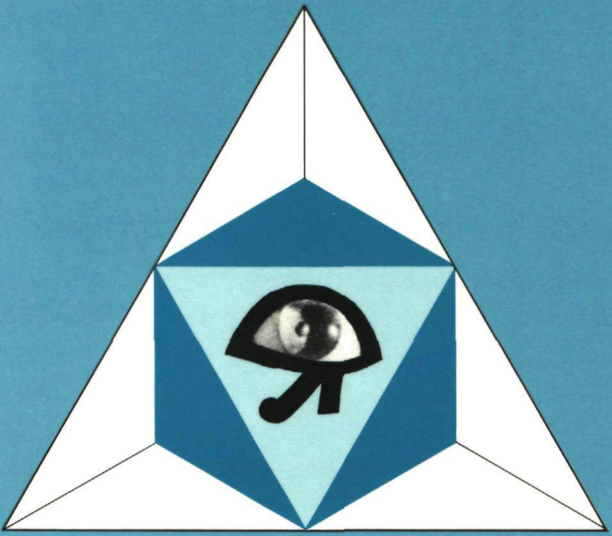

Stratified Information Disclosure

A Synthesis between Hypermedia and Information Retrieval

P.D. Bruza

Stratified Information Disclosure

A SYNTHESIS BETWEEN HYPERMEDIA AND INFORMATION RETRIEVAL

Manuscriptcommissie:

Prof. dr. M. Agosti, *University of Padua*
Prof. dr. K.M. van Hee, *Technical University Eindhoven*
Prof. dr. C.J. van Rijsbergen, *University of Glasgow*

The eye of **Ptah** , *the Disk of heaven, who illumineth the world by the fire of his eyes*
His name means the "opener"

Stratified Information Disclosure

A SYNTHESIS BETWEEN HYPERMEDIA AND INFORMATION RETRIEVAL

een wetenschappelijke proeve op het gebied
van
de Wiskunde en Informatica

Proefschrift ter verkrijging van de graad van doctor aan
de Katholieke Universiteit Nijmegen,
volgens besluit van het College van Decanen in het
openbaar te verdedigen op
vrijdag 5 maart 1993
des namiddags te **1.00 uur** precies

door
Peter David Bruza
geboren op 5 januari 1962
te Brisbane, Australië

THESIS PUBLISHERS
AMSTERDAM 1993

Promotor: Prof. dr. E.D. Falkenberg

Co-promotor: Dr. ir. T.P. van der Weide

CIP-DATA KONINGLIJKE BIBLIOTHEEK, DEN HAAG

Bruza, Peter David

Stratified information disclosure: a synthesis between
hypermedia and information retrieval / Peter David Bruza.
- [S.l. : s.n.]. - Ill.
Thesis Nijmegen - With index, ref. - With summary in Dutch.
ISBN 90-9005760-9
Subject headings: information retrieval / hypermedia / document retrieval

Contents

Preface

The research reported in this treatise was performed within the framework of the ESPRIT II project *APPED* (2499) from January 1989 to February 1992. *APPED*[1] is an acronym for **CDROM Tools: An application editor's and software developer's workbench for publishing multi-media information using optical read-only storage devices.** As the long winded title suggests, the purpose of the project was to produce a workbench, behind which, a developer could produce applications for optical media. The input of the workbench was assumed to be all forms of electronic data, for example, texts, word processor files, audio traces, video clips, graphics etc. These days, data can appear in any one of a plethora of formats, so the important first phase in building an application was the transduction of the source data into a set of internal formats amenable by the workbench. Where possible, accepted standards were adopted for this. For example, raw text data is translated into SGML marked up documents. A choice for SGML was made as it is an ISO standard for the specification of documents. Those who have ever noticed the difference between the philosophies behind Word Perfect and SGML can appreciate the difficult task of transduction which was entrusted to *ACT Systéme*, the French partner in the project consortium.

Once the raw data had been laboriously translated into the internal formats of the workbench, the author can begin weaving the data into an application. The basis for this was the document structure which had been detected by the transduction process. This structure forms a skeleton which the developer adorns with the multimedia data. The English company *Clarinet Systems* was responsible for constructing the tools for this.

As optical discs allow the storage of large amounts of information, an important issue was making the information accessible to the user. To this end, texts must be indexed. Indexing is a speciality of *Textware*, a Danish company situated in Copenhagen. As the workbench was aimed at the European community at large, the indexing process was complicated by multi-lingual considerations.

Once the developer has woven an application together and indexed it for efficient access, the task is not yet complete. As the target medium was read-only, the application should be checked before it can be mass produced. Unfortunately, there are no tools to do this automatically, so the developer must simulate the application in order to detect inconsistencies, errors and poor response time. To design and implement such a simulator was the

[1] Project 2499 had in fact the longest title of all proposals in that call

task of the Dutch company *Elektroson*. After the author had simulated the application and was satisfied, a master tape could be written out by the workbench. This tape is used for the pressing of the actual discs.

The fifth member of the project consortium was the *Department of Information Systems, University of Nijmegen*, The Netherlands. The expertise required of her was in the area of database optimization, both from the viewpoint of query optimization and internal data organization. It seemed as if the problem could be approached in a way that is traditionally done in the area of formatted databases: Querys are translated into normal forms which are designed to optimize the effort of the database engine. Data is stored in structures designed to balance disk accesses against storage overhead. At the beginning of the project it became evident that the applications produced by the workbench would not be traditional database applications, but rather so called *hypermedia*. Such applications are easily envisaged:

> Consider that you are sitting in front of a terminal. The screen shows a picture of Mozart and *Eine kleine Nachtmusik* is wofting through the audio channel of your personal computer. The screen also features some text. You read that Mozart was born in Salzburg and notice that the word "Salzburg" is highlighted on the screen. You wonder what Salzburg was like at the time Mozart. With your mouse you click on the highlighted area and you are presented with a map of Salzburg as it was in the mid-1700's. This screen also features highlighted areas, for example, various Renaissance and baroque churches. Any of these areas can again be activated.

Browsing in the above fashion has gained increasing popularity due to the ease and speed at which can move through the information in the application. Browsing, together with the enrichment offered by various media types are trademarks of hypermedia applications. A common snag, however, is that the user can become disoriented by losing sight of what he or she was originally looking for. Therefore, the problem of optimization is not necessarily one measured in disk accesses or number of bytes. It takes on another aspect, namely, the extent to which users are able to retrieve relevant information in the application. This treatise, then, is about the effective disclosure of information.

Acknowledgements

This monograph would not have manifested itself without ESPRIT Project *APPED.* Thanks go to Steven van Kervel-Jens Erlandsen for securing the project and Ruud Broers-Piet Plat for dragging it over the line. Giles Guezennec, thanks for the pleasant discussions within the plumes of *Gitanes* smoke, Stephen Scholfield for the entertaining vagueness, Holger Hvelgund for the saving Architectural Design Document, Morten Erlansen for the quick responses on the structured text database, Pieter van der Spiegel for the clever hyperindex implementation and Marten Nielen for almost getting XA simulation to work.

I extend my gratitude to Eckhard Falkenberg for providing me the chance to come to Nijmegen and giving me a free hand to pursue my interests. For his wise and detailed instruction in the creation and manipulation of formalisms, I will always be deeply indebted to Theo van der Weide. Thanks propagate to Linda van der Gaag for her guidance in the intricate and compact world of belief networks. I acknowledge with pleasure the traces of former students in this treatise: Rob Bosman, Ronald Bouwman and Anton Rosing. Wieneke Weursten and Constant Cuipers of the Art-History department, University of Nijmegen; thank you the interesting test environment and the instruction in the difference between *Maria* and *Madonna.* To present and former members of the *Afdeling Informatiesystemen*, the fond memories of our collaboration will stay with me. Arthur ter Hofstede: I found our discussions buoyant and instructive; Patrick van Bommel: my thanks for sharing a path. Kristine (Spiz) Breeze and Mirjam Bode; what a great cover illustration! As a newcomer to the Netherlands, I will never forget Remco Aalders, Noortje Hueber and Josè Kunst for their kindness and mothering. Yvonne, there is no *gezelligheid* (and thorough proof reading) without you. Finally, *Aiti* and *Tatinec*, my deep gratitude to both of you for making that long trip to Australia so your children could enjoy the chances you never had.

Chapter 1

The Information Retrieval Problem

"I don't know what I'm looking for,
but I'll know when I find it".

1.1 Introduction

It is the depths of winter and your central heating breaks down. If you are not heater mechanic, you must call for expert assistance, so you consult the yellow pages. The first problem is - under which heading does one look? In this example, *central heatıng* seems to be a logical choice. After searching the index however you find that under this heading all imaginable aspects about central heating are covered except about their repair. You then look under the heading *repair* or try to think of another heading under which central heating could fall.

This scenario characterizes the fact that searching for information is fraught with stumbling blocks. These days information is stored in so many ways in so many places that it is no simple task to satisfy an information need. The searcher is confronted with all manner of catalogi, indexes (like in the yellow pages) in both automated and unautomated systems. In automated systems there is often the additional overhead of coming to grips with the user interface.

Another common obstacle is the searcher not having a clear idea of what (s)he is actually looking for. Only when confronted with relevant material does the realization occur that this information is what was actually being sought after. In such cases, it is important that the system help the searcher discover their specific information need. Basically, if the the searcher cannot easily map their information need (vaque or otherwise) into the system designed to disclose the information, then locating relevant information can be reduced to trying to find a needle in a haystack; and the haystacks are becoming larger and larger. Society's ability to invent still new ways to proliferate and use information is fueling an information explosion. The challenge is to provide fast and effective access to the ever increasing amounts of information. We arrive at the central theme of this treatise, a theme traditionally summarized by the term *informatıon retrıeval.*

1.2 Information Retrieval and Information Disclosure

Information retrieval is something that mankind has probably been doing since it began recording its affairs in the form of symbols. A fundamental aspect of information retrieval is that information (denoted by symbols) is *stored*. It is stored so that it can be retrieved and found again without undue effort. In other words, the stored information must be coupled with a *disclosure* system so that it can be *revealed* or *divulged* to the searcher in a systematic fashion. The index in the yellow pages is one example of a disclosure system. The term *information retrieval* is inadequate in the sense that the term does not reflect that a disclosure system is involved. We will however adhere to this term as it is traditional terminology. The preferable term *information disclosure* will also be employed, particularly in discussions dealing with specific systems developed for divulging information.

1.2.1 The Information Retrieval Paradigm

Whether it is trying to find the telephone number of a heater mechanic or finding relevant literature to help solve a problem in nuclear physics, the concepts involved in the disclosure of information are the same. The following example of a typical search for information provides the context for the identification and elaboration of these concepts:

> *Roman Polanski, the famed film director, goes to a library. He wants to learn something about* 'stars'*, so he types into the automatic catalog system the term* stars*. After a few seconds the result comes back and he notices the first entries describe books about super novae and white dwarves. He therefore types in movie stars.*

Information disclosure begins with a person having an information need that (s)he wishes to fulfill. (See figure 1.1). Henceforth, we will denote this person as the *searcher* and the *information need* as N. The information need is concretized in the form of a *request*, denoted q, which is given to an automatic information disclosure system, or a human intermediary, such as a librarian. The intention is that the request be as good as possible description of the need N. In addition there is the information to be disclosed. This is modelled as a set $\mathcal{O}$ of *information objects* . The information objects are also referred to as information carriers, or documents.

The set of concepts is not yet complete. There remain hidden aspects in the above example. Roman Polanski believes that the information disclosure system understands which books are about which topics, in much the same way that a librarian does. This is in fact not the case. The system does not understand anything about the contents of the books. Within the system the content of each book is represented by a small set of descriptors for the purposes of its disclosure. These descriptors are drawn from a *descriptor language* denoted by $\mathcal{C}$. The *characterization* of an object will be denoted by $\chi(O)$. This characterization is arrived at by a process termed *indexing*. Information disclosure is typically driven by a

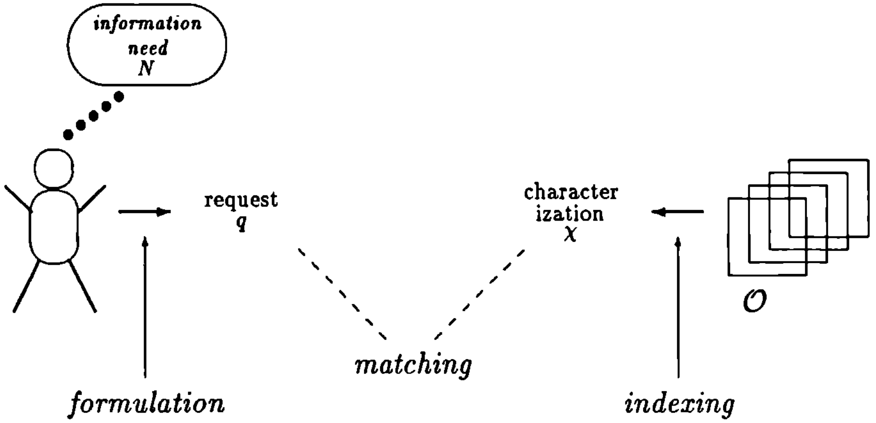

Figure 1.1: The Information Retrieval Paradigm

process called *matching*. In this process the request is compared with the characterization of objects. If the matching operation deems an object as being sufficiently similar to the request, then the object is assumed relevant and returned (thus disclosing it).

The Information Retrieval Paradigm has now been established. A limited number of concepts are involved, but, nevertheless the disclosure of information turns out to be a formidable problem due to the following reasons:

- *formulation* is not easy
- *indexing* produces incomplete object characterizations
- *matching* is based on erroneous assumptions

The above three assertions are threads which loosely bind the first three chapters of this dissertation. In the following section the problem of formulation will be studied in more detail. In the following chapter various characterization languages for information disclosure are featured. The expressiveness and incompleteness of these languages will be addressed. This thesis highlights two erroneous assumptions with regard to matching. The first is presented later in this chapter, the second is used as a point of departure for chapter 3, in which the process of matching is considered within a logical framework.

1.3 Formulation is not Easy

In the famous Cranfield information retrieval tests [Cle91], it was found that sixty percent of the time searchers produced inexact specifications of their information need. If we generalize this result, the implication is that sixty percent of the time searchers formulate erroneous requests. In view of this it is important that a disclosure system supports the searcher during the formulation process. Note that searchers in hypermedia are not confronted with the problem of request formulation as information disclosure is realized by navigation. Within the stratified hypermedia architecture request formulation is supported by navigation within a special layer to the desired description of the information need. We will deal with this further in chapter 4. First, however, we endeavour to shed some light onto the problem of request formulation. Broadly speaking, an information need can be classified into one of two types, depending on how difficult the need was to concretize. There are *extensional information needs* and *intentional information needs.*

1.3.1 Extensional Information Needs

These information needs could be formulated in terms of a single or combination of *extensional* characteristics of the information objects. For example, *retrieve all documents written by Einstein between 1910 and 1912.* Here the date and author are typical examples of extensional characteristics. Such information needs tend to be clear and specific in nature. For this reason, they can be described by the adage: *I know what I want and I know how to find it.* The information needs of this type are similar to the information needs expressed in queries given to formatted databases. For example, *return all records about employees who are managers and earn more than 100,000 dollars per year.*

1.3.2 Intentional Information Needs

Information needs of this type are typically related to the subject or *intention* of information objects. Blair [Bla90] states that the principal difficulty which confronts the searcher is predicting the words or phrases that have been used to characterize the objects which would satisfy the information need. As a result, the request itself is often no more than a single descriptor or set of descriptors; it barely begins to describe the full information need. For example, Roman Polanski came up with the formulation *stars*, which turned out to be imprecise.

