft Proposal for an Open Hypermedia Protocol. *Draft proposal presented in the second workshop on Open Hypermedia Systems*, Washington DC, USA, 1996. (http://wwwcosm.ecs.soton.ac.uk/~hcd/protweb.htm)

Davis J. and Lagoze C. Drop-In Publishing With the World Wide Web. In *proceedings of the Second International Conference WWW'94*, Chicago, USA, pp. 749-759, 1994.

De Roure D., Hall W., Davis H. and Dale J. Agents for Distributed Multimedia Information Management. In *proceedings of PAAM '96 Conference*, London, England, 1996.

Deen S. Cooperating Agents - A Database Perspective. In *proceedings of the international working conference on Cooperating knowledge based systems (CKBS 90)*, Keele University, UK, pp. 3-29, 1990.

Delisle N. and Schwartz M. Neptune: A Hypertext system for CAD applications. In *proceedings of the ACM SIGMOD 86*, Washington DC, USA, pp. 132-142, 1986.

Dunlop M. & Rijsbergen C. J. Hypermedia and Free Text Retrieval. Information Processing & Management, 29, 3, pp. 287-298, 1993.

Dunlop M. (ed.). *Proceedings of the Second Mira Workshop*, Moncelice, Italy. University of Glasgow Computing Science Research report, TR-1997-2, 1997

Durfee E. Trends in Cooperative distributed problem solving. *IEEE TKDE*, 1, 3, pp. 63-83, 1989.

Ellman J. and Tait J. INTERNET Challenges for Information Retrieval. In *proceedings of the 18th BCS IRSG Annual Colloquium on Information Retrieval Research*, Manchester, England, pp. 1-12, 1996.

Engelbart D. Authorship provisions in Augment. In *Proceedings of the 1984 COMPCON conference (COMPCON '84)* Digest, San Francisco, CA, pp. 465-472, 1984.

Engelbart D. Knowledge Domain interoperability and an open hyper-document system. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW'90), Los Angeles, CA, pp. 143-156, 1990.

Finin T., Weber J., Wiederhold G. et al. Specification of the KQML Agent Communication Language. *The DARPA Knowledge Sharing Initiative, External Interfaces Working Group*, 1993.

Fountain A., Hall W., Heath I. and Davis H. MICROCOSM: An Open Model for Hypermedia With Dynamic Linking. In *proceedings of the first European Conference on Hypertext (ECHT90)*, INRIA, France, 1990.

Fox E. World Wide Web and Computer Science Reports. *Communications of the ACM*, 38, 4, pp. 43-44, 1995.

Fox E., Akscyn R., Furuta R. and Legget J. Digital Libraries. *Communications of the ACM*, 38, 4, pp. 23-28, 1995a.

Fox E., Barnette D., Shaffer C., Heath L., et al. Progress in Interactive Learning with a Digital Library in Computer Science, 1995b.

Frakes M. and Yates B. (ed.). *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, New York, 1992.

Frei P. and Stieger D. Making use of hypertext links when retrieving information. In *Proceedings of the second European Conference on Hypertext (ECHT 92)*, Milano, Italy, pp. 102-111, 1992.

French C., Fox A., Maly K. and Selman L. A Wide Area Technical Report Service: Technical Reports Online. *Communications of the ACM*, 38, 4, pp. 45, 1995.

Frisse M. Searching for Information in a hypertext medical book. *Communications of the ACM*, 31, 7, pp. 880-886, 1988.

Fuhr N. Optimum Selection in Networked IR. Paper presented in the networked Information retrieval workshop of the *19th annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Zurich, Switzerland, 1996.

Genesereth M. and Ketchpel S. Software Agents. *Communications of ACM*, 37, 7, pp. 48-53, 1994.

Golovchinsky G. What the Query Told the Link: The integration of hypertext and information retrieval. In *proceedings of the eight ACM conference on Hypertext*, Southampton, UK, pp. 67-74, 1997.

Goose S, Dale J, Hill G, De Roure D, Hall W. An Open Framework for Integrating Widely Distributed Hypermedia Resources. In *proceedings of the third IEEE conference on Multimedia Computing and Systems*, 1996.

Goose S., Dale J., Hall W. and De Roure D. Microcosm TNG: A Distributed Architecture to Support Reflexive Hypermedia Applications. In *proceedings of the eight ACM conference on Hypertext*, Southampton, UK, pp. 226-227, 1997a.

Goose S., Lewis A. and Davis H. OHRA: Towards an Open Hypermedia Reference Architecture and a Migration Path for existing Systems. In *proceedings of the 3rd Workshop on Open Hypermedia Systems*, Hypertext '97, Southampton, England, pp. 45-61, 1997b.

Gravano L., Garcia-Molina H. and Tomasic A. The effectiveness of GLOSS for the text database discovery problem. In *proceedings of the SIGMOD 94*, pp. 126-137, 1994.

Gronb? k K. and Trigg R. Design issues for a Dexter-based hypermedia system. *Communications of the CACM*, 37, 2, pp. 40-49, 1994.

Gronb? k K., Bouvin N. and Sloth L. Designing Dexter-based hypermedia services for the World Wide Web. In *proceedings of the eight ACM conference on Hypertext*, Southampton, UK, pp. 146-157, 1997.

Gronb? k K. and Wiil U. Towards a Reference Architecture for Open Hypermedia. In *proceedings of the 3rd Workshop on Open Hypermedia Systems*, Hypertext '97, Southampton, England, pp. 62-72, 1997.

Haan B., Kahn P., Riley V., Coombs J. et al. IRIS Hypermedia Services. *Communications of the ACM*, 35, 1, pp. 36-51, 1992.

Halasz F., Moran T. and Trigg R. Notecards in a Nutshell. In *proceedings of the ACM Conference on Human Factors in Computing Systems and Graphics Interfaces* (CHI+GI'87), pp. 45-52, 1987.

Halasz F. Reflections on NoteCards: Seven Issues for the next generation of Hypermedia Systems. *Communications of the ACM*, 31, 7, pp. 836-852, 1988.

Halasz F. and Schwartz M. The Dexter Hypertext Reference Model. In *proceedings of the Second Hypertext Standardisation Workshop (NIST)*, pp. 95-133, 1990.

Halasz F. Seven Issues Revisited. *Closing Plenary at the ACM Hypertext 91 conference*, San Antonio Texas, December 1991. (http://www.parc.xerox.com/spl/projects/halasz-keynote/)

Halasz F. and Schwartz M. The Dexter Hypertext Reference model. *Communications of the ACM*, 37, 2, pp. 30-40, 1994.

Hall W. Ending the Tyranny of the Button. *IEEE Multimedia*, pp. 60-68, 1994.

Hall W., Davis H. and Hutchings G. *Rethinking Hypermedia: The Microcosm Approach*. Kluwer Academics, Boston, 1996.

Hall W. and Davis H. A Southampton Scenario for OHS. In *proceedings of the 3rd Workshop on Open Hypermedia Systems*, Hypertext '97, Southampton, England, pp. 81-85, 1997.

Hardman L., Bulterman D. and Rossum G. The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37, 2, pp. 50-62, 1994.

Harman D. Overview of the Third Text REtrieval Conference (TREC-3). *In proceedings of the third TREC Conference* (NIST SP 500-225), pp. 1-21, 1994.

Harman D. Overview of the Fourth Text REtrieval Conference (TREC-4). In *proceedings of the fourth TREC conference* (NIST SP 500-236), pp. 1-34, 1996.

Hearst M. and Pedersen J. Re-examining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *proceedings of the 19th annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Zurich, Switzerland, pp. 76-84, 1996.

Heath L., Hix D., Nowell L. et al. Envision: A User-Centered Database of Computer Science Literature. *Communications of the ACM*, 38, 4, pp. 52-53, 1995.

Hendry D. and Harper D. An architecture for implementing extensible information-seeking environments. In *proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, 1996a.

Hendry D. and Harper D. An informal information-seeking environment. *Journal of the American Society for Information Science*, 1996b.

Hersh W., Elliot D., Hickam D. et al. Towards New Measures of Information Retrieval Evaluation. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, USA, pp. 164-170, 1995.

Hill G. and Hall W. Extending the Microcosm Model to a Distributed Environment. In *proceedings of third European Conference on Hypertext (ECHT94*), Edinbrough, Scotland, 1994.