Roman Polanski had at least some notion of the information he desired. Even more difficult is when the information need is not clear to the searcher and consequently is almost impossible to formulate precisely. Consider the following illustration taken from [Coo71].

> "Just what is it, exactly, that you would like to find out about halogen elements?"
>
> ... "Oh, I don't know - all about them, I guess".

Even though the above searcher claims that (s)he wants to know everything about a halogen elements, this proves in general not to be the case. During the search the information need often evolves and becomes more concrete. For this reason it is important that the disclosure system helps the searcher *home in* on a good formulation of their information need.

Librarians, indexes and classification systems are typical examples of aids in this regard. Blair [Bla90] postulates that librarians have not been replaced by automatic disclosure systems because the homing in process is so difficult.

An interesting facet of searchers which do not have a clear idea of their information need is that they can almost always determine whether a given information object is relevant, or not. Hence the adage, ***I don't know what I want, but I'll know when I find it.***

Other aspects are also involved when considering why the formulation of an information need is difficult. For example, the language in which the request is to be expressed can also influence the formulation process. Basically, if the language is not "*formulation friendly*", the chance of erroneous formulations increases. The Boolean language, in which requests can be expressed as terms combined using the logical connectives AND, OR and NOT, is often criticized as being "formulation *un*-friendly" as it leads to complex formulations. Heise [Hei91] aptly illustrates this with the following request which attempts to specify all documents that are about *computer security in UNIX environments.*

```
(‘Unix‘ OR ‘BSD‘ OR PHRASE(‘system‘,‘V‘) OR ‘Ultrix‘ OR ‘SunOS‘
OR ‘HP/UX‘ OR ‘A/UX‘ OR ‘AIX‘) AND PHRASE(‘computer‘,‘security‘) OR
PHRASE (‘data‘,‘security‘) OR ‘decoding‘ OR ‘coding‘ OR ‘encryption‘
OR ‘cryptography‘ OR PHRASE(‘access‘,‘control‘) OR ‘classification‘
OR ‘password‘ OR PHRASE(‘user‘,‘identification‘)
```

The example also shows the intricacies of the formulation process. Note how the above request attempts to specify all of the contexts in which *computer security* and *UNIX* can be found. The average searcher is in general unaware of such contexts. Blair [Bla90] succinctly expresses this phenomenon as follows:

> *It is not a simple matter for users to forsee the exact words and phrases that will be used in the documents they will find useful, and only in those documents.*

1.4 Relevance

Once the information need has been formalized in the form of a request, it can be responded to via manual means, for example, by a librarian locating potentially relevant books, or as is more common these days, by an automated information disclosure system. In both cases, information objects are returned, *but* most often not all of the objects in the result set will be *relevant* with respect to the specific information need.

The question of *relevance* is an important one and has long been scrutinized and philosophized in information retrieval research, simply because the goal of information retrieval is to return as many relevant objects as possible to a given information need. (Blair [Bla90] gives a thorough discussion on relevance). In order to fulfill this goal it is necessary to define formally what relevance is. The underlying problem seems to be that the information need defies formalization. Cooper [Coo71] describes the information need as a "psychological state" and as such not a "visible object or complex of symbols ... something not directly observable". Furthermore, the determination of relevance appears to be a *subjective* process. In [Bla90] research is cited which concludes that searchers can consistently and easily determine the relevance of an information object with respect to their information need without being able to concretize the criteria they use for this. This phenomenon has been compared with how people can recognize faces without being able to elucidate how they do this. In other words, relevance is a clearly defined issue with respect to the searcher, *but* difficult to define operationally.

Cooper [Coo71] made an important contribution to increasing the understanding what relevance is, by his critical assessment of it. He first made a distinction between so called *logical relevance* and *utility*. A given object is logically relevant to the information need if the object is topically related to the need. Utility, on the other hand, is purely a pragmatic notion: Is the object *useful* to the searcher? The difference between the two notions is made clear when one considers the possibility that even though an object may be topically related to an information need, the searcher may reject it because they do not trust the information in it as being accurate.

Cooper also provided a formal definition of logical relevance in terms of logical consequence. If the (representation of) information need is a logical consequence of the object, then the object is deemed to be relevant. This definition can be considered a milestone in information retrieval because it was the first rigid definition of relevance and twenty years later it is fundamental to logic-based information disclosure. (This topic is studied at length in Chapter 3). Defining relevance in terms of logical consequence implies that an object is (logically) relevant with respect to an information need, or it is not. Therefore, Cooper did not view relevance as being "grey". According to Cooper, the greyness is a product of a matching algorithm which does not fully complete the inferential process.

A crucial assumption in the above discussion on logical relevance is that there is a perfect representation of the information need. Cooper, in fact, makes a careful distinction between this representation and the request, which is most often an imperfect description of the information need. In most cases, however, the request q is assumed to be a perfect representation of the information need N. The Roman Polanski example shows how dubious this assumption can be. Even though the book on supernovae and white dwarves is relevant with respect to the request *stars*, it is clearly not relevant with respect to Roman Polanski's specific information need. On the other hand, $q = N$ seems to be the only reasonable assumption that can be made because a perfect representation of the information need N is probably as elusive as N itself.

The assumption $q = N$ allows the notion of logical relevance to be defined in terms of q. This has nice pragmatic consequences - the difficult notion N can be filtered out allowing relevance to be equated with *aboutness*. From this point on we will use $\texttt{Relevant}(O, N)$ to denote that object O is relevant to information need N, and $\texttt{About}(O, q)$ to denote that O is *topically related*, is *about* or *deals with* request q. An object O is logically relevant to a request q (and thus to N), if q is a logical consequence of O. In short, a disclosure system can assume relevance to an information need by adopting $q = N$ and proving that O deals with q:

$$q = N \;\Rightarrow\; [\texttt{About}(O,q) \Rightarrow \texttt{Relevant}(O,N)]$$

The information retrieval problem is in this way equated with the problem of finding all objects that are logically relevant to q.

1.5 Information Disclosure Effectiveness

		$\texttt{Relevant}(O,N)$	
		yes	no
$\texttt{About}(O,q)$	yes		*noise*
	no	*lying by omission*	

Figure 1.2: Relevance vs Aboutness

Once the information need has been formulated into request q, the information disclosure system responds by retrieving objects that deal with q. As the Roman Polanski example demonstrates, objects may be returned that are not relevant with respect to N. A consequence of assumption $q = N$ generally not being valid, and exacerbated by the fact that disclosure mechanisms work with incomplete object characterizations (the subject of the next chapter), is an imbalance between relevance and aboutness. This imbalance is reflected in the partition depicted in figure 1.2. The partioning shows that the searcher is confronted with two undesirable situations: The first is that relevant objects are not disclosed ($\texttt{Relevant}(O,N)$ and $\neg\texttt{About}(O,q)$), or in other words, the disclosure system is *lying by omission.* This can be a serious problem, particularly in so called *exhaustive searches.* As the name suggests, such searches require all relevant objects to be found, for example, patent searching. Very often the searcher is not aware that the disclosure system is lying by omission. In the STAIRS experiment [Bla90] lawyers must have reviewed seventy-five percent of the relevant documents in order to suitably prepare for a case. Using the automated disclosure system the lawyers were on average retrieving less that twenty percent of the relevant material, but believed that they had retrieved all of the relevant documents!

The second undesirable situation is that the disclosure system returns *noise* in the form of objects that are not relevant to the information need of the searcher ($\neg\texttt{Relevant}(O,N)$ and $\texttt{About}(O,q)$). This can be a burden on the searcher because (s)he must sift through non-relevant objects in the result set in order to locate the relevant ones. If there is too much noise they may give up prematurely, thus missing potentially interesting information. The stratified hypermedia architecture aims to fight the noise factor on two fronts. Firstly, a special layer within the architecture supports formulation of the information need thereby hopefully leading to precise requests. Secondly, employment of the logic-based information disclosure paradigm aims at realizing a sensitive disclosure mechanism. More about this in ensuing chapters.

In the light of the above discussion, the effectiveness of a disclosure system should depend on how successful it is at divulging relevant objects with a minimum of associated noise. A number of criteria for evaluating disclosure have evolved from this motivation. Foremost among them are *recall* and *precision.*

1.5.1 Recall and Precision

Recall is a normalized measure of how much the disclosure system is lying by omission. Precision, on the other hand, is a normalized measure which reflects the noise being produced by the disclosure system. If Rel(N) denotes the set of objects relevant with respect to the information need N, then recall and precision are formally defined as follows:

Definition 1.5.1

$$\texttt{precision}_N(q) = \frac{|\texttt{Rel}(N) \cap res(q)|}{|res(q)|}$$

□

This formula implies that if all returned objects in the result set are relevant with respect to q, then maximum precision (no noise) is achieved.

Definition 1.5.2

$$\texttt{recall}_N(q) = \frac{|\texttt{Rel}(N) \cap res(q)|}{|\texttt{Rel}(N)|}$$

□

Maximum recall is attained if all relevant objects are disclosed ($\texttt{Rel}(N) \subseteq res(q)$). A disclosure system attempts to maximize recall and precision, however it has been shown experimentally that these criteria interact with each other in a way similar to the *time-space* trade-off, that is recall can be increased at the expense of precision and *vice versa.*

1.5.2 Experimental Information Disclosure

Recall and precision are employed in an experimental setting to compare the effectiveness of disclosure systems. Given a set of objects and a set of queries, two disclosure systems can be compared based on average recall and precision figures. Statistical tests of significance can be applied to test whether one disclosure system is better than the other at a given confidence level. (Many books on information retrieval give details regarding such tests, for example, the book by Salton [Sal83]).

A problem with such experiments is defining a suitable set of relevant objects $\mathtt{Rel}(N)$ for a given test query q. The usual procedure is to associate *a priori* each request with a set of objects which are deemed relevant to it. In this way the experimenter tries to capture what the average searcher would find relevant with respect to the information need suggested by the request. As an aside, the fact that the supposed information need is established from the very beginning is why N often does not feature in definitions of recall and precision given in the literature. From this point on we will also adhere to this convention.

The Cranfield experiments [Cle91] of the early sixties were conducted using the above experimental paradigm. Information disclosure experiments up to the present day have basically followed this scheme. There are some, however, who question this. Van Rijsbergen [Rij89] argues that the paradigm, which was developed in the context of controlled library environments, is not suitable in other contexts such as offices and large distributed computer environments. Blair [Bla90] is also of this opinion and postulates that experiments with large object bases are fundamentally different from small scale systems. To date, disclosure effectiveness has really only been studied via small scale experiments in controlled environments. Unfortunately, there does not exist a theory in which disclosure systems can be described and compared inductively. It is not the intention of this treatise to fill this gap. Like the above authors, we acknowledge the weaknesses of the experimental approach, but will nevertheless apply it in chapter 5.

Chapter 2

Characterization Languages for Information Disclosure

> *Meaning depends very greatly on the connectives between nouns and verbs, and these connectives are the means of expressing relations.*
>
> Jason Farradane

2.1 Introduction

In the previous chapter, three assertions were put forward as to why information disclosure is problematic. So far we have dealt with the first of these by giving some background why the formulation of the information need can be difficult. This chapter has to do with the second assertion, namely that information disclosure is not ideal because the characterization of the information objects is incomplete. An incomplete characterization means that there are aspects of the object that are not represented in its associated characterization. As a result, if there is a request directed at any of these hidden aspects, the object cannot be disclosed, thereby reducing recall.

How to characterize an object to facilitate its disclosure has long been one of the driving questions in information disclosure. In terms of the terminology introduced in chapter 1, this question boils down to the choice of the descriptor (characterization) language $\mathcal{C}$. This choice implicitly involves other questions. How can it be determined that a language $\mathcal{C}_1$ offers better disclosure than a language $\mathcal{C}_2$? Another issue is whether $\mathcal{C}$ should also serve as the language of requests. It is thinkable that a given language may be well suited for the purposes of characterization, but not suitable for request formulation. Maron [Mar77] points out that once $\mathcal{C}$ is chosen two issues must be resolved. The first is the so called *indexing problem*, namely how to assign those words in $\mathcal{C}$ to an object O to facilitate O's disclosure. The second issue is how the disclosure mechanism is to use $\mathcal{C}$ in order to realize effective information disclosure. These two questions constitute underlying themes for this and succeeding chapters. After providing initial background in some state-of-the-art characterization languages and their associated indexing algorithms, this chapter highlights so called *index expressions* as an improved characterization mechanism. In addition, an automatic expression indexing algorithm is specified and tested using the CACM and Cranfield document collections. These are standard collections used for research in information disclosure.

2.2 Characterization of Information Objects

In the world around us an object usually has a unique identification so that it can be distinguished from other objects. Such a unique identification is necessary to singularly disclose that object. In the framework of the information disclosure paradigm introduced earlier, an object O has a characterization $\chi(O)$ drawn from the language C. Therefore, in order to effectively disclose O, $\chi(O)$ must distinguish object O from other objects. For example, if a book about movie stars is characterized by the index term **stars**, it will not be distinguishable from books dealing with astrophysics. Furthermore, $\chi(O)$ must also *usably* distinguish O. The disk address of an information object distinguishes it perfectly from other objects, but for the purposes of disclosure by a human searcher this characterization is almost certainly useless. Usability and discrimination are fundamental to the *Information Disclosure Principle* which states that in order to effectively disclose an information object it must be characterized so that it is usably distinguishable from other information objects. It follows from this principle that a descriptor language must have sufficient expressive power to realize discrimination and at the same time be useful in a pragmatic sense. (Chapter 4 of Blair's book [Bla90] gives a linguio-philosophical motivation regarding the pragmatics of characterization languages).

Before we introduce some common characterization languages, the notion of indexing is formalized. Given a descriptor language C and a set $\mathcal{O}$ of objects, the indexing process defines a relation $\chi \subset \mathcal{O} \times C$. An object characterization is derivable from χ as follows, $\chi(o) = \{c|(o,c) \in \chi\}$ In the past, indexing was performed by a person who scanned the information object and assigned descriptors from C which (s)he believed were both a good reflection of the content of the object, and were likely to be used in a request for that object. With the advent of computers, *automatic indexing* has come to the fore, as it is requisite to disclosing the ever growing mountains of information.

2.2.1 The Finite Language of Terms

There are a number of forms that a descriptor can take. The most elementary form of descriptor is a *keyword*, or *term*. Keyword descriptors have the advantage that there are a number of straightforward indexing algorithms to derive them automatically from the objects ([Sal89]). These methods all have the same underlying motivation, namely to extract "good" terms from the object. With the *Information Disclosure Principle* in mind, a *good* term is one which usably discriminates the object from other objects. The following rationale is often used to identify such terms: A term t usably discriminates an object O, if it occurs relatively frequently in O and relatively infrequently in the other objects. It is instructive to look at this rationale more closely on the basis of the following definitions.

Definition 2.2.1

$\texttt{intfreq}(t, O)$ *denotes the occurrence frequency of term t in object O* □

The *external* frequency of a term with respect to an object base $\mathcal{O}$ is the number of objects that contain the term.