Hill G., Wilkins R. and Hall W. Open and Reconfigurable Hypermedia Systems: A Filter Based Model. *Hypermedia*, 5, 2, 1993.

Hirata K., Mukherjea S., Okamura Y., Li W., et al. Object-based Navigation: An Intuitive Navigation Style for Content-oriented Integration Environment. In *proceedings of the eight ACM conference on Hypertext*, Southampton, UK, pp. 75-86, 1997.

Hopper A. The Network Computer. Keynote address in the *19th annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Zurich, Switzerland, pp. 1-2, 1996.

Huibers T., Lalmas M., and van Rijsbergen C. J. Information retrieval and Situation Theory. *Technical report UU-CS-1996-04, Department of Computer Science, Utrecht University*, the Netherlands, January 1996.

Jardine N. and Van Rijsbergen C. J. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7, pp. 217-240, 1971.

Jean F. Question Answering and Learning with Hypertext. In *IFIP transactions: Lessons from Learning*, 1994.

Kacmar J. and Legget J. PROXHY: A process-oriented extensible hypertext architecture. *ACM transactions on Information Systems*, 9, 4, pp. 399-419, 1991.

Kanji G. *100 Statistical tests*. SAGE Publications, London, 1994.

Kappe F. Hyper-G, A Universal Hypermedia System. *Journal of Educational Multimedia and Hypermedia*, 2, 1, pp. 39-66, 1993.

Kellog R. and Subhas M. Text to Hypertext: Can Clustering Solve the Problem in Digital Libraries. In *proceedings of the first ACM Int. conference on Digital Libraries*, Washington DC, USA, pp. 144-150, 1996.

Kolb D. Scholarly Hypertext: Self-Represented Complexity. In *proceedings of the eight ACM Conference on Hypertext*, Southampton, England, pp. 29-37, 1997.

Kraij W. And Pohlmann R. Viewing Stemming as Recall Enhancement. In *proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96*, Zurich, Switzerland, pp. 40-48, 19996.

Kwok K., Grunfeld L. and Lewis D. TREC-3 ad-hoc routing retrieval and thresholding experiments using PIRCS. *In proceedings of the third TREC Conference* (NIST SP 500-225), pp. 220-245, 1994.

Lagoze C. and Davis J. Dienst: An Architecture for Distributed Document Libraries. *Communications of the ACM*, 38, 4, pp. 47-48, 1995.

Labrou Y. and Finin T. Proposal for a new KQML Specification. *Computer Science and Electrical Engineering Department technical report TR CS-97-03*, University of Maryland, February 1997.

Lee J. Combining Multiple Evidence from Different Properties of Weighted Schemes. In *proceedings of the 18th annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Zurich, Switzerland, pp. 180-188, 1995.

Legget J. and Schnase J. Viewing Dexter with open eyes. *Communications of the ACM*, 37, 2, pp. 145-166, 1994.

Lewis P., Davis H., Griffiths S., Hall W., et al. Media-based Navigation with Generic Links. In *proceedings of the seventh ACM conference on Hypertext*, Washington DC, USA, pp. 215-223, 1996.

Li Z., Davis H. and Hall W. Hypermedia Links and Information Retrieval. In *proceedings of the 14th BCS IR-SG research colloquium on Information retrieval*, 1992.

Lieberman H. Letizia: An Agent that Assists Web Browsing. In *proceedings of AI Applications in Knowledge Navigation & Retrieval*, 1995 AAAI Fall Symposium (FS-95-03), pp. 97-102, 1995.

Longstaff J, Duncan R, Jennings N and Salampasis M. The Identification and Modelling of Co-operating Agents. In *proceedings of the Second Singapore International Conference on Intelligent Systems*, Singapore, 1994.

Lucarrela D. and Zanzi A. Information Modelling and retrieval in Hypermedia Systems. In *Agosti M. and Smeaton A. (ed.) Information retrieval and hypertext*, pp. 121-135, 1996.

Lynch C. The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present and Future. *D-lib magazine*, April 1997.

Magavi S., Wong J. and Bodla P. Design and Implementation of Heterogeneous Distributed Multimedia System Using Mosaic GSQL. *Software Practice and Experience*, 25, 11, pp. 1223-1241, 1995.

Malcolm K., Poltrock S. and Schuler D. Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise. In *proceedings of the 4th ACM Hypertext 91 conference*, San Antonio, Texas, pp. 13-25, 1991.

Maly K., French J., Selman A. and Fox E. Wide Area Technical Report Service. In *proceedings of the Second International World Wide Web Conference (WWW'94)*, Chicago, USA, pp. 523-533, 1994.

Marchionini G., Schneiderman B. Finding facts vs. browsing knowledge in hypertext systems. *IEEE computer*, 21, 1, pp. 70-80, 1988.

Marchionini G. Information seeking strategies of novices using a full-text electronic encyclopaedia. *Journal of the American Society for Information Science*, 29, 3, pp. 165-176, 1989.

Marchionini G. and Maurer F. The Roles of Digital Libraries in Teaching and Learning. *Communications of the ACM*, 38, 4, pp. 67-75, 1995.

Marchionini G. *Information Seeking in Electronic Environments*. Cambridge University Press, New York, 1995.

Marcus R. An Experimental comparison of the effectiveness of computers and humans as search intermediaries. *Journal of the American Society for Information Science*, 34, pp. 381-404, 1983.

Marsh S. A Community of Autonomous Agents for the Search and Distribution of Information in Networks: A Preliminary report. In *proceedings of the 19th Annual BCS-IRSG Colloquium on IR research*, Aberdeen, Scotland, pp. 108-124, 1997.

Meyrowitz N. The missing Link: Why we're all doing hypermedia wrong. In *Barret E. (ed.) The Society of Text*, MIT press, Cambridge MA, pp. 107-114, 1989.

Moffat A. and Zobel J. Information retrieval systems for large document collection. In *proceedings of the third Text Retrieval Conference (TREC-3)*, 1994.

Nelson T. Replacing the Printed Word: A Complete Literary System. In *Proceedings of the IFIP Congress*, pp. 1013-1023, 1980

Nwana H. Software Agents: An Overview. *Knowledge Engineering Review*, 11, 3, pp. 1-40, 1996.

Nurberg P., Furuta R., Leggett J., Marshall C., et al. Digital Libraries: Issues and Architectures. In *proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries* (DL95), Austin, Texas, USA, 1995.

Nurberg P. and Legget J. And now for the tricky part: broadening the applicability of open hypermedia systems. In *proceedings of the 3rd Workshop on Open Hypermedia Systems*, Hypertext '97, Southampton, England, pp. 93-95, 1997.

Osterbye K. and Wiil U. The Flag Taxonomy of Open Hypermedia Systems. In *proceedings of the seventh ACM conference on Hypertext*, Washington D.C, USA, pp. 129-139, 1996.

Paepcke A., Cousins S., Garcia-Molina H., Hassan S., et al. Using Distributed Objects for Digital Library Interoperability. *IEEE Computer*, 30, 5, pp. 61-68, 1996.

Payette S. and Rieger O. Z39.50: The User's Perspective. *D-lib magazine*, April 1997.

Pearl A. Sun's Link Service: A protocol for Open Linking. In *proceedings of second ACM conference on Hypertext*, Pittsburgh, USA, pp. 137-147, 1989.

Rada R. and Murphy C. Searching versus Browsing in Hypertext. *Hypermedia*, 4, 1, pp. 1-10, 1992.

Rao R., Pedersen J., Hearst M., Mackinlay J., et al. Rich Interaction in the Digital Library. *Communications of the ACM*, 38, 4, pp. 29-39, 1995.

Rearick T. Automating the conversion of text into Hypertext. In *Berk E. and Devlin J. (ed.) Hypertext/Hypermedia Handbook*, pp. 1133-1140, 1991.

Rizk A. and Sauter L. Multicard: An Open Hypermedia System. In *proceedings of the second European Conference on Hypertext Technology (ECHT92)*, Milano, Italy, pp. 4-10, 1992.

Robertson S. The probability ranking principle in IR. *Journal of Documentation*, 33, pp. 294-303, 1977.

Robertson S. The methodology of Information retrieval experiment. In *Karen Sparck Jones (ed.) Information retrieval experiment*, Butterworths, London, 1981.

Rusbridge C. The UK Electronics Libraries Programme. *D-lib magazine*, December 1995.

Sairamesh J., Nikolaou C., Ferguson F., and Yemini Y. Economic Framework for Pricing and Charging in Digital Libraries. *D-lib magazine*, February 1996.

Salampasis M & Tait J. HyperTree: An Alternative Approach To Web Authoring. *University of Sunderland, School of Computing & Information Systems, Occasional paper 95-8*, August 1995.

Salampasis M., Tait J. & Bloor C. (in press). Evaluation Of Information Seeking Performance in Hypermedia Digital Libraries. To appear in *Interacting with Computers*.

Salampasis M., Tait J. & Hardy C. (in press). HyperTree: A Structural Approach to Web Authoring. To appear in *Software Practice and Experience*.

Salton G. Automatic Indexing Using Bibliographic Citations. *Journal of Documentation*, 27, 2, pp. 98-110, 1971.

Salton G., Yang C. S. and Wong A. A Vector space Model for Automatic Indexing. *Communications of the ACM, 18*, 6, pp. 613-620, 1975.

Salton G., Fox E. and Wu U. Extended Boolean Information Retrieval. *Communications of the ACM*, 26, 12, pp. 1022-1036, 1983.

Salton G. and Buckley C. Term Weighting approaches in automatic text retrieval. *Information Processing & Management*, 24, 5, pp. 513-523, 1988.

Saracevic T. Evaluation of Evaluation in Information Retrieval. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, USA, pp. 138-146, 1995.

Savoy J. An extended Vector Processing Scheme for Searching Information in Hypertext Systems. *Information Processing and Management*, 32, 2, pp. 155-170, 1996.

Schatz B. Building the Interspace: The Illinois Digital Library Project. *Communications of the ACM*, 38, 4, pp. 62-63, 1995.

Schatz B. and Chen H. Building Large Scale Digital Libraries. *IEEE Computer*, 30, 5, pp. 22-28, 1996.

Schatz B., Mischo W., Cole T., Hardin J., et al. Federating Diverse Collections of Scientific Literature. *IEEE Computer*, 30, 5, pp. 28-36, 1996.

Schutt H. and Streitz A. Hyperbase: A Hypermedia Engine Based on a Relational Database Management System. In *proceedings of the First European Conference on Hypertext (ECHT90)*, Versailles, France, pp. 95-108, 1990.

Shackelford D. Smith J. and Smith D. The Architecture and Implementation of a Distributed Hypermedia Storage System. In *proceedings of fifth ACM conference on Hypertext*, Seattle, USA, pp. 1-13, 1993.

Smith J. and Smith F. ABC: A hypermedia system for artefact-based collaboration. In *proceedings of second ACM conference on Hypertext*, San Antonio, USA, pp. 179-192, 1991.

Smith J. The King is Dead; Long Live the King. Keynote Address at *the eight ACM conference on Hypertext*, Southampton, England, pp. 240, 1997.

Smith T. and Frew J. Alexandria Digital Library. *Communications of the ACM*, 38, 4, pp. 61-62, 1995.

Sparck Jones K. Introduction to IR experiment. In *Karen Sparck Jones (ed.) Information Retrieval Experiment*, Butterworths, London, 1981.

Sparck Jones, K. Reflections on TREC. *Information Processing and Management*, 31, 3, pp. 291-314, 1995.

Stein R. Browsing through terabytes-wide area information servers open a new frontier in personal and corporate information services. *Byte*, 16, 5, pp. 177-187, 1991.

Stubenrauch R., Kappe F., Andrews K. Large Hypermedia Systems: The End of the Authoring Era. In *Proceedings of ED-MEDIA '93-World Conference on Educational Multimedia and Hypermedia*, Orlando, Florida, USA, 1993.

Turtle H and Croft B. Evaluation of an Inference Network-based Retrieval Model. *ACM Transactions on Information Systems*, 9, 3, pp. 187-222.

Van Rijsbergen C. J. A non-classical logic for information retrieval. *The Computer Journal*, 29, 6, pp. 481-485, 1986

Van Rijsbergen C. J. *Information Retrieval*. Butterworths, London, second edition, 1979.

VanHeyningen M. The Unified Computer Science Technical Report Index: Lessons in indexing diverse resources. In *proceedings of the Second World Wide Web International Conference* WWW'94, Chicago, USA, pp. 535-543, 1994.

Vanzyl A. Open Hypermedia Systems, Comparisons and Suggestions for Implementation strategies. In *Proceedings of the ECHT 94 Workshop on Open Hypermedia Systems,* Edinbrough, Scotland, pp. 11-15, 1994.

Vickery & Vickery

Viles C. and French J. Dissemination of Collection Wide Information in a Distributed Information retrieval system. In *proceedings of the 18th annual International ACM SIGIR Conference on Research and Development in Information retrieval,* Zurich, Switzerland, pp. 12-20, 1995.

Viles C. Maintaining State in a Distributed Information Retrieval System. In *proceedings of the 32nd ACM South east conference,* Tuscaloosa, pp. 157-161, March 1994.

Voorhees H. Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval. Cornell University, Computer Science Department technical report TR86-765, March 1986.

Voorhees E., Gupta N., Johnson-Laird B. The Collection Fusion Problem. In *proceedings of the third Text Retrieval Conference (TREC-3),* NIST publication 500-225, pp. 95-105, November 1994.

Voorhees E., Gupta N., Johnson-Laird B. Learning Collection Fusion Strategies. In *proceedings of the 18th annual International ACM SIGIR Conference on Research and Development in Information retrieval,* pp. 172-179, 1995.

Voorhees E. Siemens TREC-4 Report: Further Experiments with Database Merging. In *proceedings of the fourth Text Retrieval Conference* (TREC-4). NIST publication 500-236, 1996.

Wactlar H., Kanade H., Smith M., Stevens S. Intelligent Access to Digital Video: Informedia Project. *IEEE Computer,* 30, 5, pp. 46-52, 1996.

Weiss R., Velez B., Sheldon M., Manprempre C. et al. HyPursuit: A hierarchical Network Search Engine that Exploits Content-Link Hypertext Similarity. In *proceedings of the seventh ACM conference on Hypertext,* Washington DC., USA, pp. 180-194, 1996.

Whitehead J. An Architectural Model for Application Integration in Open Hypermedia Environments. In *proceedings of the eighth ACM conference on Hypertext*, Southampton, England, pp. 1-12, 1997.

Wiederhold G. Digital Libraries, Value and Productivity. *Communications of the ACM*, 38, 4, pp. 85-96, 1995.

Wiesener S., Kowarschick W. and Bayer R. SemLink: An Approach for Semantic Browsing through Large Distributed Document Spaces. In *proceedings of the third forum on Advanced Digital Libraries (ADL 96)*. Library of Congress, Washington D.C, USA, pp. 86-95, 1996.

Wiil U. and Legget J. HyperForm: Using Extensibility to Develop Dynamic, Open and Distributed Hypertext Systems. In *proceedings of the fourth ACM conference on Hypertext (ECHT92*), Milano, Italy, pp. 251-261, 1992.

Wiil U. and Osterbye K. *Proceedings of the ECHT 94 Workshop on Open Hypermedia Systems*. Aalborg University, Department of Mathematics and Computer Science technical report, R-94-2038, October 1994.

Wiil U. K., Legget J.J. The HyperDisco Approach to open hypermedia systems. In *proceedings of the 7th ACM Conference on Hypertext*, pp. 140-148, 1996.

Wiil U. and Whitehead E. Interoperability and Open Hypermedia Systems. In *proceedings of the 3rd Workshop on Open Hypermedia Systems*, Southampton, England, pp. 137-145, April 1997.

Wiil U. and Legget J. HyperDisco: Collaborative Authoring and Internet Distribution. In *proceedings of the eight ACM conference on Hypertext*, Southampton, UK, pp. 13-24, 1997.

Wildemuth B. Defining Search Success: Evaluation of Searcher Performance in Digital Libraries. *SIGOIS Bulletin*, 16, 2, pp. 29-32, 1995.

Wilensky R. Toward Work-Centered Digital Information Services. *IEEE Computer*, 5, pp. 37-44, 1996.