Definition 2.2.2

$$\texttt{extfreq}(t, \mathcal{O}) \;=\; \sum_{o \in \mathcal{O}} \texttt{intfreq}(t, o) > 0$$

□

When the object base is understood, $\texttt{extfreq}(t)$ will be used for short.

		$\texttt{intfreq}(t, O)$	
		large	small
$\texttt{extfreq}(t)$	large	*stop word*	*N.A.*
	small	**good**	*not usable*

Figure 2.1: Internal versus External frequency

There are four possibilities regarding the internal versus external frequency of a term t. These are depicted in figure 2.1. The first possibility affirms that term t occurs frequently both within a given object and across the object base. Imagine if t were to be used for object characterization, then the whole object base would be returned in response to the request t. In short, term t is hopeless at distinguishing objects from one another. Such terms are typically referred to as *stop words*. Examples of common stop words in English are the and a. (For a detailed list of English stop words refer to [Fox90]). Note that even though a term such as computer is not a stop word, it would probably be lousy at distinguishing books in a computer science library.

The second case is diametrically opposed to the stop words, namely term t discriminates objects well ($\texttt{extfreq}(t)$ is small), but is *not usable* under the assumption that usability is proportional to internal frequency.

The third case suggests that if t occurs frequently within the object it is likely to be *usable* ($\texttt{intfreq}(t, O)$ large) as well as being an effective discriminator of objects ($\texttt{extfreq}(t)$ is small). Therefore, in the light of the Information Disclosure Principle, t is a "good" term.

The fourth case is not applicable because the chance that a term will have a large external frequency and small internal frequency is negligible. In any case, such a term would not be a useful for object characterization for the same reason stop words are not.

Term Indexing

The term indexing algorithm depicted in figure 2.2 derives a characterization of an object on the basis deriving "good" terms as introduced in the previous section.

```
characterization function TermIndexer(O: Object)
    $\chi \leftarrow \emptyset$
    while $t \leftarrow gettoken(O)$ do
        if $\neg stopword(t)$ then
            if $t \in \chi$ then
                increment $intfreq(t, O)$
            else
                $\chi \leftarrow \chi \cup \{t\}$
                $intfreq(t, O) \leftarrow 1$
            fi
        fi
    od
    for $t \in \chi$ do
        if $w(t, O) < \epsilon$ then $\chi \leftarrow \chi \setminus \{t\}$ fi
        fi
    od
    $\chi$
end
```

Figure 2.2: A simple keyword indexing algorithm

In the first part of the algorithm all terms which are not stopwords are considered as being potentially good and are added to the object characterization χ. During this phase the internal frequency of such terms is calculated. In the second phase of the algorithm, the so called weight of each term is determined via the function w. The purpose of w is to quantify how good a given term will function as a descriptor of O. If this weight is below a given threshold ϵ, the term is deemed not to be a good descriptor for characterizing of O. A common way to calculate term weights is as follows:

$$w(t, O) = \mathtt{intfreq}(t, O) \log_2 \frac{|\mathcal{O}|}{\mathtt{extfreq}(t, \mathcal{O})}$$

This formula embodies the Information Disclosure Principle as it expresses the weight of a term is proportional to the internal frequency, thus reflecting the *usability* of the term, and inversely proportional to the external frequency, thus taking into account the ability of the term to distinguish objects. The factor $\frac{|\mathcal{O}|}{\mathtt{extfreq}(t,\mathcal{O})}$ is commonly referred to as the *inverse document frequency*.

The above formula has been shown by Salton [Sal89] to have some relation with information theory. From an information theoretic point of view, the terms in the object base with the smallest probability of occurrence have the highest so called information value. If $Pr(t)$ denotes the term t's probability of occurrence, then its information value is defined by Shannon's formula: $-log_2 Pr(t)$. As a consequence, stopwords have a low information value as they have a high probability of occurrence. The average information value conveyed by the underlying probability distribution is given by the so called entropy, $H(Pr) = -\sum_t Pr(t) log_2 Pr(t)$. Entropy has been used by Salton to derive a measure for term usefulness known as the *signal-noise ratio*. This ratio favours terms that are concentrated and is not favourable for terms that occur evenly across the object base. Salton states that the signal-noise ratio is essentially equivalent to the function w.

The above algorithm can be refined in a number of ways. A commonly used variation is to employ word stemming which reduces tokens to stems, which are then evaluated for suitability. In this way grammatical variations of a indexing term can be mapped to a single descriptor stem. For example, the tokens indexer and indexing would be mapped by the stem algorithm to the descriptor index. (See [Por80] for a commonly used stem algorithm).

The advantage of keyword descriptors is that the indexing process is straightforward and efficient. Their major disadvantage is that the object characterizations are necessarily incomplete. Consider that term based indexing reduces the content of a whole book to a relatively small set of keywords. Consider also that a term appears in certain context(s) within an object. In keyword indexing, the term is stripped out of its context. This context may well be crucial in determining if the object is relevant or not. Nevertheless, many information disclosure systems have been developed having term descriptors as their basis. Such models accept the limitations of the term-based object characterizations and strive to wring effective disclosure from these very incomplete characterizations.

2.2.2 The Language of Term Phrases

If computer is a term descriptor which characterizes objects dealing computers and analogously for programming, the problem of how to characterize an object that is specifically about computer programming arises. A *term phrase* is an extension of the term descriptors allowing more specific descriptors; the phrase computer programming is more detailed, and therefore distinguishes objects better, than the term computer or programming. Even though straight forward indexing methods exist for the generation of term phrases [Sal89], these methods often suffer from the problem that either too many non-meaningful phrases are generated, or conversely a large percentage of the phrases are meaningful but the resulting characterizations are incomplete. A good survey of term phrases together with advanced indexing algorithms can be found in the work of Gay and Croft [GC90]. With regard to the information disclosure principle, term phrase characterization languages offer better possibilities to distinguish objects, but the open question seems to be how to effectively use the term phrases in the associated information disclosure mechanism. There is no empirical evidence that suggests that term phrases offer better information disclosure than keywords.

2.2.3 The Language of N-grams

Up to this point all descriptors have been based on whole words. The n-grams of a word w are overlapping substrings of w of length n. For example, if $w =$ theory, then the *tri*-grams are the, heo, eor, ory. The n-gram characterization of an object can be indexed by the union of the n-grams produced by each word in the object. For example, the information objects:

O1 queueing theory is the basis of server systems

O2 server systems are based upon queueing theory

have the following *tri*-gram based characterizations which are presented in alphabetical order:

1. { are, ase, *bas, ein, ems, eor, erv, eue, heo, ing, ory*, pon, *que, rve*, sed, *ser, ste, sys, the, uei, ueu*, upo, *ver, yst* }
2. { asi, *bas, ein, ems, eor, erv, eue, heo, ing, ory, que, rve, ser*, sis, *ste, sys*, tem, *the, uei, ueu, ver, yst* }

These characterizations contain 24 and 22 elements respectively, 19 of which are in the intersection.

The number of different n-grams grows exponentially with n. For $n = 3$ there are $26^3 = 17,576$ different *tri*-grams, whereas there are 456,976 distinct *tetra*-grams. Teufel and Schmidt [TS88], however, state that only about 25% of the possible trigrams occur in real texts. They based their estimate on tests done using German and English texts. Research with Dutch texts has found that only between six and seven thousand trigrams actually occur [Sta90]. This represents roughly 35% of trigrams and is notably higher than the Teufel and Schmidt estimate. (The difference can be explained by the occurrence of vowel combinations in Dutch not found in German or English). Teufel and Schmidt claim that $n = 3$ is optimal with respect to the computational cost of producing the n-grams and information disclosure effectiveness.

N-grams have proven to be versatile because they can be readily produced by a straightforward syntactic process. For example, *tri*grams are often used in spelling checkers because spelling variations of a word normally have a similar set of associated trigrams [Sal89]. In another application, trigrams and tetragrams were used to as characterization mechanism for information in a wide area network [WF89].

Even though the language of trigrams allows objects to be usably distinguished, there is little empirical evidence regarding the effectiveness of disclosure systems based on this language. It can be predicted that the recall of trigram based disclosure system will be higher than that of a term based system because of the former's ability to deal with spelling variations. How this increase in recall will be paired by a decrease in precision. The Document Information Technology section of the Dutch research organization *TNO* claims that the loss in precision can be compensated by using trigram based characterizations only in conjunction with small information objects.

Sometimes the spelling variations of a term are so variable, that there is little or no overlap between their respective *tri*grams. For example, the word Krushchev verses Chroesjtsjov.

(This word apparently has 2880(!) spelling variations [NvJ91]). The effect is that if Kruschev is given as a request and a document contains the word Chroesjtsjov it is not possible to establish the link between these two terms by matching the respective trigram characterizations because the overlap of the respective trigram characterizations is empty. An offshoot of the trigrams, the so called triphones, have been developed to counter this sort of problem. A *triphone* is a three letter trace which denotes a sound. In the above case the two terms would map to a similar set of triphones thereby allowing the information disclosure mechanism or spelling checker to detect the strong relationship between these terms.

2.3 The Language of Index Expressions

Index expressions are an extension to the term phrases whereby the relationships between terms are modelled. Their philosophical basis stems from Farradane's *relational indexing* [Far80a][Far80b]. Farradane projected the idea that much of the meaning in information objects is denoted in the relationships between terms. A parallel can be drawn here with the conceptual model from the database world where relationship types between entities play an important role; the characterization of an object consisting solely of keywords would be like an entity relationship model without relationship types.

As there are many possible relationships between terms, Farradane proposed a framework of nine relationship *types* with which any given term relationship could be classified. For example, author wrote book exhibits a *functional dependence* relationship type between book and author. Farradane motivated his relationship types on the basis of psychological thought mechanisms.

In relational indexing, trained indexers would peruse an object and classify the term relationships. The resulting characterization comprises a network of terms, where each arc in the network represents one of the nine relationship types. Even though the resulting characterizations clearly capture more of the content of an object than the descriptor languages presented so far, the disadvantage is that indexing has to be performed manually. This is probably the reason why Farradane's relational indexing never blossomed.

Craven uses a similar approach to Farradane in his *linked phrase indexes* [Cra78][Cra86]. Like relational indexing, the basis of a linked phrase index is a network of terms, in which the arcs correspond to relationships denoted by prepositions. Such networks are also produced by a manual indexing process, although Craven does propose that automatic network derivation is possible from the titles of objects.

Index expressions have their roots in linked phrase indexes. In contrast to terms and term phrases, index expressions form a structure.

Definition 2.3.1

Let T be a set of terms and C a set of connectors. We define the language $\mathcal{L}(T,C)$ of index expressions over T and C by the following abstract syntax (in extended BNF format):

Expr $\rightarrow$ ϵ | Nexpr
Nexpr $\rightarrow$ Term {Connector Nexpr}*
Term $\rightarrow$ t , t $\in T$
Connector $\rightarrow$ c , c $\in C$

□

The symbol ϵ signifies the empty index expression. A term t basically corresponds to a noun, noun-qualifying adjective or noun phrase; a connector c denotes a relationship type between two terms and is basically restricted to the prepositions and the so-called *null connector* which is denoted by o. Figure 2.3 shows some of the allowable connectors and the relationship types they denote. Index expressions also have a concrete syntax. For example, the index expression **attitudes of (students in (universities)) to (war in (vietnam))**. Note how brackets are used to explicitly represent the structure specified by the abstract syntax. In effect, the structure corresponds to the interpretation of the index expression. We will often omit the brackets when presenting concrete syntax. Although it is possible that an index expression without the structure explicitly denoted is ambiguous, but in practice such cases seem to be rare. This facet of index expressions is comparable to natural language; even though it is often theoretically ambiguous, humans seem to be able to effectively resolve such ambiguity if the context is known.

Connector	*Relationship Type*	*Examples*
of	possession action-object	castle of queen pollination of crops
by	action-agent	voting by students
in, on, *etc.*	position	trees in garden
to, on, for, in	directed association	attitudes to courses research on voting
with, o, and	association	assistance with problems fruit o trees
as	equivalence	humans as searchers

Figure 2.3: Connector Table

From this point on, when we need to explicitly represent the structure of an index expression, it will often be represented graphically as a tree. The previous example is depicted graphically in figure 2.4.

The first term in an index expression I is referred to as the *lead term* denoted by $\lambda(I)$. (Later in this chapter the lead term will feature in many proofs of index expression properties). For

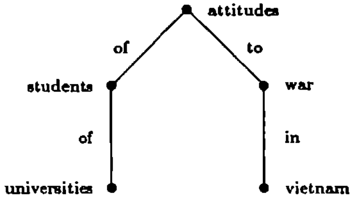

Figure 2.4: Example index expression

notational convenience in algebraic operations, the index expression depicted in figure 2.5 is signified by $t_0 \otimes_{i=1}^{k} c_i I_i$. Note that if $k = 0$, then $I = t_0$, meaning that I is a term. Each $I_i, 1 \leq i \leq k$ is referred to as a *nested subexpression.*

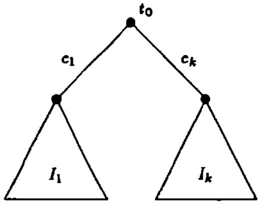

Figure 2.5: Representation of expression I

Finally, two interesting classes of index expressions are introduced. Both are depicted in figure 2.6

Definition 2.3.2

A path expression P_n of $n, (n \geq 1)$ terms is defined as:

$$\begin{aligned} P_1 &= t \\ P_n &= tcP_{n-1} \end{aligned}$$

□

Definition 2.3.3

An umbrella expression U_n of $n, (n \geq 1)$ terms is defined as

$$U_n = t_0 \otimes_{i=1}^{n-1} c_i t_i$$

□

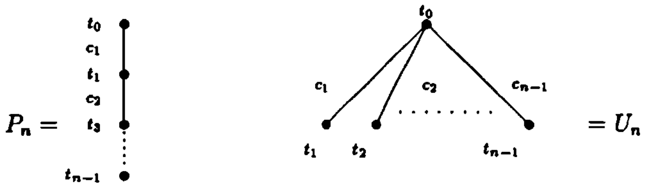

Figure 2.6: Path and umbrella expressions

2.3.1 The Expressiveness of Index Expressions

Earlier this chapter the hypothesis was put forward that better characterization of objects will lead to more effective disclosure. One way of deciding if a language L_1 is better than a language L_2 is to compare their expressive power in terms of formal language theory. The more expressive nature of the index expressions over terms and term phrases will be evident from the following observation. Let $\mathcal{L}(T, C)$ denote the language of index expressions based on a set of terms T and a set of connectors C such that the null connector, $\circ \in C$. Then, the term phrases are described by the language $\mathcal{L}(T, \{\circ\})$ and the terms by $\mathcal{L}(T, \varnothing)$. Now note that

$$\mathcal{L}(T, C) \supset \mathcal{L}(T, \{\circ\}) \supset \mathcal{L}(T, \varnothing)$$

Also note that $\mathcal{L}(\{t\}, \{c\})$ is an infinite language whereas $\mathcal{L}(\{t\}, \varnothing) = \{t\}$ is finite.

The index expressions can be related to the Boolean language of terms in the following way:

$$\mathcal{L}(T, C) \supset \mathcal{L}(T, \{\mathsf{and}, \mathsf{or}\})$$

The above relationship formally expresses Farradane's opinion that Boolean expressions only capture a "small part of the relations between terms which we try to indicate in language". Farradane regarded the Boolean and as "as almost completely unspecific", and the or as purely a mechanism for term replacement.