Wilensky R. UC Berkeley's Digital Library Project. *Communications of the ACM*, 38, 4, pp. 60, 1995.

Winograd T. Digital Libraries: Bridging the Two Cultures. Keynote address in *the 18th annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Zurich, Switzerland, pp. 1, 1995.

Witten I., Cunningham S., Vallabh M. and Bell T. A New Zealand Digital Library for Computer Science Research. In *proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries (DL95)*, Austin, Texas, USA, 1995.

Wooldridge M. and Jennings N. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review Volume*, 10, 2, 1995.

Glossary

ACQUAINTANCE_CIRCLE. The list of other agents that a hypermedia agent is able to communicate with.

Agent Body. The part of a VHA file which stores information about the contents which represents.

Agent Head. The part of a VHA having information which is independent of the information object that the VHA represents.

Anomalous State of Knowledge (ASK). An Information theory based on the concept of an anomalous state of knowledge.

CDPS. Cooperative Distributed Problem Solving.

CKBS. Cooperative Knowledge Based Systems.

Cluster hypothesis. *A* hypothesis stated as: "closely associated (in content) documents tend to be relevant to the same information needs (requests)".

Collection Fusion Problem (CFP). The problem in which the results of query runs in different, autonomous and distributed document collections must be merged to produce a single, effective result.

Digital Library (DL). A dynamic, highly interactive and distributed electronic information seeking environment.

Effectiveness. The factor indicating the degree of success that a user has in finding all or some of the required information.

Efficiency. The factor deciding the level of success that a system has in providing information to users in a certain amount of time, using the minimum level of resources, and, with a fair degree of user effort.

Facilitator. The agent playing a special communication role in federated architectures.

Federated communication architecture. The communication architecture in which agents do not communicate directly, but only through facilitators.

Hypermedia Agent Content Language (HACL). The "content" language used by HAs in the agent-based OHS architecture.

Hypermedia Agent Protocol (HAP). The protocol which is used between hypermedia agents to communicate each.

Hypermedia Agents (HAs). Hypermedia agents are software agents exchanging messages using the commonly agreed hypermedia agent protocol (HAP).

Hypermedia Digital Library (HDL). A digital library based on a hypermedia paradigm.

Integrated merging strategies. The fusion strategies which have access to additional information (e.g. collection wide word frequencies) in order to merge the results from multiple document collections.

Interoperability. The ability that individual tools, or components of a system, or whole systems may have to communicate, agree and coordinate in mutually providing services, handling subtasks, or achieving sub-goals.

Isolated merging strategies. The fusion strategies producing the single merged result without using any run-time information from remote collections except the ranked list of documents returned from individual collections.

Knowledge Query and Manipulation Language (KQML). A general-purpose language for communication between software agents.

Link-hypothesis: "closely interlinked documents tend to be relevant to the same information needs".

MAS. Multi Agent Systems.

Maximisation function. The algorithm approximating the distribution of relevant documents in other remote collections.

MESSAGE_LIST. The message list which keeps messages from/to other agents.

NIKOS. An agent-based Open Hypermedia System.

Ontology. A set of formal specifications about a concept, model, object etc.

Open Hypermedia Systems (OHS). Hypermedia systems purposefully designed to satisfy architectural requirements such as openness, extensibility and scalability. Their main goal is to deliver hypermedia functionality in existing information environments in an open manner.

Performative. A message type in KQML that express an attitude regarding the content of the exchange.

Personal Digital Library (PDL). A digital library considered at the lowest level of granularity. A PDL is the collection of first or higher class information objects (e.g. data and indexes) which are produced by or belong to an individual, and the data models, tools, methods and resources used by this individual to manage and share his/her personal information workplace.

Sampling collection. The collection initially used to extract linkage data in the link-based fusion strategy.

Software agents. Autonomous programs that can achieve a goal on behalf of a user or on behalf of another agent.

System-centered evaluation. Evaluations which do not involve users directly.

TREC (Text Retrieval Evaluation Conferences). Conferences dedicated to evaluation of information retrieval systems.

Virtual Hypermedia Agent Markup Language (VHAML). The markup language used in the prototype agent-based OHS to create VHAs.

Virtual Hypermedia Agents (VHAs). Virtual Hypermedia Agents are files storing meta-data for different purposes.

Virtual Knowledge Base (VKB). The set of all VHAs known to a particular Hypermedia Agent.

Appendix A

Author's list of publications during the Ph.D. programme

1. Longstaff J, Duncan R, Jennings N and Salampasis M. The Identification and Modelling of Cooperating Agents. In *proceedings of the Second Singapore International Conference on Intelligent Systems*, Singapore, November 1994
2. Salampasis M. Hypermedia: An Introduction and Survey. *University of Sunderland, School of Computing & Information Systems, Occasional paper 95-5*, June 1995.
3. Salampasis M. and Tait J. HyperTree: An Alternative Approach To Web Authoring. *University of Sunderland, School of Computing & Information Systems, Occasional paper 95-8*, August 1995.
4. Salampasis M. An Agent-Based Hypermedia Model. *Presented at the Hypertext 96 Doctoral Consortium*, Washington D.C, USA, March 1996.
5. Salampasis M., Tait J. and Hardy C. An Agent-Based Hypermedia Framework for Designing and Developing Digital Libraries. In *proceedings of the third forum on Advanced Digital Libraries (ADL 96)*, Library of Congress, Washington D.C, USA, pp. 1-10, May 1996.
6. Salampasis M. Tait J. and Bloor C. Cooperative Information Retrieval in Digital Libraries In *proceedings of the 18th annual colloquium of the BCS IR SG*, Manchester, UK, pp. 13-27, March 1996.
7. Salampasis M. and Tait J. Problems and Issues in Evaluation of Networked IR. In *Proceedings of the first Glasgow HCI & IR workshop, GIST Technical Report G96-2*, Glasgow University, September 1996.
8. Batzios C., Salampasis M., Liakos V et al. A Hypermedia Digital Library for the Education and Extension Training of Greek beekeepers. In *proceedings of the first European Conference on Information Technology in Agriculture*, Copenhagen, Denmark, pp. 159-163, June 1997.

9. Salampasis M. Modelling Open & Extensible Hypermedia Digital Libraries as a Society of Cooperating Agents. In *proceedings of the third workshop on Open Hypermedia Systems*, Hypertext 97, Southampton, UK, pp. 116-125, April 1997.

10. Salampasis M., Tait J. and Bloor C. (in press). Evaluating the Information Seeking Performance in Hypermedia Digital Libraries. To appear in the journal *Interacting with Computers*.

11. Salampasis M., Tait J. & Hardy C. (in press). HyperTree: A More Structural and Effective Approach to Web Authoring. To appear in the journal *Software Practice & Experience.*

12. Salampasis M. Towards A New Collection Fusion Strategy. *University of Sunderland, School of Computing & Information System, Occasional paper 97-4*, April 1997.

Appendix B

Document explaining the WWW-based HDLs

1. Introduction

This document describes the experimental environment in which the study you have been asked to participate is going to take place. Please read this document carefully and make sure that you can understand all the aspects of the information seeking environment (i.e. the WWW-based CACM digital library).

IMPORTANT

It is essential to have a basic understanding of the experimental environment, the organisation of the digital library, the methods which are available for searching the digital library and the layout of the documents. You can achieve this if you access and search the CACM digital library for about 15-20 minutes. The CACM digital library is accessible through: **http://osiris.sund.ac.uk/~cs0msa/cacm/cacmmain.htm** .

If you have any questions asked them before the experiment starts.

2. The Experimental Environment

The experimental environment is a World Wide Web-based digital library implementation of the CACM test collection. This is a collection of scientific documents/papers published in 60's and 70's in the computer science journal "Communications of ACM" (Association for Computing Machinery).

The CACM test collection is a standard document collection which is been used to perform experiments in Information Retrieval. It contains 3204 documents. Each record/document includes the title of the document, the author, abstract and associated keywords. Additionally some documents (1751 in precise) have references, cocitations and bibliographic couplings (these relationships will be explained later in detail) to other documents.

We have constructed a hypermedia network exploiting the links that CACM documents have to other CACM documents. This hypermedia network, or in our terms hypermedia digital library, comprises of 1751 CACM documents (out of the total 3204).

NOTE

Our experimental environment (i.e. the hypermedia digital library) comprises of **1751 CACM documents**.

These 1751 documents have been clustered based on their content similarity in 8 different distributed clusters/sub-libraries. Each sub-library X is autonomous to the rest seven sub-

libraries. **For example, when you make a query-based search in library X, only the documents which belong to library X will be considered and compared with your query** Figure 1 depicts how the CACM documents are allocated in the 8 sub-libraries

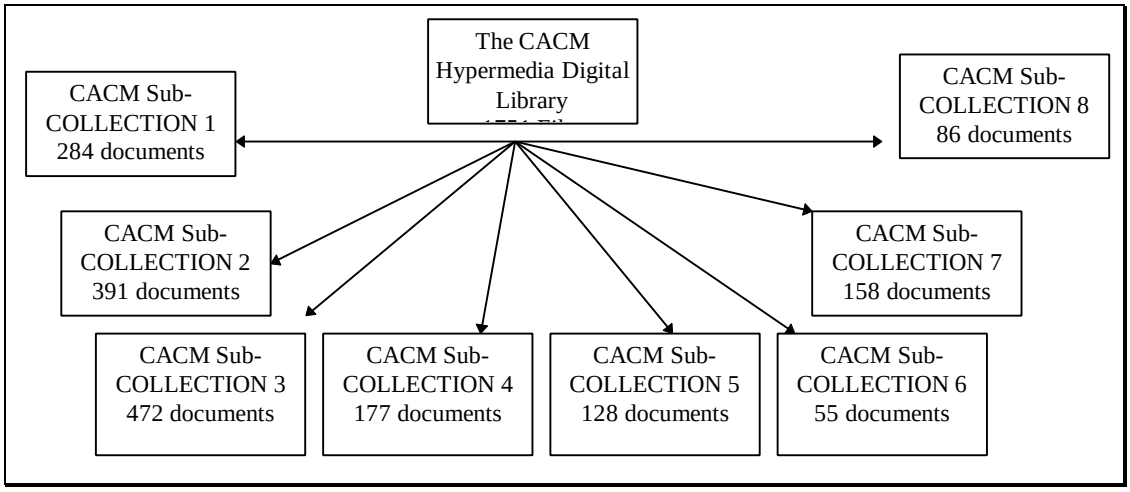


Figure 1. The Distributed CACM Hypermedia Digital Library

NOTE

The WWW-based CACM digital library is clustered in 8 autonomous and distributed digital sub-libraries. The distribution was based on the documents' content similarity. Therefore, it is likely (without this being a strict rule) that "similar" in content documents will be in the same library.

NOTE

The digital sub-libraries are autonomous. When you make a query-based search to sub-library only the documents which belong to this sub-library are being considered and compared with your query.

NOTE

Each document in our digital library has links to other (semantically) interrelated documents. Three different types of links exist (i.e. references, bibliographic couplings and co-citations). You can **browse** and navigate the CACM hypermedia digital library using these links. It is possible for a link to have its starting point in a document in a sub-library X and its endpoint in a document in a different sub-library Y.

NOTE

There are three different types of links between documents. Figure 2. presents and explains the three different link types.

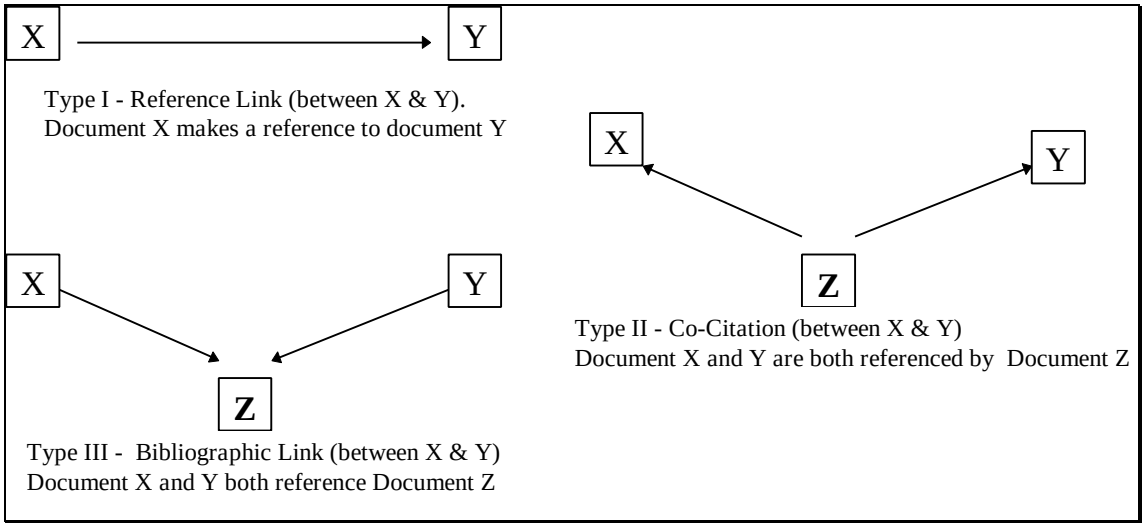


Figure 2. Link Types between two CACM documents X and Y

Reference Links usually represent a strong semantic relation between two documents. Although many reasons exist for making a reference from a document X to another document Y (i.e. Y provides an wider explanation of an issue, provides complementary information, proves an argument etc.), usually when a document X makes a reference to a document Y, it is likely that X and Y are on the same subject, and probably they are both relevant to a given an information need.

Therefore, when a user finds a document X which is relevant to an information need, it is likely that if he/she follows the reference links which have X as their starting point, will find more relevant documents to the same information need.

Quite similar, cocitations and bibliographic couplings express an interrelation between two documents, but which is usually less stronger.

NOTE

If you find a document X which is relevant to a query Q, you can follow the links starting from document X to find more relevant documents to query Q.

3. The World Wide Web-Based CACM Distributed Digital Library

The 1751 CACM documents have been published onto the web. This experimental study needs your help to evaluate the effectiveness of this WWW-based digital library in supporting users to find relevant information.