2.3.2 Power Index Expressions and Lithoids

Building on the notion of an index expression, the so called *power index expression* is introduced. This notion bears a strong resemblance to the power set concept: the power index expression of an index expression is the set of all its index subexpressions. First, the notion of a subexpression of an index expression is informally introduced in terms of the graphical representation: An index subexpression of a given index expression I is an index expression represented by a subtree of the tree representation of I. We will use $\underline{\subset_*}$ to denote the is-subexpression-of relation, that is, we take $I_1 \underline{\subset_*} I_2$ to denote that I_1 is a subexpression of I_2 in the sense described above. Note that the relation $\underline{\subset_*}$ is reflexive, antisymmetric and transitive; in fact, $(\mathcal{L}(T, C), \underline{\subset_*})$ is a poset.

We now define the power index expression of a given index expression more formally.

Definition 2.3.4

Let I be an index expression in a language $\mathcal{L}(T,C)$. The power index expression *of I, denoted by $\wp(I)$, is the set*

$$\wp(I) \;=\; \left\{J \mid J \,\underline{\subset\!\!\!\ast}\, I\right\}$$

where $\underline{\subset\!\!\!\ast}$ is the is-subexpression-of relation as above. □

Note that for any index expression I in a language $\mathcal{L}(T,C)$, $\wp(I) \subseteq \mathcal{L}(T,C)$ holds.

Like the power set of a given set, the power index expression of a given index expression forms a lattice where the underlying ordering relation is $\underline{\subset\!\!\!\ast}$. The top of the lattice is the index expression itself and the bottom is the empty index expression ϵ. The Hasse diagram of the power index expression of the index expression represented in figure 2.4 is depicted in figure 2.7. This thesis is very much about how the lattice structure of power index expressions can be exploited for information disclosure.

Thus far, the power index expression of a single index expression has been considered. For a set of objects, however, a core set of index expressions is generated each of which gives rise to a power index expression. These power index expressions may have a non-trivial overlap. For example, consider the power index expressions of the index expressions effective o information o retrieval and people in need of information. (See figure 2.8). By forming the union of all power index expressions for a set of objects, that is, by taking

$$\bigcup_{I \in \mathcal{I}} \wp(I)$$

where $\mathcal{I}$ is the core set of index expressions, a lattice-like structure is rendered. For the index expressions mentioned above this results in the structure shown in figure 2.8. Such a structure will be termed a *lithoid* because the associated diagram resembles a crystalline structure. One way of exploiting the lithoid for the purpose of information disclosure is as follows. If we take every vertex in the lattice as a potential focus of the searcher, then the surrounding vertices are enlargements or refinements of the context represented by the focus. The searcher can browse across the lithoid by refining or enlarging the current focus until a focus is found that fits the information need. Searching in this way has been coined *Query By Navigation* [BvdW90b][Bru90]. This notion will be further elaborated in chapter 4.

2.3.3 Automatic Indexing of Index Expressions

An important problem to address is how to arrive at the core set of index expressions from which a lithoid can be constructed. In his book Craven states that when stopwords such as the and a are omitted from the titles of documents, sections, subsections, figures etc., the resultant strings often have a form amenable to the automatic derivation of linked phrase

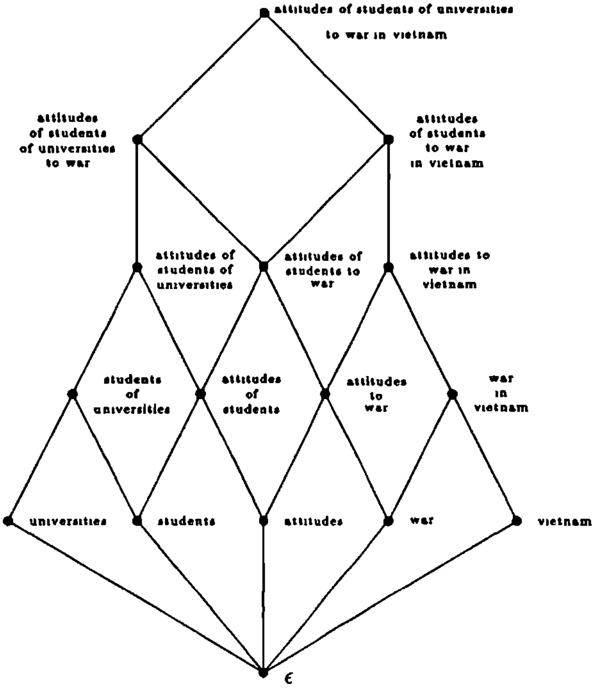
attitudes of students of universities to war in vietnam
attitudes of students of universities to war
attitudes of students to war in vietnam
attitudes of students of universities
attitudes of students to war
attitudes to war in vietnam
students of universities
attitudes of students
attitudes to war
war in vietnam
universities
students
attitudes
war
vietnam
ϵ

Figure 2.7: Example power index expression

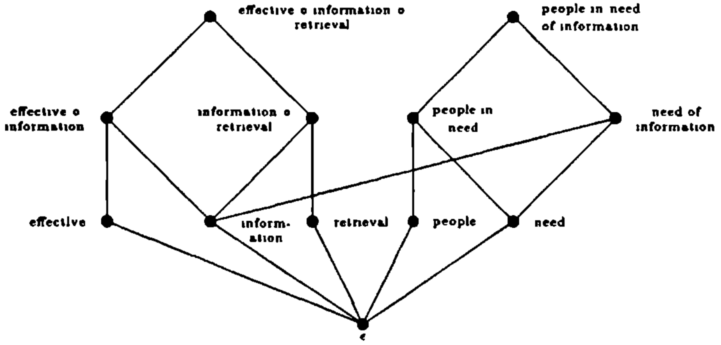

Figure 2.8: Example lithoid

indexes (an index expression variant) [Cra86]. Therefore, the characterization of an object using index expressions can be formed by the automatic derivation of such expressions from the titles of structural elements in an object. Note that titles are not always content revealing. Sometimes they are used purely for structural organization, for example, the heading *Introduction*. In that case they should be ignored.

We begin by summarizing the main algorithm from [BB91], which was used to index the titles of slides of an art-history library. After the removal of stopwords the remaining tokens of the title are successively processed in order to to attribute an interpretation to this expression by deriving a structure from it. The expectation is that the resultant structure corresponds to the interpretation that most searchers would expect. The underlying basis of structure detection is to consider the connectors as operators with an associated priority. The priority is used to decide whether the current structure is to be deepened, or broadened. To this end, a two level priority scheme over the connectors is employed based on the observation that some connectors bind terms more strongly than others; those that bind stronger lead to deepening in the structure. An example of how the structure detection algorithm works is depicted in figure 2.9 using The Elimination of Special Functions from Differential Equations as input and the two level priority scheme specified in figure 2.10.

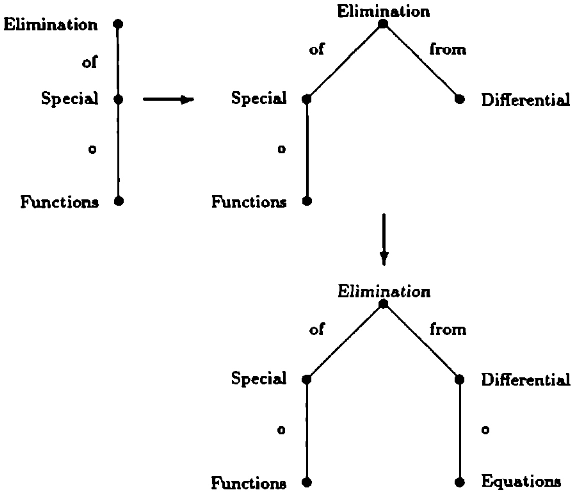

Figure 2.9: Example of structure detection

The above illustration exemplifies the underlying idea behind the automatic derivation of index expressions. Attention will now be directed to specific details of the structure detection algorithm. The input is assumed to be a string S from the language T {C T}*, where C corresponds one of the allowable connectors (see figure 2.10) and T is a token from the title in question which is not a connector and not a stop word. This language corresponds to the concrete syntax of the index expressions, brackets excluded. Preprocessing of the title being indexed renders S. For example, certain stop words are removed and connector irregularities are resolved. The input string S is consumed by processing initially the first term separately and thereafter successive connector-term pairs. Parallel to this a structure conforming to the abstract syntax of index expressions is built up. To begin with, the structure is empty, that is it corresponds to the empty index expression. This empty structure is initialized using the first term t_0 taken from S. The resultant structure is an index expression comprising t_0. More formally,

$$\mathtt{initexpr}(\epsilon, t_0) \;=\; t_0$$

The handling of a connector-term pair denoted c, t can be likened to the placing of a branch in the current tree structure which will be denoted by I. A branch is placed by a *broaden* or *deepen* function according to the priority of the connector. Broaden and deepen are functions that take an index expression and a connector-term pair as input and have an index expression as output. The following algorithm specifies the main body of the structure detection algorithm:

```
$I \leftarrow \mathtt{initexpr}(getterm(S))$
while $S \neq \epsilon$ do
    $c, t \leftarrow getconntermpair(S)$
    if $isprio0connector(c)$ then
        $I \leftarrow \mathtt{deepen}(I, c, t)$
    else
        $I \leftarrow \mathtt{broaden}(I, c, t)$
    fi
od
```

The function *isprio0connector* returns true if connector c is a higher priority connector. (See figure 2.10). The *broaden* and *deepen* operations are defined recursively as follows where the current structure is denoted by $t_0c_1(I_1)\ldots c_k(I_k)$. This is a handy representation for denoting the effects on the structure: $I_i, 1 \leq i \leq k$ are nested index subexpressions.

$$\begin{aligned}
\mathtt{deepen}(t_0, c, t) &= t_0c(t)\\
\mathtt{deepen}(t_0c_1(I_1)\ldots c_k(I_k), c, t) &= t_0c_1(I_1)\ldots c_k(\mathtt{deepen}(I_k, c, t))
\end{aligned}$$

$$\begin{aligned}
\mathtt{broaden}(t_0, c, t) &= \mathtt{deepen}(t_0, c, t)\\
\mathtt{broaden}(t_0c_1(I_1)\ldots c_k(I_k), c, t) &= t_0c_1(I_1)\ldots c_k(I_k)c(t)
\end{aligned}$$

Note that when the current structure I comprises only a term, *broaden* and *deepen* are identical operations. Furthermore, deepening occurs in the rightmost nested subexpression and broadening takes place at the root of the structure.

After an initial test of this algorithm on the titles of the CACM document collection, the above structure detection algorithm was found to produce well formed structures approximately ninety percent of the time. The priority scheme used is depicted in figure 2.10.

Most of the ill formed structures involved improper handling of so called term sequences. A *term sequence* $\mathcal{T}_l$ is a path expression of length l which involves only the null connector. Term sequences are constructed by a process of continual deepening, the motivation being that null connectors bind terms strongly. Taking $\mathcal{T}_1 = t$, then $\mathcal{T}_l = \mathtt{deepen}(\mathcal{T}_{l-1}, \circ, t), l > 1$.

Priority	Connector
0	∘
	about
	and
	as
	for
	including
	of
	or
	see
	with
1	are
	around
	at
	behind
	between
	by
	from
	in
	into
	is
	on
	over
	through
	to
	under
	using
	within
	without

Figure 2.10: Two level connector priority scheme

It was observed that after an initial term sequence the structure should be deepened, regardless of the priority of the connector. This is because the first connector encountered after the end of the term sequence very often refers to the last term in the sequence. For example, the structure proposed ∘ (interpretation) in (ALGOL) is ill formed as the *interpretation* is in the language *ALGOL*, implying that interpretation and ALGOL should be related via the connector in. In other words, the path expression proposed ∘ (interpretation in (ALGOL)) is the correct structure. The structure detection algorithm was improved to take the above problem into account.

Other ill formed structures came about because broadening at the root of the structure is not always appropriate. Consider the structure depicted in figure 2.11. If the next two tokens in the stream are the connector in and the term language, then broadening at the root establishes a relationship between the terms handling and language. Note that establishing a relationship between the terms identifiers and language is clearly more appropriate. (See figure 2.12).

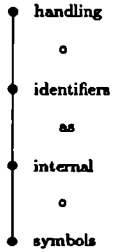

Figure 2.11: Structure before broadening

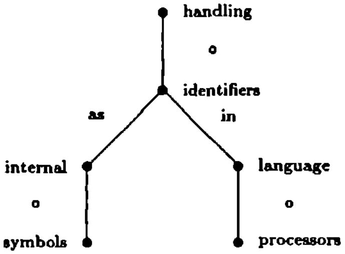

Figure 2.12: Complete structure

On the basis of cases such as the previous example, a new broading heuristic was developed which broadens the structure at the *father* of first term in a term sequence. More formally,

$$\texttt{broaden}(t(\Gamma)c_x(\tilde{t}c_y(\mathcal{T}_i)), c_z, \hat{t}) \;=\; t(\Gamma)c_x(\texttt{broaden}(\tilde{t}c_y(\mathcal{T}_i), c_z, \hat{t})$$

This broadening heuristic is depicted in figure 2.13. The father of the term phrase $\mathcal{T}_i$ is denoted by $\tilde{t}$. If no father exists, broadening occurs at the root as specified earlier.

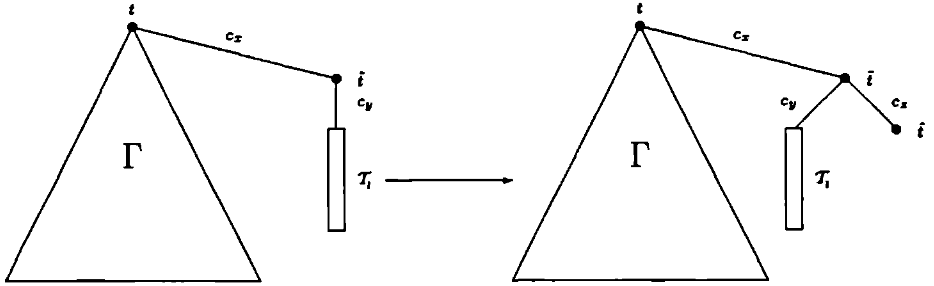

Figure 2.13: Enhanced broadening heuristic

The above improved structure detection algorithm can be implemented efficiently because only a single pass of the input string is necessary and the broadening and deepening heuristics do not involve any complex analysis. On the negative side, however, the restriction to titles results in incomplete characterizations. This restriction was taken because titles and headings are often in a form which permits a ready transformation to index expressions. The automatic derivation of index expressions from general text requires more advanced parsing techniques; a problem which was beyond the scope of this thesis. A potential solution to this problem may be found in the work of the *SIMPR* Esprit project [Sme90]. The syntactic trees devised in that project are not unlike index expressions.

A note on the and connector

Consider the title *The attitudes of administrators, students and teachers to courses in universities.* The indexing algorithm of the previous section would produce the index expression depicted in figure 2.14.