Figure 3. shows the home page of the Distributed WWW CACM digital library

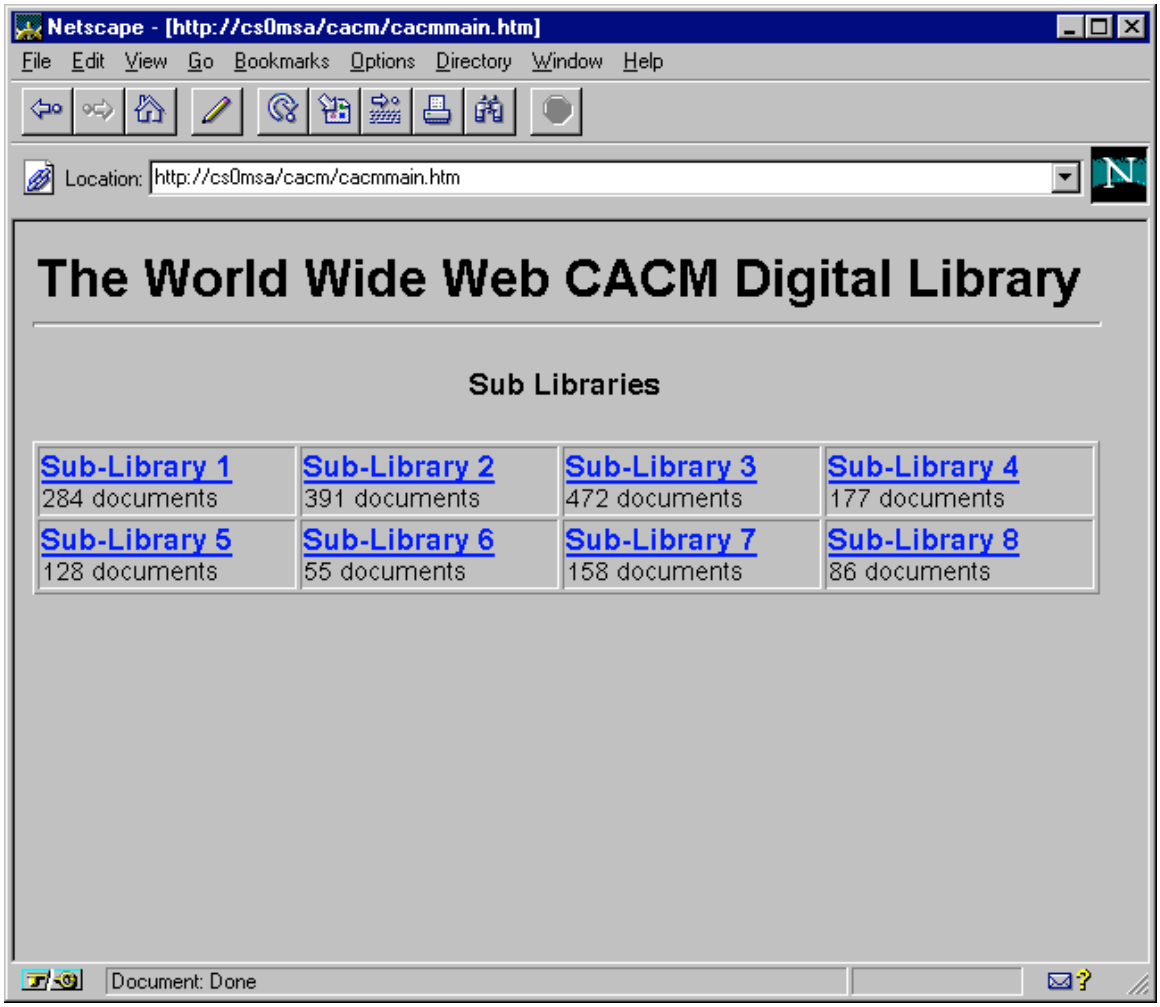


Figure 3. The home page of the WWW-based CACM digital library

NOTE

Starting from this home page you can move to the home page (see figure 4.) of each individual sub-library **X**.

4. How Can I Search the CACM digital library ?

There are two different ways in which you can search in the library:

1. By browsing i.e. by following links from one document to another document. There are three different types of links which one can activate:

   - **Reference links** or simply links which indicate that a document X directly cites (makes a reference) to document Y.
   - **Co-citation links** which means that exists document Z which **cites both X and Y** in the collection**.**
   - **Bibliographic couplings** that means that X and Y have common references in the bibliography.

NOTE

Because Reference links (or simply links) express a direct relationship between documents usually are better links for browsing.

2. By searching the sub-libraries using queries (e.g. similar to searching the web with a search engine e.g. Altavista, Lycos etc.).

In query-based searching you can type any terms you believe express better your information need. For example for the information need below:

" I'm interested in mechanisms for communicating between disjoint processes, possibly, but not exclusively, in a distributed environment. I would rather see descriptions of complete mechanisms, with or without implementations, as opposed to theoretical work on the abstract problem. Remote procedure calls and message-passing are examples of my interests. "

You can use this as your query:

" communicating disjoint processes distributed environment complete mechanisms remote procedure calls message passing "

Another example, if you are looking for:

" Interested in articles on semiology and politics. "

you can formulate this query:

" semiology and politics ".

NOTE

There is no limit in the terms that you can use in a query. However, the less terms you are using and more representative they are, you have more changes to retrieve relevant documents.

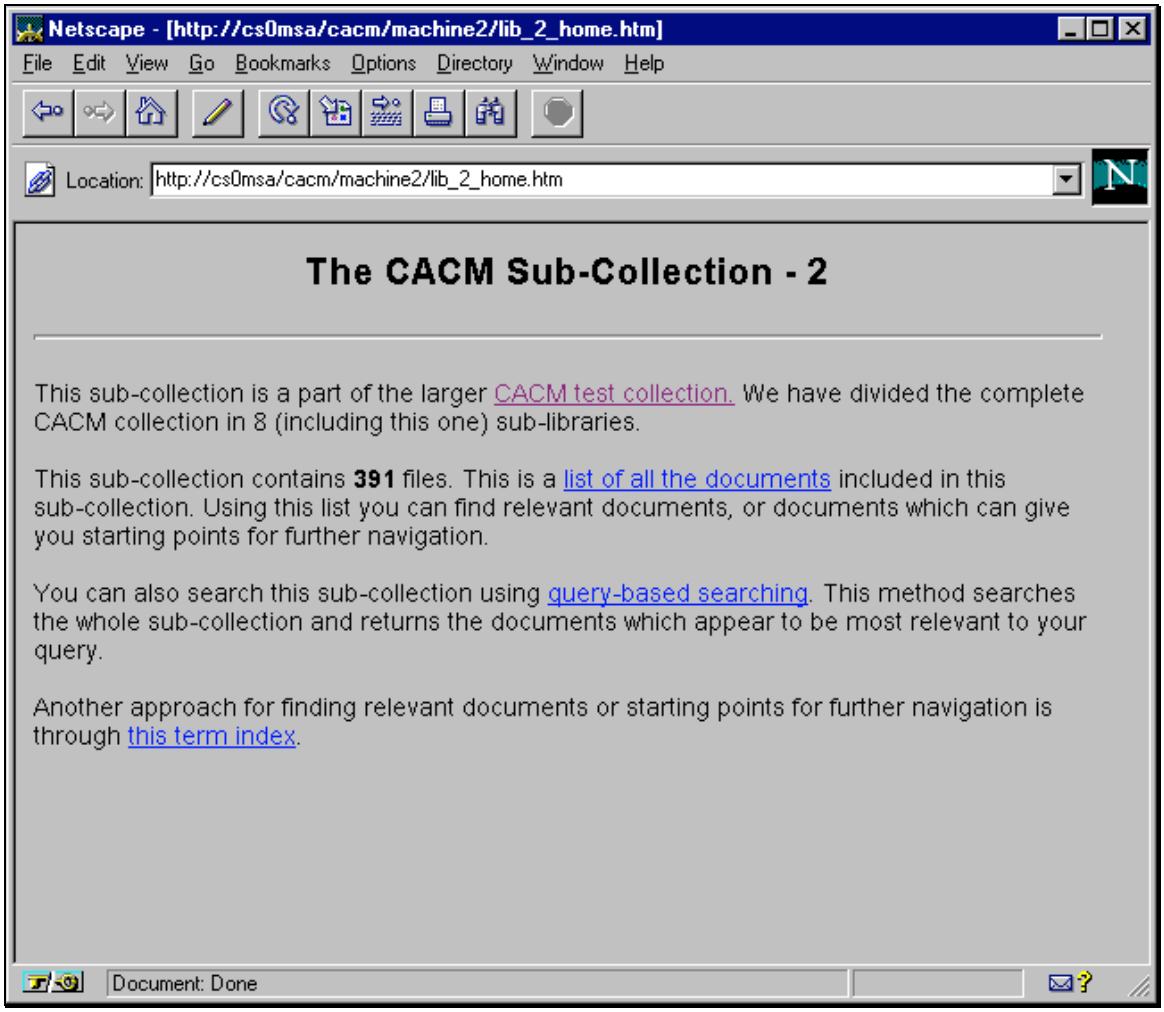


Figure 4. Home page of Sub-Library 2

Figure 4. presents the home page of a sub-library X and explanations of the links starting from this page. All the sub-libraries have similar home pages.

From the home page of a sub-library **X** a user has access to a list of all the documents which are included in sub-library **X** together with their titles (figure 5). You can also have access to a page where you can make a query **to sub-library X** (figure 6).

4.1 List of documents page

From the home page of each sub-library you can open the "list page" (link B in figure 4). This page is a list of all the documents which are included in sub-library X. In this list you can find the filename and the title of the document. All the documents in the collection are named based on a common naming convention. The first digit (these between the underscores) specifies to which sub-library this document belongs. Figure 5. presents the layout of a list of documents page.

NOTE

In the list page (figure 5) all the documents which are members of a sub-library are listed together with their titles. You can look through this list and identify documents which are relevant or documents which can be used as starting points for your navigation.

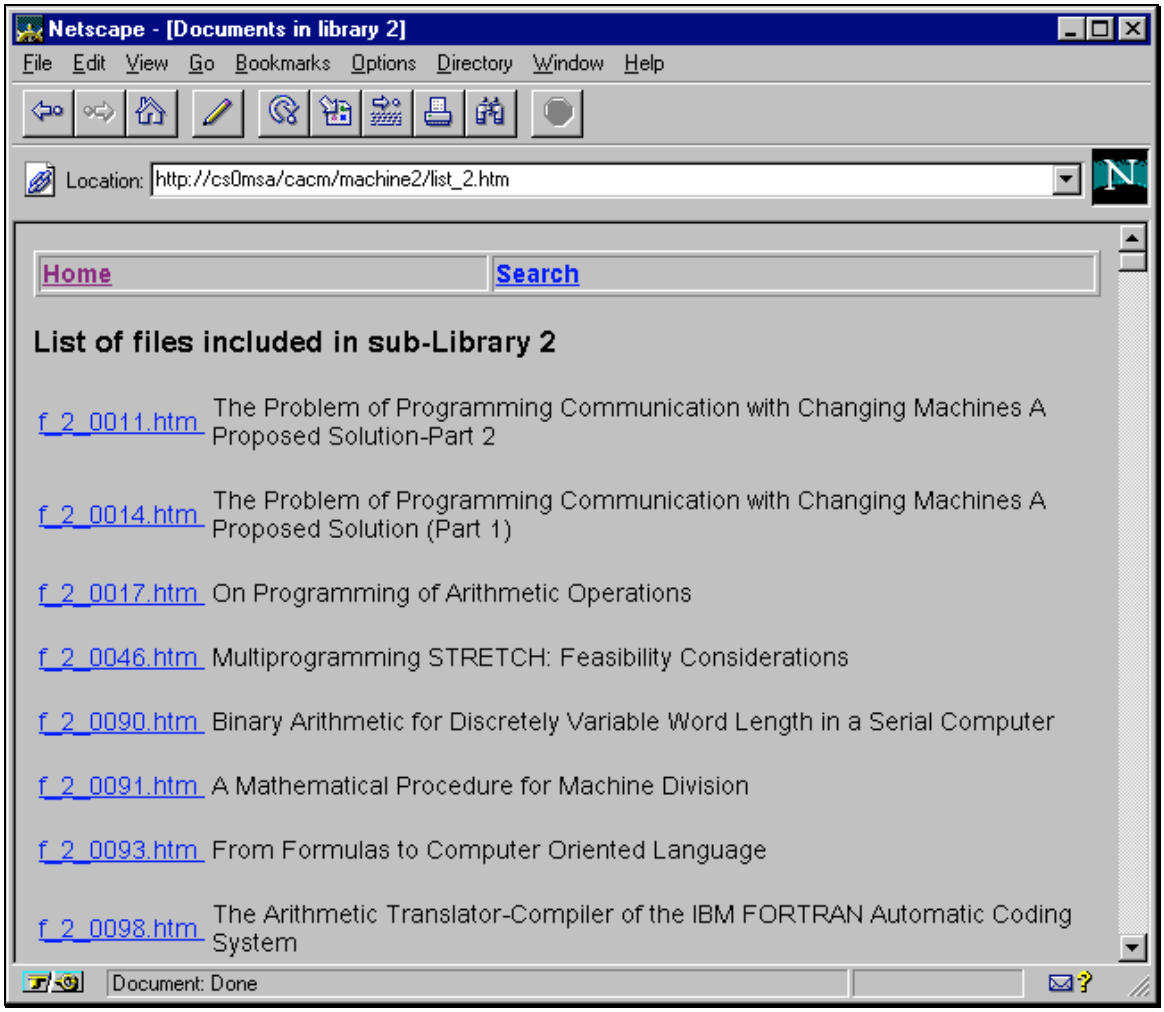


Figure 5. An example of a "list of Documents" page

NOTE

The 1751 files are named so someone can easily understand to which sub-library they belong to. The naming convention is f_X_YYYY.htm where X indicates the sub-library and YYYY is a code given by the constructors of the CACM collection.

4.2 Query page

Figure 6. presents the page which someone can use to make a query-based search to a sub-collection.

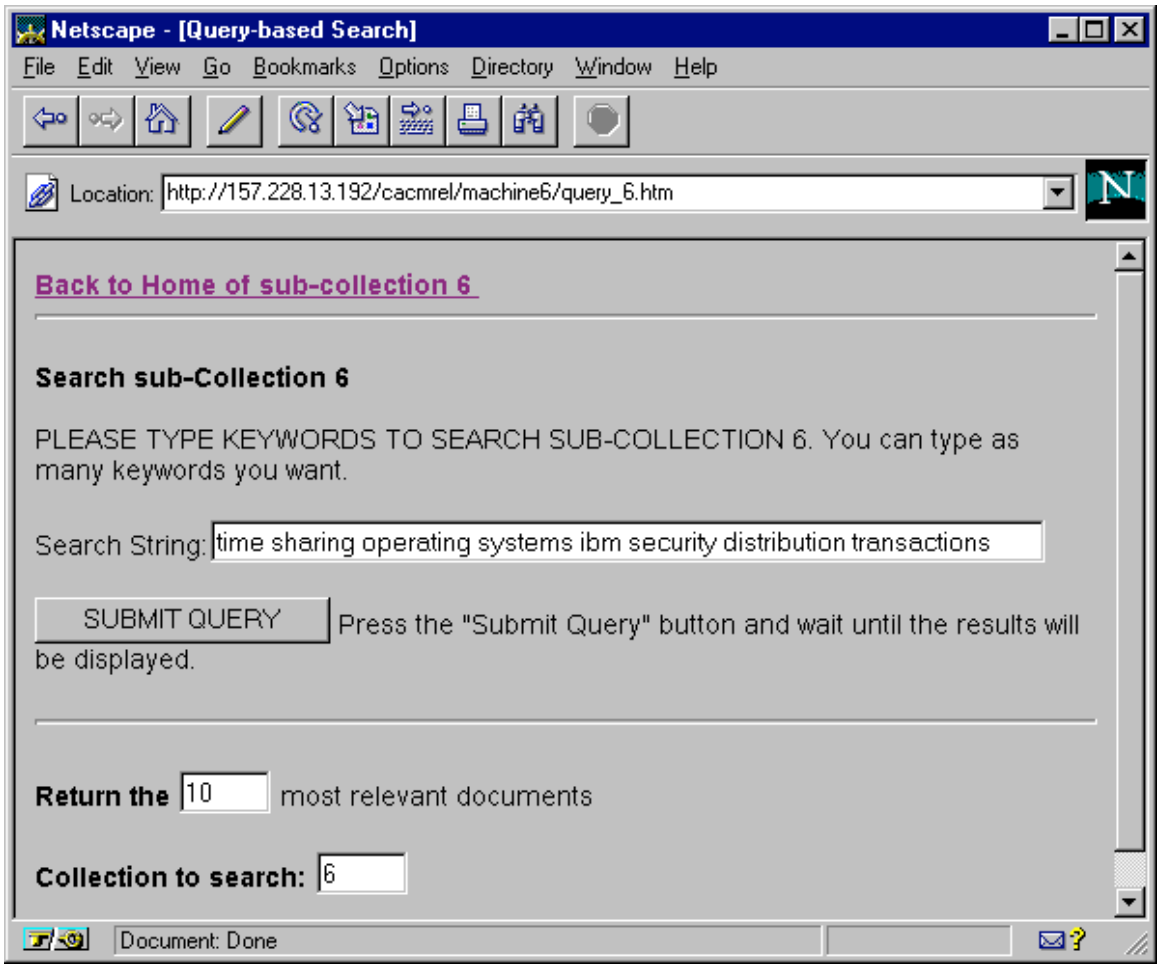


Figure 6. The page used to construct and run queries

NOTE

The result of a query is list of documents **ranked in relevance order** (i.e. the first document is regarded by the search engine to be more relevant etc.). The fact that a file is regarded as relevant by the search engine, does not quarantee that it is actually relevant to your query. Use the returned list to examine and decide which of the returned documents are actually relevant.

NOTE

Only the sub-collection in which the search page belongs is being searched.

If you want to search all the sub-libraries for a particular query, you must search all the sub-collections one by one .

If the search engine can not find any relevant documents a page will be displayed which will appropriately inform you.

5. The CACM documents

Figure 7. shows an example of an CACM document. Not all the documents are complete (i.e. some of them don't have abstracts and keywords), but all of them have links to other CACM documents.