The and connector should not be confused with logical conjunction; it should be interpreted as a *linguistic* conjunctor. Furthermore, and and commas in the input string really suggest the existence of three separate index expressions, namely

attitudes of administrators to courses in universities
attitudes of students to courses in universities
attitudes of teachers to courses in universities

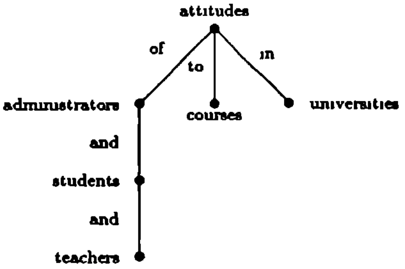

Figure 2.14: Example index expression involving the and connector

One way of allowing the generation of such expressions is to adopt a network representation. By way of illustration, see figure 2.15 which is taken from [Cra88]. The disadvantage of the network representation is that it is more difficult than tree structures to derive automatically.

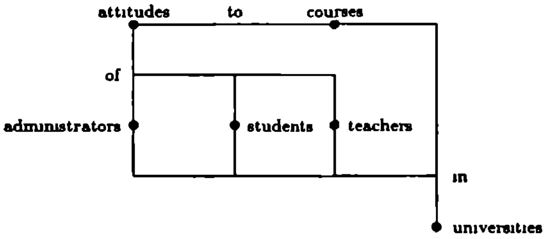

Figure 2.15: Network representation of an index expression

2.3.4 Properties of Index Expressions

Given an arbitrary index expression, how large is $\wp(I)$? This is a pertinent question as the power index expression forms an integral part of an information disclosure mechanism to be featured in ensuing chapters. The analysis of the properties of index expressions will proceed on the basis of the following definition.

Definition 2.3.5

Given the index expression $I = t_0 \bigotimes_{i=1}^{k} c_i I_i$, then

1. *$n(I)$ is the number of terms in $I, n(I) = 1 + \sum_{1 \leq i \leq k} n(I_i)$*
2. *$I\{m\} = \{J \mid J \subseteq I \wedge n(J) = m\}$, the set of subexpressions of I containing m terms.*

3. $I(m) = |I\{m\}|$
4. $\tau(I) = I\{1\}$, *the set of terms of I*
5. $\lambda(I) = t_0$, *the lead term*
6. I^t *denotes the largest subexpression* J *of* I *beginning with term* t. *(Note that* $I^{\lambda(I)} = I$ *and if* t *is a leaf term then* $I^t = t$*).*
7. $\Lambda(I) = \{J \mid J \overset{*}{\subseteq} I \wedge \lambda(J) = \lambda(I)\}$, *the lead expressions*
8. $l(I) = |\Lambda(I)|$
9. $\Lambda\{I, m\} = \left\{J \mid J \in \Lambda(I) \wedge n(J) = m\right\}$, *the lead expressions containing m terms.*
10. $\Lambda(I, m) = |\Lambda\{I, m\}|$
11. $\wp^+(I) = \wp(I) - \{\epsilon\}$, *the information bearing subexpressions of I*
12. $p(I) = |\wp^+(I)|$

□

Furthermore, as we are interested in worst case behaviour it is assumed that the terms within I are distinct ($n(I) = |\tau(I)|$). We begin by defining $\wp^+(I)$ in a way which will facilitate the counting of its elements.

Theorem 2.3.1 Power Index Expression

Let $I = t_0 \bigotimes_{i=1}^{k} c_i I_i$ be an index expression, then

$$\wp^+(I) = \Lambda(I) \cup \bigcup_{1 \le i \le k} \wp^+(I_i)$$

Proof:

It follows from definition 2.3.4 that all the subexpressions of $I_i, 1 \le i \le k$ are in the power index expression of I. Therefore, $\wp^+(I) \supseteq \bigcup_{1 \le i \le k} \wp^+(I_i)$. The remaining subexpressions are precisely those that begin with the lead term of I. From definition 2.3.5 states that these expressions comprise the set $\Lambda(I)$. □

The lead expressions of I can be defined in terms of the lead expressions of the nested subexpressions of I. This works as follows: All combinations of nested subexpressions generate the lead expressions of I by appropriately concatenating the elements of the nested lead expressions. By way of illustration; if $I = t_0 c_1 A c_2 B c_3 C$ where A, B and C denote the nested subexpressions of I. There are eight combinations of 3-set $\left\{A, B, C\right\}$.

$$\varnothing, \left\{A\right\}, \left\{B\right\}, \left\{C\right\}, \left\{A, B\right\}, \left\{A, C\right\}, \left\{B, C\right\}, \left\{A, B, C\right\}$$

These combinations underly the way sets of lead expressions of the *nested* subexpressions can be concatenated with each other.

$$\varnothing, \Lambda(A), \Lambda(B), \Lambda(C), \Lambda(A)\Lambda(B), \Lambda(A)\Lambda(C), \Lambda(B)\Lambda(C), \Lambda(A)\Lambda(B)\Lambda(C)$$

Introducing the lead term t_0 and the connectors attached to t_0 results in all the lead expressions of I.

$$\begin{aligned}
&\Lambda(I) = \\
&\quad \left\{t_0\right\} \cup \\
&\quad \left\{t_0c_1\right\}\Lambda(A) \cup \left\{t_0c_2\right\}\Lambda(B) \cup \left\{t_0c_3\right\}\Lambda(C) \cup \\
&\quad \left\{t_0c_1\right\}\Lambda(A)\left\{c_2\right\}\Lambda(B) \cup \left\{t_0c_1\right\}\Lambda(A)\left\{c_3\right\}\Lambda(C) \cup \left\{t_0c_2\right\}\Lambda(B)\left\{c_3\right\}\Lambda(C) \cup \\
&\quad \left\{t_0c_1\right\}\Lambda(A)\left\{c_2\right\}\Lambda(B)\left\{c_3\right\}\Lambda(C)
\end{aligned}$$

The above discussion forms the intuition behind the following theorem which is the general case.

Theorem 2.3.2 The Lead Expressions

Let $I = t_0 \bigotimes_{i=1}^{k} c_i I_i$ be an index expression, then

$$\Lambda(I) = \left\{t_0\right\} \bigodot_{i=1}^{k} \left\{ \left\{\epsilon\right\} \cup \left\{c_i\right\} \cdot \Lambda(I_i) \right\}$$

where $\bigodot$ and $\cdot$ denote expression concatenation analogous to formal language theory

Proof:

(by induction on k, the number of nested subexpressions)

Basis step: $k = 0$

If $k = 0$, then there are no nested subexpressions, so $I = t_0$. Therefore, $\Lambda(I)$ contains a single lead expression, namely the lead term t_0.

Induction Hypothesis:

Assume that for all $0 \leq j \leq k$, where k denotes the number of nested subexpressions, that $\Lambda(I) = \left\{t_0\right\} \bigodot_{i=1}^{j} \left\{ \left\{\epsilon\right\} \cup \left\{c_i\right\} \cdot \Lambda(I_i) \right\}$

Induction Step:

It will now be shown that by the addition of a nested subexpression to I resulting in $\tilde{I}$ that $\Lambda(I) = \left\{t_0\right\} \bigodot_{i=1}^{k+1} \left\{ \left\{\epsilon\right\} \cup \left\{c_i\right\} \cdot \Lambda(I_i) \right\}$ (See figure 2.16)

The lead expressions of $\tilde{I}$ comprise all the lead expressions of I. Furthermore, with the addition of the nested subexpression I_{k+1}, all expressions in $\Lambda(I)$ can be concatenated with all the lead expressions in $\Lambda(I_{k+1})$ via the connector c_{k+1}. That is,

$$\Lambda(\tilde{I}) \quad = \quad \Lambda(I) \cup \Lambda(I) \cdot \left\{c_{k+1}\right\} \cdot \Lambda(I_{k+1})$$

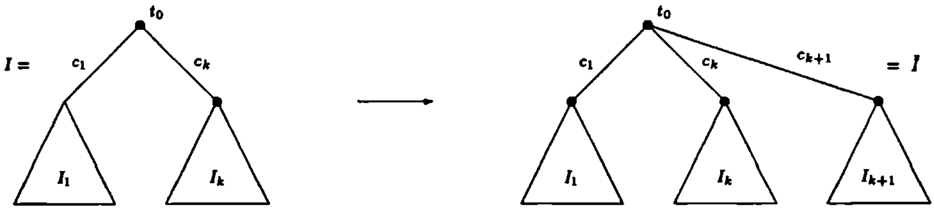

Figure 2.16: Addition of a nested subexpression

$$= \Lambda(I)\cdot\left\{\{\epsilon\}\cup\{c_{k+1}\}\cdot\Lambda(I_{k+1})\right\}$$

$$= \langle\texttt{Induction Hypothesis}\rangle$$

$$\underbrace{(\{t0\}\bigodot_{i=1}^{k}\left\{\{\epsilon\}\cup\{c_i\}\cdot\Lambda(I_i)\right\})}_{\Lambda(I)}\cdot\left\{\{\epsilon\}\cup\{c_{k+1}\}\cdot\Lambda(I_{k+1})\right\}$$

$$= \{t0\}\bigodot_{i=1}^{k+1}\left\{\{\epsilon\}\cup\{c_i\}\cdot\Lambda(I_i)\right\}$$

□

In order to determine the magnitude of $\wp^+(I)$ using theorem 2.3.1, the number of lead expressions of I must be determined.

Theorem 2.3.3 How many Lead Expressions
Let $I = t_0 \bigotimes_{i=1}^{k} c_i I_i$ be an index expression, then

$$l(I) = \prod_{i=1}^{k}(1 + l(I_i))$$

Proof:
We count the numer of words of the language $\Lambda(I)$ (See theorem 2.3.2). If $k = 0$ then $\Lambda(I) = \{t_0\} \Rightarrow l(I) = 1$. If $k > 0$, then each component $\left\{\{\epsilon\}\cup\{c_i\}\cdot\Lambda(I_i)\right\}, 1 \le i \le k$ consists of $1 + l(I_i)$ words. These components are concatenated with each other, therefore $l(I) = \prod_{i=1}^{k}(1 + l(I_i))$. □

The following lemma is a direct consequence of the restriction that terms are distinct within an index expression. As a result, there can be no overlap between the lead expressions and the power index expressions of component nested subexpressions. This result will be used shortly.

Lemma 2.3.1 Let $I = t_0 \bigotimes_{i=1}^{k} c_i I_i$ be an index expression, then

$$\forall_{1\le i\le k}[\wp^+(I_i)\cap\Lambda(I)=\varnothing]$$

The groundwork has now been laid to express the size of the power index expression.

Theorem 2.3.4 Power index expression size
Let $I = t_0 \otimes_{i=1}^{k} c_i I_i$ be an index expression, then

$$p(I) = l(I) + \sum_{i=1}^{k} p(I_i)$$

Proof:

$$\begin{aligned}
p(I) &= \langle \texttt{Theorem 2.3.1} \rangle \\
&\quad \left| \Lambda(I) \cup \bigcup_{1 \leq i \leq k} \wp^{+}(I_i) \right| \\
&= \langle \texttt{Lemma 2.3.1} \rangle \\
&\quad |\Lambda(I)| \cup \left| \bigcup_{1 \leq i \leq k} \wp^{+}(I_i) \right| \\
&= \langle \texttt{Definition 2.3.5, item 12} \rangle \\
&\quad l(I) + \sum_{1 \leq i \leq k} p(I_i)
\end{aligned}$$

□

The result of the above theorem can be simplified further using the observation that if expression J is a term, then the set of information bearing subexpressions of J is simply the singleton set containing J, which is also by definition equal to $\Lambda(J)$. Thus by recursive application of theorem 2.3.1 an expression must result consisting of a union of Λ's.

Lemma 2.3.2 Let $I = t_0 \otimes_{i=1}^{k} c_i I_i$ be an index expression, then

$$\wp^{+}(I) = \bigcup_{t \in \tau(I)} \Lambda(I^{t})$$

Proof:

($\Rightarrow$)

$$J \in \wp^{+}(I) \Rightarrow \lambda(J) \in \tau(I) \Rightarrow J \in \Lambda(I^{\lambda(J)})$$

($\Leftarrow$)

$$J \in \bigcup_{t \in \tau(I)} \Lambda(I^{t}) \Rightarrow \exists_{\bar{t} \in \tau(I)} [J \in \Lambda(I^{\bar{t}})] \Rightarrow J \in \wp^{+}(I) \text{ as } I^{\bar{t}} \underline{\subset_{*}} I$$

□

The size of the power index expression can be calculated as follows.

Lemma 2.3.3 Let I be an index expression, then

$$p(I) = \sum_{t \in \tau(I)} l(I^t)$$

Proof:

There can be no overlap between the sets $\Lambda(I^t), t \in \tau(I)$ because each term t is distinct in I. □

The significance of lemma 2.3.2 and lemma 2.3.3 is that they underly an efficient mechanism for generating the power index expression. The function Λ need only be implemented and unleashed over I. As there is no overlap between component lead expression sets, the union operation can be efficiently implemented by list concatenation. Such a strategy underlies the practical experiments involving power index expressions documented in this thesis.

Another useful result of lemma 2.3.3 is that it offers a simple way of calculating $p(I)$ by hand. For example, if I is the expression depicted in figure 2.12, then one need only determine the number of lead expressions contributed by each nested subexpression and then sum them up. This is done by working upward from the leaves and applying theorem 2.3.3. (See figure 2.17). In this case $p(I) = 10 + 9 + 2 + 2 + 1 + 1 = 25$

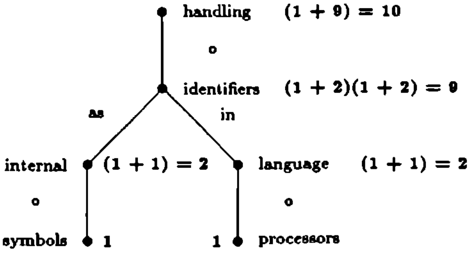

Figure 2.17: Manual determination of $p(I)$

2.3.4.1 The lower bound of power index expression size

From our discussions earlier we know that the size of the power index expression is in fact dependent on the number of lead expressions. Theorem 2.3.3 gives an expression for the number of lead expressions. This expression shows that the number of leads is determined by a product which, in turn, is dependent on the number of connectors connected to the lead term of I. The number of lead expressions decreases as the number of such connectors decreases. The path expression (definition 2.3.2) is the extreme of this and therefore it can be supposed that these expressions generate the smallest power index expressions. To prove that this is in fact the case is the goal of this section. We first begin by counting the lead expressions in a path.

Lemma 2.3.4 Leads of a Path

Let P_n be a path expression of n terms, then

$$l(P_n) = n$$

Proof:

$$\begin{aligned}
l(P_n) &= \langle \texttt{Theorem 2.3.3 with } k = 1 \rangle \\
&\quad 1 + l(P_{n-1}) \\
&= 1 + \underbrace{1 + l(P_{n-2})}_{l(P_{n-1})} \\
&= \underbrace{1 + 1 + \ldots + 1}_{n-1} + l(P_1) \\
&= n
\end{aligned}$$

□

The magnitude of a power path expression is established as follows.