Additionally, all the documents have links which allow an information seeker to move quickly to home pages, search pages, list of documents, links etc.

6. The Task

You will be asked to perform information retrieval tasks. In other words **I will give you a query and tell how many relevant documents to this query exist** in the CACM document collection. **I won't tell you in which sub-libraries these files exist**. Then you will be asked to search all the sub-libraries (starting form everyone you want) and try to find these relevant documents. If you believe that a document is relevant to a query simply write its code in a sheet that it will be provided to you.

An example of an information need could be:

What articles exist which deal with PLL, an insecure and distributed operating system for computers X?

You should try to find all the documents which can be relevant to this information need. In order to achieve that can use browsing or query-based searching or arbitrary mixtures of both to find the relevant documents.

We expect that users will initiate a search session by making query-based searches to sub-libraries, because it will be very difficult to start by browsing, due to the large number of documents.

However, feel free to select the search strategy you believe is more effective

Note

I will give a query and I will ask you to find all the documents which you believe are relevant to the query.

Note

These documents can be in any sub-library and not necessarily in just one. For example if a query X has two relevant documents, one relevant document can be in sub-library 1 and the second relevant document can be in sub-library 6.

Note

*I will tell you **in advance** how many relevant document do exist in the collection. So, you will know for how many documents are you looking for.*

Note

Use query-based searches or/and browsing to find the relevant documents

Note

Write down any documents you believe is relevant

Note

You may be asked to answer a very small questionnaire at the end of the experiment.

IMPORTANT

The documents in the CACM collection are quite old and you may not be familiar with the subject of the query. It is understandable if you can't find any relevant document at all.

The experiment aims to study the behaviour of information seekers, not to test your performance.

Therefore, please feel relaxed.

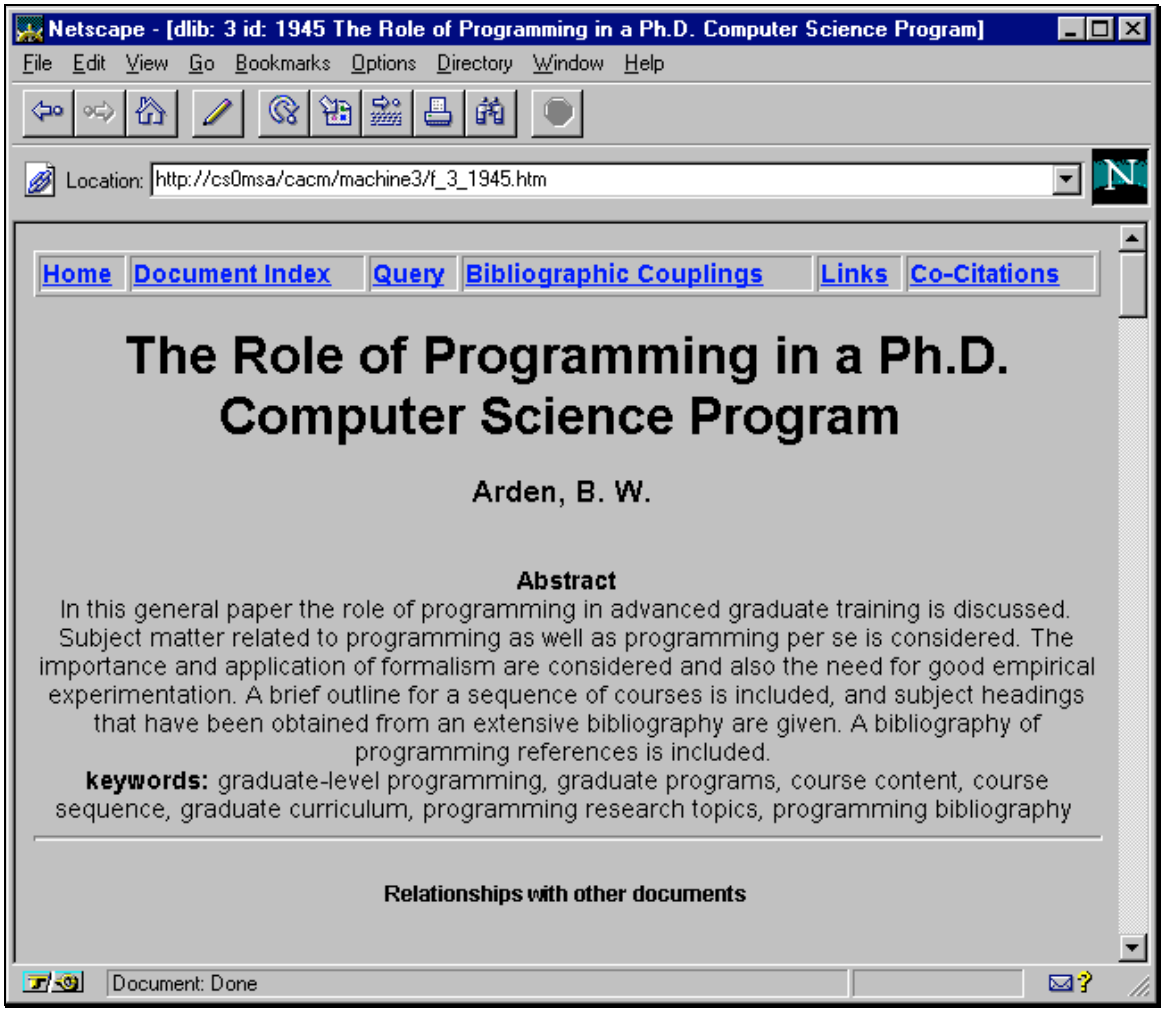


Figure 7. An example of a CACM document

Appendix C

Questionnaire used in the evaluation of the NIKOS OHS

About this questionnaire

Check the answer you think is correct. A five scale is used for answering the questions. The two ends of the scale are positive and negative. The middle choice is neutral. If you think a question is not relevant, or you do not have any opinion circle the N/O option.

System speed

Was the system acceptable in terms of speed ?

Poor ☐ ☐ ☐ ☐ ☐ Good N/O

System comprehension

Was the data/organisational model easy to understand ?

Difficult ☐ ☐ ☐ ☐ ☐ Easy N/O

Were the hypermedia agents difficult to understand ?

Easy ☐ ☐ ☐ ☐ ☐ Difficult N/O

Was the process model/information seeking process difficult to understand ?

Easy ☐ ☐ ☐ ☐ ☐ Difficult N/O

Was the interface of the hypermedia/agents system easy to understand ?

Easy ☐ ☐ ☐ ☐ ☐ Difficult N/O

Interface

Was the interface of the hypermedia/agents system difficult to use ?

Easy ☐ ☐ ☐ ☐ ☐ Difficult N/O

Was the interface of the interface/layout of the hypermedia agents easy to adapt/customise ?

Difficult ☐ ☐ ☐ ☐ ☐ Easy N/O

Usability

Was the system easy to use ?

Difficult ☐ ☐ ☐ ☐ ☐ Easy N/O

Did you find difficult to co-ordinate the different tools/hypermedia agents ?

Easy ☐ ☐ ☐ ☐ ☐ Difficult N/O

How will you characterise the system in terms of interactivity ?

Non Interactive ☐ ☐ ☐ ☐ ☐ Highly Interactive N/O

Was is difficult to make an analytical searching ?

Easy ☐ ☐ ☐ ☐ ☐ Difficult N/O

Was is easy to make a distributed/parallel analytical searching ?

Difficult ☐ ☐ ☐ ☐ ☐ Easy N/O

Information seeking

Do you think that distributed analytical searching was useful during your information seeking process ?

Not useful ☐ ☐ ☐ ☐ ☐ very useful N/O

Do you think that the source suggestion/selection is not useful for your searching ?

very useful Not useful

☐ ☐ ☐ ☐ ☐ N/O

Was the clustered browsing useful for your searching ?

Not useful very useful

☐ ☐ ☐ ☐ ☐ N/O

Was is difficult to "switch" from an information seeking strategy to another ?

Difficult Easy

☐ ☐ ☐ ☐ ☐ N/O

What do you think for the following statement: "the combined use of multiple strategies is more effective than using a single strategy "

Strongly Disagree Strongly Agree

☐ ☐ ☐ ☐ ☐ N/O