Theorem 2.3.5 Power path expression size

Let P_n be a path expression of n terms, then

$$p(P_n) = \frac{n(n+1)}{2}$$

Proof:

$$\begin{aligned}
p(P_n) &= \langle \texttt{Theorem 2.3.4 with } k = 1 \rangle \\
&\quad p(P_{n-1}) + l(P_n) \\
&= \langle \texttt{Theorem 2.3.3 with } k = 1 \rangle \\
&\quad p(P_{n-1}) + (1 + l(P_{n-1})) \\
&= \underbrace{p(P_{n-2}) + (1 + l(P_{n-2}))}_{p(P_{n-1})} + (1 + l(P_{n-1})) \\
&= p(P_1) + \sum_{i=1}^{n-1} (1 + l(P_i)) \\
&= 1 + (n-1) + \sum_{i=1}^{n-1} l(P_i) \\
&= \langle \texttt{Lemma 2.3.4} \rangle \\
&\quad n + \frac{(n-1)n}{2} \\
&= \frac{n(n+1)}{2}
\end{aligned}$$

□

The following theorem establishes the lower bound of power index expression size. Shortly thereafter it will be shown that power path expressions have this lower bound.

Theorem 2.3.6 Lower bound of power index expression size
Let I be an index expression of n terms, then

$$p(I) \geq \frac{n(n+1)}{2}$$

Proof:

(by induction on n, the number of terms in I)
Basis step:

$$n = 0 \Rightarrow I = \epsilon \Rightarrow p(I) = 0$$

Induction Hypothesis:

Assume that for all index expressions I of i terms, $0 \leq i \leq n$ that $p(I) \geq \frac{i(i+1)}{2}$

Induction Step:

It will now be shown that by extending expression I, which has n terms, to expression $\tilde{I}$ by adding a term in an arbitrary way, the minimal size of $p(\tilde{I})$ is $\frac{(n+1)(n+2)}{2}$ (See figure 2.18).

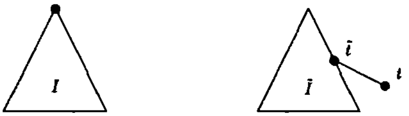

Figure 2.18: Adding a branch

Note that a subexpression of $\tilde{I}$ is either a subexpression of I or it is a subexpression containing the added term t. From the shortly to be presented Help Lemma 2.3.5, we will see that there are at least $n+1$ expressions in the latter category. As a result,

$$\begin{aligned} p(I_{n+1}) &\geq p(I_n) + (n+1) \\ &\geq \langle\texttt{Induction Hypothesis}\rangle \\ &\quad \frac{n(n+1)}{2} + (n+1) \\ &= \frac{n^2+n+2n+1}{2} \\ &= \frac{(n+1)(n+2)}{2} \end{aligned}$$

□

Corollary 2.3.1 Power path expressions are minimal

Proof:

Follows directly from theorems 2.3.6 and 2.3.5 □

The following Help Lemma specifies the minimal participation of a term in the power index expression.

Lemma 2.3.5 Help Lemma

Let I be an index expression of n terms, then

$$t \in \tau(I) \quad \Rightarrow \quad \left|\left\{J \in \wp(I) \text{ where } t \subsetneqq J\right\}\right| \geq n$$

Proof:

Sketch: Use induction on the number of terms. When I is extended to $\tilde{I}$ via term t (see figure 2.18), then t is guaranteed to be involved in at least $n+1$ subexpressions of $\tilde{I}$. □

2.3.4.2 The upper bound of power index expression size

The reason why power path expressions are minimal lies in the fact that the "fan out" of terms in the expression is minimal. In umbrella expressions (definition 2.3.3), the fan out of the lead term is maximal. This section explores power umbrella expression size and shows that this in fact constitutes the upper bound of power index expression size. As was the case with the path expressions, the lead expressions of an umbrella will first be counted.

Lemma 2.3.6 Leads of an Umbrella

Let U_n be an umbrella expression of n terms, then

$$l(U_n) = 2^{n-1}$$

Proof:

Follows directly from theorem 2.3.3 with $k = n - 1$ □

The magnitude of the power umbrella expression can now be established as follows.

Theorem 2.3.7 Power umbrella expression size

Let U_n be an umbrella expression of n terms, then

$$p(U_n) \quad = \quad 2^{n-1} + (n-1)$$

Proof:

$$\begin{aligned}
p(U_n) &= \langle\texttt{Theorem 2.3.4 with } k = n-1\rangle \\
&\quad l(U_n) + \sum_{i=1}^{n-1} p(t_i) \\
&= \langle\texttt{Lemma 2.3.6}\rangle \\
&\quad 2^{n-1} + \sum_{i=1}^{n-1} p(t_i) \\
&= 2^{n-1} + n - 1
\end{aligned}$$

□

The above theorem states that the fanout of the lead term produces a power index expression with a size which is exponential in size with respect to the number of terms n. This turns out to be the upper bound as the following lemma and theorem prove. The lemma proves that the promotion of a nested subexpression never generates less expressions in the power index expression.

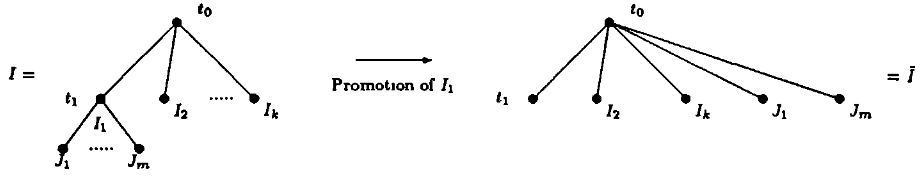

Figure 2.19: Promotion of a Nested Subexpression

Lemma 2.3.7 Help Lemma: The promotion of a nested subexpression always produces as much or more

Let $I = t_0c_1(t_1 \bigotimes_{i=1}^{m} c_iJ_i) \bigotimes_{i=2}^{k} c_iI_i$ and $\tilde{I} = t_0c_1(t_1) \bigotimes_{i=2}^{k} c_iI_i \bigotimes_{i=1}^{m} c_iJ_i$ be index expressions, then $p(\tilde{I}) \geq p(I)$

Proof:

It will be shown that $p(\tilde{I}) - p(I) \geq 0$. (See figure 2.19).

$$\begin{aligned}
p(I) &= \langle\texttt{Theorem 2.3.4}\rangle \\
&\quad l(I) + \sum_{i=1}^{k} p(I_i)
\end{aligned}$$

$$
\begin{aligned}
&= l(I) + p(I_1) + \sum_{i=2}^{k} p(I_i) \\
&= \langle \texttt{Theorem 2.3.4} \rangle \\
&\quad \underbrace{l(I) + l(I_1) + \sum_{i=1}^{m} p(J_i)}_{p(I_1)} + \sum_{i=2}^{k} p(I_i) \\
p(\tilde{I}) &= \langle \texttt{Theorem 2.3.4} \rangle \\
&\quad l(\tilde{I}) + p(t_1) + \sum_{i=2}^{k} p(I_i) + \sum_{i=1}^{m} p(J_i)
\end{aligned}
$$

Therefore,

$$
p(\tilde{I}) - p(I) = l(\tilde{I}) + 1 - l(I) - \underbrace{l(I_1)}_{X}
$$

Now focussing on counting the respective lead expressions,

$$
\begin{aligned}
l(\tilde{I}) &= \langle \texttt{Theorem 2.3.3} \rangle \\
&\quad (1+1) \underbrace{\prod_{i=2}^{k} (1 + l(I_i))}_{Y} \underbrace{\prod_{i=1}^{m} (1 + l(J_i))}_{X} \\
&= 2XY \\
l(I) &= \langle \texttt{Theorem 2.3.3} \rangle \\
&\quad \prod_{i=1}^{k} (1 + l(I_i)) \\
&= (1 + l(I_1)) \prod_{i=2}^{k} (1 + l(I_i)) \\
&= (1 + X) \prod_{i=2}^{k} (1 + l(I_i)) \\
&= (1 + X)Y
\end{aligned}
$$

Now,

$$
\begin{aligned}
p(\tilde{I}) - p(I) &= 2XY - (1 + X)Y - X + 1 \\
&= 2XY - Y - XY - X + 1 \\
&= XY - X - Y + 1 \\
&= X(Y - 1) - (Y - 1) \\
&= (X - 1)(Y - 1)
\end{aligned}
$$

The last result implies $p(\tilde{I}) - p(I) \geq 0$ as X, which denotes the number of lead expressions of I_1 is certainly greater than zero and for similar reasons $Y > 0$. (See figure 2.19). □

Corollary 2.3.2 Power umbrella expressions are maximal

Let I be an index expression of n terms, then

$$p(I) \leq 2^{n-1} + n - 1$$

Proof:

This follows from lemma 2.3.7 and theorem 2.3.7. Continual promotion in an arbitrary index expression will result in an umbrella expression. □

2.3.4.3 The bounds of growth of power index expressions

Theorem 2.3.6 and corollary 2.3.2 give the bounds of the power index expression size. These bounds are depicted graphically in figure 2.20.

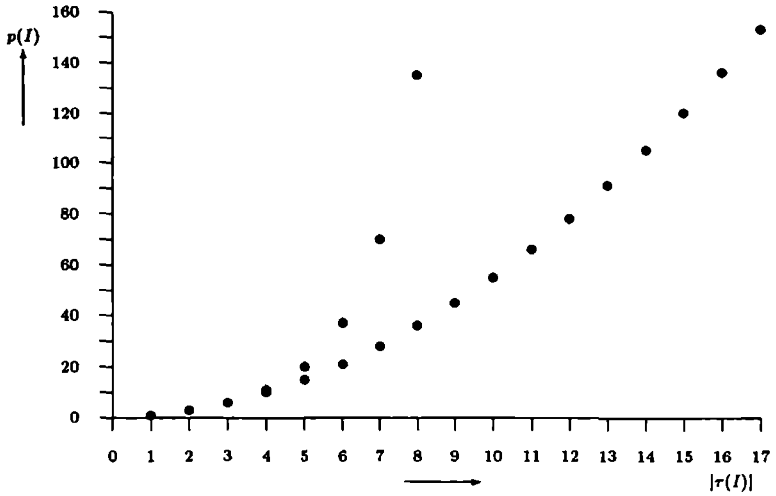

Figure 2.20: Power index expression growth

2.3.5 Growth Behaviour of Lithoids

Recall that a lithoid is a union of power index expressions. From a practical standpoint it is important to have some indication of the growth characteristics of lithoids. In this section growth analyses of two lithoids are featured. The first lithoid was derived from the CACM collection of 3204 documents and the second from the first 750 documents of the Cranfield collection. These document collections are often used for research in information disclosure. In both cases, the titles of the documents in the collection were extracted and indexed using the improved algorithm of section 2.3.3. In each case a core set $\mathcal{I}$ of index expressions resulted from which a lithoid was constructed.

The theoretical arguments of the previous section show that umbrella expressions spawn the largest power index expressions. The upper bound of lithoid size, therefore, is if $\mathcal{I}$ consists solely of umbrella expressions and moreover that there is no overlap between respective power umbrella expressions. Assuming that on average a core index expression consists of x terms and there are m core expressions ($|\mathcal{I}| = m$), the upper bound of lithoid size can be approximated as follows:

$$\begin{aligned}
\left|\bigcup_{U \in \mathcal{I}} \wp(I)\right| &= \langle \texttt{no overlap} \rangle \\
&= \sum_{U \in \mathcal{I}} |\wp(U)| \\
&= \sum_{U \in \mathcal{I}} p(U) \\
&\approx \langle \texttt{thm:maxpowerexp} \rangle \\
&\quad \sum_{1 \le i \le m} 2^{x-1} + x - 1 \\
&\approx m2^{x-1} + mx - m
\end{aligned}$$

We will refer to this upper bound as the worst case.

An important question is how the actual size of a typical lithoid compares with the worst case. During the construction of the Cranfield lithoid, its size was recorded after the processing of 200, 400, 600 and 750 titles. In addition, a value of $x = 7.5$ was calculated for the Cranfield titles so the worst case could be analyzed. On the basis of these measurements, growth curves can be plotted. These are depicted in figure 2.21. This figure shows that the actual growth is between the worst case and the growth of the unary expressions (the terms). It is interesting to compare the actual growth with the term growth because most current disclosure systems use only terms. In this way an indication can be gained of the space overhead of a lithoid based disclosure system and a term based equivalent. Note that from roughly 1330 terms more than 26000 index expressions have been generated. From the above figure we conclude that the speed at which the lithoid grows is worrying from a storage overhead perspective[1]. Similar growth was experienced in [BB91] in the context of an art history lithoid.

The growth of the CACM lithoid proved markedly less than that of the Cranfield but is nonetheless substantial. (See figure 2.22) The reason for the difference is that the titles from the CACM collection contain on average 3.5 terms as opposed to 7.5 in the Cranfield collection. On average a power index expression in the CACM lithoid contained only ten expressions compared to 41 expressions in the Cranfield lithoid.

[1] Out of this concern, a strategy was developed to dynamically generate the necessary part of the lithoid according to the current context built up by the searcher. This strategy was incorporated in the workbench produced by Esprit project APPED.

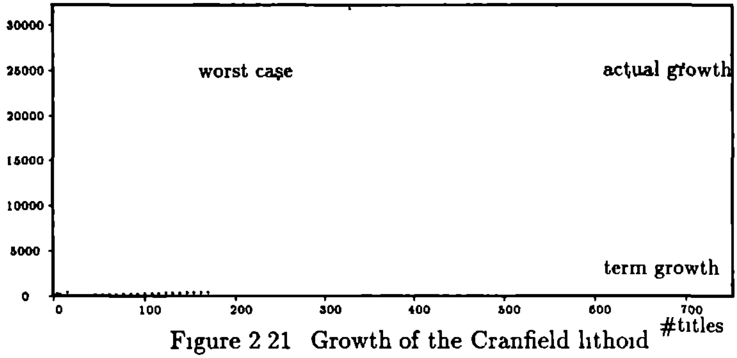

Figure 2 21 Growth of the Cranfield lithoid

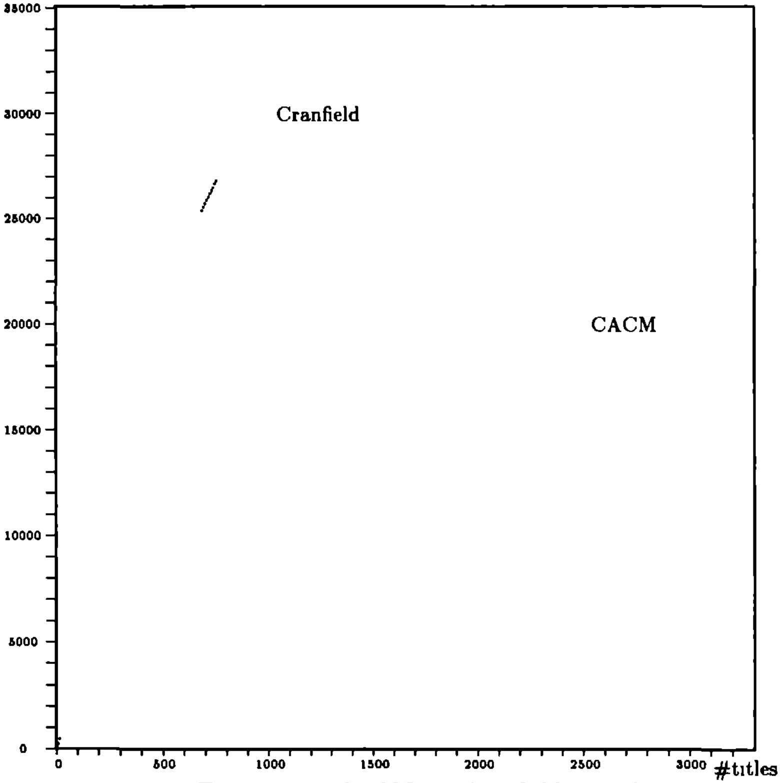

Figure 2 22 CACM vs Cranfield growth

The distribution of the index expressions over the terms is interesting because it reflects the general topology of the lithoid. Figure 2.23 shows that binary and ternary index expressions are easily the most frequently occurring in the CACM lithoid, whereas the Cranfield lithoid has a flatter peak. This is due to the Cranfield titles being longer. Keep in mind that the height of the peaks is not relevant due to the different sizes of the respective core index expression sets.

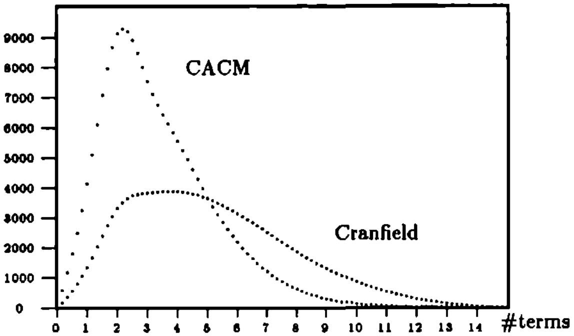

Figure 2.23: Distribution of expressions over terms

One would expect that there would be some overlap between the power index expressions that comprise a given lithoid because it is extremely unlikely that titles in a document collection don't share some terms. Such overlap can be exploited for information disclosure as it furnishes the possibility to establish a relationship between expressions, in particular between a characterization and a request. In chapter 3, we will show how expression overlap within a lithoid can be used to drive context free plausible inference over index expressions. Furthermore, a shared expression resides in different contexts denoted by the expressions which it is part of. Showing these contexts to the searcher is a useful guide to help him or her clarify their information need. This aspect will be dealt with in length in chapter 4. In order to scrutinize the overlap factor in a lithoid, the concept of the *uniqueness* of an index expression is introduced. An index expression is termed *unique* if it is a subexpression of a single core expression.

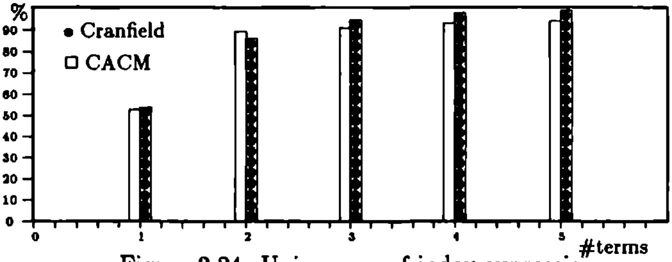

Figure 2.24: Uniqueness of index expressions

Figure 2.24 depicts the uniqueness of index expressions against the number of terms they contain. It is not surprising that uniqueness increases in relation to the number of the underlying terms because the specificity of an index expression increases with its length, thereby reducing the likelihood of it being shared. Note that uniqueness increases rapidly. This opens possibilities for optimizing the storage overhead of the lithoid. Basically, unique index expressions which are not core expressions are redundant and could be ignored by the plausible inference mechanism. This point will be further addressed in chapter 5. In both lithoids, more than ninety percent of ternary index expressions are unique. As a final note, [Ros91] contains additional analyses of the CACM and Cranfield lithoids.

Chapter 3

Logic-based Information Disclosure Machines

If people prefer to reason qualitatively, why should machines reason with numbers? Probabilities are summaries of knowledge that is left behind when information is transferred to a higher level of abstraction. The summaries can be encoded logically or numerically; logic enjoys the advantages of parsimony and simplicity, while numbers are more informative and sometimes necessary

Judea Pearl - Probabilistic Reasoning in Intelligent Systems

3.1 Dubious Assumptions in Matching

In the previous chapter the important issue of object characterization was broached. Indeed, some regard the question of how to characterize objects for effective information disclosure as the most important theme in information retrieval research [Bla90] [Far80a] [Far80b]. Underlying this theme is the assumption that by coming up with better representations of objects, the information disclosure effectiveness will be improved. The language of the index expressions was argued as being more powerful than the languages of terms and term phrases due to its superior expressivity.

Characterizations, however, are not an end in themselves. They must be used by a disclosure machine to determine the relevance of objects in response to a request. This process is termed *matching*. (See figure 1.1). In the introduction it was proposed that the disclosure of information is in part problematic due to the matching process being based on dubious assumptions. In this chapter we will substantiate this hypothesis and put forward a paradigm for the matching process, which is largely free from these assumptions.

3.1.1 A Simple Disclosure Mechanism

Assume that each object in a given object base is characterized by itself. In terms of the Information Disclosure Principle, this is at first sight not an unreasonable choice because such a characterization distinguishes the object perfectly. This choice also has the practical advantage that there are no indexing costs. Assume also that the matching process is driven by the rule that if the request is contained in the object, then the object is relevant. An example of a such disclosure mechanism is the **grep** utility under UNIX.

A serious problem with this mechanism is the inherent *Closed World Assumption*, namely if the request is *not* contained in the object, then the object is deemed *not* to be relevant. This assumption can be extremely dubious. For example, if the request q =**programming language**, then the above machine will not return the modified report on the algorithmic language ALGOL-60. The example demonstrates that the Closed World Assumption can adversely affect recall.

Another important issue regarding the matching process is that it should be efficient. For example, unleashing the above disclosure mechanism on the Lockheed database of ten million documents with the request q = **stars**, implies a response time of approximately five and a half days[1].

3.2 Logic-based Information Disclosure Machines

In state-of-the-art disclosure mechanisms, the matching process is typically driven by empirical relations between query terms and object characterizations. For example, in the Vector Space approach [Sal83] both object characterizations and requests are modelled as vectors in an n-dimensional space spanning the underlying finite language of terms. Matching is specified by an equation which determines the angle between a vector modelling an object characterization and another vector modelling the request. If the angle is small the corresponding object is deemed relevant and returned.

The empirical approaches have the inherent assumption that they do not in any substantial way need to incorporate the meaning of an object in the matching process. Much of the research in information disclosure over the last thirty years has been directed at maximizing disclosure effectiveness within this limiting assumption. It seems that the limit of exploitation of empirical models has been reached [Rij86a]. In order to transcend this barrier, considerable amounts of recent research has been directed at incorporating some notion of semantics in the matching process [SvR90][Wea88][BC89] [CN90]. The crucial questions with regard to semantics in information disclosure are:

- how the meaning of an object should be represented
- can this representation be automatically derived

[1] Assuming the Knuth-Morris-Pratt string search algorithm [BY89] on english texts

- given that the meaning is available, how can it be used to render effective disclosure.

The logic-based approach to information disclosure presented in this dissertation is motivated by the work of Van Rijsbergen [Rij86b][Rij86a][Rij89]. He promotes a view in which the matching between a request and an object characterization is founded on the notion of logical inference. In the logic-based approach to information disclosure an object is assumed to have a formal semantics in the form of a set of axioms. Each axiom describes or characterizes a part of the content of the object. In logic, a *model* is an interpretation in which all given axioms hold. In this sense, an information object O can be said to form a model of its associated axioms A, denoted $O \models A$. In formal theories a well-formed formula W can be deduced, or proven, from a set A of axioms by applying so called rules of inference; the provability of W from A is denoted as $A \vdash W$. An example of such an inference rule is *modus ponens* which states: if formulae W and $W \Rightarrow X$ are already proven, then X is also proven.

A formal theory for information disclosure can now be constructed in the following way. The basis is a set of primitive descriptors, for example terms, which are used to describe the content of information objects. The primitive descriptors are the atomic formulae in information disclosure. The well-formed formulae are complex expressions involving primitive descriptors and can be used as more sophisticated characterizations of objects than the primitive descriptors. The language of complex descriptors is denoted by $\mathcal{C}$, and the axiomatization of objects is drawn from this language, that is, for each object O we have $\chi(O) \subseteq \mathcal{C}$. In addition, it is assumed that the request q is a complex descriptor, so $q \in \mathcal{C}$. Furthermore, a set of rules of inference is assumed with which we attempt to derive the request q from a given object characterization $\chi(O)$. The first possibility is $\chi(O) \vdash q$, meaning that request q can indeed be proved from the axiom set of object O. From $\chi(O) \vdash q$ we are sure that O is a model for q ($O \models q$), or in less formal terms, object O deals with, or *is about*, request q. Therefore, O is relevant with respect to q and should be returned in response to the request q. (In other words the logic-based approach also adopts the $q = N$ assumption). If q cannot be deduced from $\chi(O)$, then no definitive statement can be said about O being relevant with respect to q; it only means that the relevance of O with respect to q cannot be proved from the axioms associated with O. The above concepts form a so called *Disclosure Structure*, which is defined as follows.

Definition 3.2.1

A Disclosure Structure is a system $\langle \mathcal{O}, \mathcal{C}, \chi \rangle$ *where*

- $\mathcal{O}$ *is a set of information objects*
- $\mathcal{C}$ *is a descriptor language*
- $\chi \subseteq \mathcal{O} \times \mathcal{C}$ *is an indexing relation*

□

After having defined a Disclosure Structure, the question remains as to how the relevance of an object O in response to a request q can be established. This is realized by the so called *Information Disclosure Machine* which is powered by a process of logical inference.

Definition 3.2.2

An Information Disclosure Machine, or Disclosure Machine for short, is a system $\Delta = \langle D, S, P\rangle$ *where*

- D *is a Disclosure Structure*
- S *is a set of rules of strict inference*
- P *is a set of rules of plausible inference*

□

The notion of a derivation or proof using the rules of inference is defined as follows.

Definition 3.2.3

Let $\Delta = \langle D, F, P\rangle$ *be a Disclosure Machine. Let* C *be the descriptor language of the Disclosure Structure* D*. For* $d \subseteq C, x \in C$ *and* $s \in S$, $d \vdash_s x$ *denotes that* x *can be deduced from* d *by applying the strict inference rule* s*. Furthermore,* $d \vdash_\Delta x$ *denotes a sequence of zero or more deduction steps involving rules of strict inference of* Δ*. Analogously, we use* $d \mid\!\sim_p x$ *to denote that* x *is plausibly deduced from* d *via the plausible inference rule* $p, p \in P$*;* $d \mid\!\sim_\Delta x$ *denotes a sequence of one or more deduction steps involving rules of strict inference and rules of plausible inference of* Δ *such that at least one step involves a rule of plausible inference.* □

For reasons of clarity, in the sequel the subscript Δ will often be dropped from $d \vdash_\Delta$ and $d \mid\!\sim_\Delta$.

To begin with, strict inference based on object characterizations will be detailed. Earlier it was stated that from $\chi(O) \vdash q$ it is sure that $O \models q$. This statement is based on the assumption that all object characterizations are valid descriptions and the rules of strict inference preserve relevancy. Under this assumption the Disclosure Machine is said to be *sound*:

$$\chi(O) \vdash q \;\Rightarrow\; O \models q$$

In other words, in a sound Disclosure Machine the objects from whose characterization the request q can be proved are relevant with respect to q.

The converse of *soundness* is *completeness*. Completeness states that all valid propositions are provable, or:

$$O \models q \;\Rightarrow\; \chi(O) \vdash q$$

A complete Disclosure Machine has the advantage that relevance can be established purely by the strict inference process. Unfortunately, a complete Disclosure Machine turns out to be very difficult to realize. We have seen from the previous chapter that object characterizations are incomplete. This restricts the power of the strict inference mechanism, meaning that in general it is not often the case that an object can be proved relevant via the strict inference process.

If a request q cannot be (strictly) deduced from the axioms of O, this does *not* necessarily mean that O is not relevant. It only means that the axioms of O are too weak to establish the validity of q in O. In other words, it is important that the Disclosure Machine does not employ an implicit Closed World Assumption. This assumption states that if a request q is not provable via the rules of strict inference, then the object O is deemed to be irrelevant with respect to q:

$$\neg(\chi(O) \vdash q) \;\Rightarrow\; \neg(O \models q)$$

It will be evident that applying only rules of strict inference results in an imbalance between relevance and provability. To alleviate this imbalance plausible inference is used. The plausible inference mechanism strives to generate high probabilities of relevance for those relevant objects that escaped the strict inference mechanism. If, for a given object O, the probability of relevance is high, then the Disclosure Machine might return the object after all.

The plausible inference mechanism can be founded on the principle of *minimal axiomatic extension* which has its roots in the so called *logical uncertainty principle* [Rij86b]. The principle of minimal axiomatic extension states that the probability of an object being relevant to a request is inversely proportional to the minimal extension of the object description allowing to prove the request. It is important to note that either the characterization of the object must be extended with new axioms, or some axiom(s) of the description have to be *strengthened*; an inverse approach to description strengthening is query weakening [Nie86]. It is easy to see that by strengthening the axioms, the deduction process becomes less certain because it involves suppositions that were not originally a part of the semantics of the object. One way to understand plausible deduction is to view it as a process in which a set of axioms is evolved into the request [BvdW92]. Within this vision, the more evolution necessary, the more dissimilar the request from the original axioms. The amount of evolution can be formalized under the notion of the *evolutionary distance* between characterizations denoted $\delta(x, y)$. As y may evolve from x in a number of ways, meaning that there may be several plausible deductions of y from x, the evolutionary distance between x and y is defined as:

$$\delta(x, y) = \mathtt{min}\left\{\text{plausibility } x \mathrel{|\!\!\sim} y\right\}$$

A simple approach is to define the plausibility of a derivation as the number of plausible inference steps in this derivation. The *similarity* $\sigma(x, y)$ between x and y is then specified as a function that is inversely proportional to $\delta(x, y)$. For example: $\sigma(x, y) = 2^{-\delta(x,y)}$

The probability of relevance can be established as a measure of the similarity between the object's characterization and the request [BvdW91a]:

$$P_{\text{Rel}}(O, q) = \sigma(\chi(O), q)$$

It should be mentioned that derivations of probability of relevance can be supplemented by so called *spatial coherence*. This assumes that the objects are connected in a network.

When relevance cannot be established via the strict inference mechanism, it is "borrowed" from neighbouring objects. (See [BvdW91a] for more details). Spatial coherence is particularly relevant for hypermedia object bases due to their network structure. The notion of spatial coherence will not further be considered in this thesis. The focus will remain plausible inference driven by specific rules. Later in this chapter we will feature a concrete Disclosure Machine, the so called *Refinement Machine*, which functions according to a plausible inference mechanism defined over the language of index expressions. First, however, the expressive power of the logic-based approach will be demonstrated by showing how two common approaches to information disclosure can be considered within the framework of the Disclosure Machine.

3.2.1 Boolean Disclosure Machines

Most people have encountered the automatic information disclosure systems found in libraries. In such systems the information need is often expressed as a set of keywords which can be combined with the logical operators $\wedge, \vee, \neg$. For this reason they are referred to as *Boolean retrieval systems.*

Boolean Disclosure Machines are based on a disclosure structure which is defined in the following way: The set of primitive descriptors comprises a set of keywords. The descriptor language $\mathcal{C}$ consists of logical expressions as mentioned above. Information objects are only characterized by primitive descriptors.

Boolean Disclosure Machines operate according to two rules of strict inference:

Addition

$$A \vdash x \;\Rightarrow\; A \vdash x \vee y$$

Conjunction

$$A \vdash x, A \vdash y \;\Rightarrow\; A \vdash x \wedge y$$

There is no explicit rule of inference to infer negation. It is brought into play via the adoption of a Closed World Assumption. As $\neg(O \models q)$ means that q is not a valid descriptor for O, then it is also assumed that O validates the negation of q ($O \models \neg q$). A consequence of the adoption of the Closed World Assumption is the exclusion of plausible inference ($P = \emptyset$).

That Boolean Disclosure Machines operate under the Closed World Assumption is one of the principle clarifications why these machines offer ineffective disclosure. Of course, the low expressivity of the underlying characterization mechanism is another reason for ineffectiveness. The choice for this characterization mechanism is only motivated by the algorithmic efficiency with which keywords can be extracted from information objects.

3.2.2 Coordination Machines

Another well known retrieval model is the so called coordination model. Just as with the Boolean Disclosure machine, the primitive descriptors is a set of keywords. Both requests and object characterizations consist of a set of keywords. In the coordination machine, there are *no* rules of strict inference. Unlike Boolean information disclosure, the Coordination Machine does not operate under the Closed World Assumption, and thus can offer better disclosure than Disclosure Machines based on the Boolean retrieval system. There is one plausible inference rule, which allows for the addition of a single primitive descriptor to an object characterization. This can be viewed as extending the semantics of an object by the addition of an assumption. The evolutionary distance between characterization x and characterization y, is the number of assumptions that must be added to x in order to validate y:

$$\delta(x,y) = |y - x|$$

The usual method for estimating probability of relevancy is by using the ratio between the intersection and union of the request and characterization. This can be expressed in terms of evolutionary distance as follows:

$$\sigma(x,y) = \frac{|y| - \delta(x,y)}{|x| + \delta(x,y)}$$

3.3 The Refinement Machine: Rules of Strict Inference

In chapter 2, the language of index expressions was introduced as a language for expressing object characterizations. The inference mechanism of the *Refinement Machine* is defined over this language. The Refinement Machine owes its name to the particular way axiomatic extension is realized: axioms are in the form of index expressions which are *strengthened* by a process called *refinement*. The refinement mechanism is defined in terms of an underlying lithoid. In this section, the strict inference mechanism of the Refinement Machine is detailed; its plausible inference mechanism will form the focus of ensuing sections.

3.3.1 Rules of Strict Inference

The set of rules of strict inference consists of three rules called *modus continens*, *modus generans* and *modus substituens*. These rules will be discussed separately.

The inference rule *modus continens* may be looked upon as deduction by way of containment. To illustrate this, we consider the following example. Suppose an information object has the index expression pollution of rivers as an axiom. From this index expression we can see that the object is about pollution, because the information conveyed by pollution is also inherent in pollution of rivers; a similar observation holds for rivers. *Modus continens* is formally defined as follows.

Definition 3.3.1

Let I and J be index expressions in a language $\mathcal{L}(T,C)$ and let $\mathrel{\underline{\subset\kern-0.55em\ast}}$ be the is-subexpression-of relation over $\mathcal{L}(T,C)$. Then, if J is a subexpression of I, J can be deduced from I, or:

$$J \mathrel{\underline{\subset\kern-0.55em\ast}} I \;\Rightarrow\; I \vdash_{\mathrm{MC}} J$$

This rule of inference is called modus continens *and is denoted by* MC. □

Note that there is a analogy here with *modus ponens.*

The intuition behind *modus generans* is deduction by way of generalization. The basis of this rule of inference are generalizations captured in the form of an ISA-relation. For example, given the generalization salmon ISA fish, *modus generans* affirms that any information object that deals with salmon also implicitly deals with fish.

Definition 3.3.2

Let $\mathcal{L}(T,C)$ be a language of index expressions and let $I,J \in \mathcal{L}(T,C)$. Let $\textsc{isa} \subseteq T \times T$. *If $I \,\textsc{isa}\, J$, then J can be inferred from I, or:*

$$I \,\textsc{isa}\, J \;\Rightarrow\; I \vdash_{\mathrm{MG}} J$$

This rule of inference is called modus generans *and is denoted by* MG. □

Care must be exercised when using *modus generans* due to homonyms. For example, the generalization crane ISA bird can only be exploited if the context is avian and not building construction. The ISA-relation is quite common in frame-based knowledge-representation languages ([LG91]). It brings domain knowledge into play within the Refinement Machine. Unfortunately, the ISA-relation cannot typically be derived automatically.

The third rule of inference, called *modus substituens*, drives deduction by way of substitution. Recall from a previous example that pollution is deducible from pollution of rivers by *modus continens.* From that we may conclude for example that any object that is about the effects of POLLUTION OF RIVERS in Australia is also about the effects of POLLUTION in Australia. *Modus substituens* provides for this type of inference.

Definition 3.3.3

Let K and I be index expressions in a language $\mathcal{L}(T,C)$ such that I is a subexpression of K. Furthermore, let K_I^J be the index expression K with I substituted by J. Then,

$$I \vdash J \;\Rightarrow\; K \vdash_{\mathrm{MS}} K_I^J$$

This rule of inference is called modus substituens *and is denoted by* MS. □

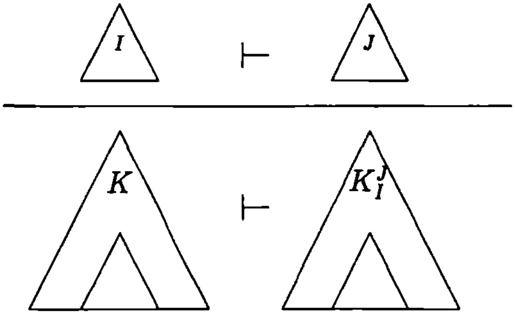

Figure 3.1: *Modus substituens*

The general idea of *modus substituens* is schematically represented in Figure 3.1.

Note that *modus substituens* describes context-free substitution. The disadvantage of this approach is well documented in the work of Chomsky. Within the realms of the Refinement Machine the problem manifests itself in the form of spurious index expressions. For example, from the index expression effects of pollution of rivers in Australia in the above example, effects of rivers can also be derived. However, it is highly unlikely that the object actually deals with this subject. Recently, attempts have been made at restricting conditions under which substitution can take place so as to prevent the occurrence of wild substitutions. Rosing [Ros91], for example, proposes that a term should only be substituted by one of its generalizations:

$$I \text{ ISA } J \Rightarrow K \vdash_{\text{MS}} K_I^J$$

Another approach is to allow substitutions to take place only within the context of a term sequence. A term sequence is a sequence of expressions involving only the null connector. If one considers a term sequence as describing a particular context, then this context can often be implicitly described by any term subsequence which contains the last term. For example, the information conveyed by the expression green ∘ martians is also implicitly contained in little ∘ green ∘ martians (little ∘ green ∘ martians $\vdash_{\text{MC}}$ green ∘ martians). This paves the way for the following deduction step:

invasion of LITTLE ∘ GREEN ∘ MARTIANS $\vdash$
invasion of GREEN ∘ MARTIANS

Note that this restricted *modus substituens* provides a context-sensitive notion of substitution.

The three rules of inference *modus continens*, *modus generans* and *modus substituens* provide the driving mechanism with which index expressions can be derived from others.

3.3.2 The Lithoid Revisited

Note that *modus continens,modus generans* and *modus substituens* all transform one index expression into another. From this observation, it follows that a strict derivation takes the form of a sequence of transformations on an index expression mutating it into another one. An immediate consequence is that the relevance of an object with respect to a given request can be established by proving the request from a *single* index expression in the characterization of the object. This property is stated more formally in the following theorem, called the Hook Theorem. The name signifies that the relevance of an object is proven via a single characterization (which acts as a hook for the disclosure of the object).

Theorem 3.3.1 Hook Theorem
Let $\mathcal{L}(T, C)$ be a language of index expressions and let $q \in \mathcal{L}(T, C)$. If for some object O we have $\chi(O) \vdash q$, then there is an index expression $I \in \chi(O)$ such that $I \vdash q$.

Using this property, the following theorem shows that a lithoid constructed from a core set of index expressions, with the empty index expression excluded, constitutes all theorems derivable from this core set of index expressions by the Refinement Machine restricted to *modus continens.*

Theorem 3.3.2 Let $\mathcal{I}$ be a set of index expressions and $\mathcal{K}_S(\mathcal{I})$ denote the theorems provable from $\mathcal{I}$ using the rules of strict inference S. Then,

$$\mathcal{K}_{\text{MC}}(\mathcal{I}) \;=\; \bigcup_{I \in \mathcal{I}} \wp(I)$$

Proof:
From the Hook Theorem it follows that each theorem originates from a single axiom. Therefore, $\mathcal{K}_{\text{MC}}(\mathcal{I}) = \bigcup_{I \in \mathcal{I}} \mathcal{K}_{\text{MC}}(I)$. Via definition 3.3.1, *modus continens* can be equated with $\subseteq_*$, the *is subexpression of* relation. That is, $\mathcal{K}_{\text{MC}}(I)$ is the set containing all subexpressions of expression I. We know from definition 2.3.4 that this is $\wp(I)$. Therefore, $\mathcal{K}_{\text{MC}}(\mathcal{I}) = \bigcup_{I \in \mathcal{I}} \wp(I)$ □

This theorem has practical significance: All theorems provable via *modus continens* can be produced by generation of a lithoid over the core set of expressions $\mathcal{I}$. It was mentioned in chapter 2 that lithoids can be constructed efficiently. On the other hand, the computational tractability of $\mathcal{K}_{\text{MC,MG,MS}}$ is likely to be expensive. For this reason our practical experiences are confined to a Refinement Machine restricted to *modus continens.* These experiences will be elaborated in chapter 5. First, however, we explore the plausible inference mechanism of the Refinement Machine.

3.4 Context-Free Plausible Inference

It was mentioned earlier that plausible inference is founded on the principle of minimal axiomatic extension. The key question is, if the axiomatization of an object is expressed in terms of index expressions, how can this axiomatization be strengthened. In this section the plausible inference mechanism suggested in [BvdW91b] is shown to be too blunt because of its context free nature. It is therefore rejected as the plausible inference mechanism of the Refinement Machine. In the following sections a more sensitive plausible inference mechanism will be put forward.

3.4.1 Rules of Context-Free Plausible Inference

If the axioms of an object characterization have the form of index expressions, existing axioms can be strengthened by an operation called *refinement*. Informally speaking, refining of index expressions has to do with making them more specific and is achieved by adding a connector-term pair as is demonstrated by the following example. Consider the index expression information. This expression can be refined into need of information which can in turn be refined into people in need of information; these refinements can be better understood by considering the lithoid depicted in Figure 3.2.

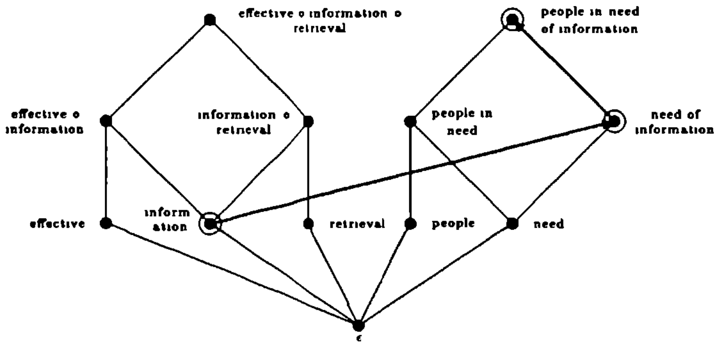

Figure 3.2: refining index expressions

The above refinements are a direct result of the inverse $\subsetneq$ relation over the index expressions and therefore are in turn determined by the underlying lithoid. However, refinement can also be defined via the inverse of the ISA-relation between terms. For example, the expression fish can be refined into salmon. More formally, refinement is defined as follows.

Definition 3.4.1

Let I and J be index expressions in a language $\mathcal{L}(T,C)$. We say that I can be refined *into J, denoted as $I \twoheadrightarrow J$, if and only if one of the following conditions apply:*

1. $I \subseteq^{*} J$ *and* $\forall_K[I \subseteq^{*} K \subseteq^{*} J \Rightarrow K = J \vee K = I]$
2. $I \;\textsc{isa}\; J$ *and* $\forall_K[I \;\textsc{isa}\; K \;\textsc{isa}\; J \Rightarrow K = J \vee K = I]$

□

The refinement operation can be taken as the basis for plausible inference. Via refinement, inference is plausible in the sense that the deductions involve aspects that were not originally a part of the characterization of the object. The first rule based on refinement is *plausible inference through refinement.* The general idea is as follows: Assume, for example, that within the set of information objects characterized by pollution there are objects that deal with the pollution of rivers. On the basis of this we can derive the index expression pollution of rivers from the index expressions pollution. For the moment we will not consider the certainty of the deduction, but merely observe that it is plausible.

Definition 3.4.2

Let I and J be index expressions in a language $\mathcal{L}(T, C)$. Then, if I can be refined into J, J can be plausibly derived from I, or:

$$I \twoheadrightarrow J \Rightarrow I \mathrel{|\!\!\sim}_{\text{PR}} J$$

This rule of inference is called plausible inference through refinement *and is denoted by PR.* □

Note that this rule of plausible inference drives context-free derivation. The same holds for the second rule of plausible inference, called *plausible substitution.* This rule bears a strong resemblance to *modus substituens.*

Definition 3.4.3

Let K and I be index expressions in a language $\mathcal{L}(T, C)$ such that I is a subexpression of K. Furthermore, let K_I^J be the index expression K with I substituted by J. Then,

$$I \twoheadrightarrow J \Rightarrow K \mathrel{|\!\!\sim}_{\text{PS}} K_I^J$$

This rule of inference is called plausible substitution *and is denoted by PS.* □

Together with the rules of strict inference, *plausible inference through refinement* and *plausible substitution* can provide the driving mechanism with which an index expression can be plausibly derived from another index expression. For example, the index expression metals can be derived from pollution of rivers as follows:

$$\begin{array}{rl}
\text{pollution of rivers} & \vdash_{\text{MC}} \\
\text{pollution} & \mathrel{|\!\!\sim}_{\text{PR}} \\
\text{pollution from metals} & \vdash_{\text{MC}} \\
\text{metals} &
\end{array}$$

The following example demonstrates the use of plausible substitution:

$$\begin{array}{rl} \text{effects of POLLUTION OF RIVERS} & \vdash_{\text{MC}} \\ \text{effects of POLLUTION} & \mid\!\sim_{\text{PS}} \\ \text{effects of POLLUTION FROM METALS} & \vdash_{\text{MC}} \\ \text{effects of METALS} & \end{array}$$

In the last example only a single plausible inference step is involved in the (plausible) deduction of effects of metals from effects of pollution of rivers. This means that these expressions have a fairly high degree of similarity. So, if effects of metals is a request, and an information object O is characterized by effects of pollution of rivers, then it is fairly likely that O would be relevant.

3.4.2 Problems with Context-Free Plausible Inference

The plausible inference mechanism constituted by the two rules introduced in the previous section turns out to be inadequate. In this section we will demonstrate this inadequacy with an example. Consider the following three information objects. Object O_1 is about river pollution in Australia, object O_2 is about the effects of pollution in rivers and the third object, O_3, deals with air pollution in Holland. Assume that these objects have the following characterizations:

$$\begin{aligned} \chi(O_1) &= \text{riv} \circ \text{poll in australia} \\ \chi(O_2) &= \text{eff of poll in riv} \\ \chi(O_3) &= \text{air} \circ \text{poll in holland} \end{aligned}$$

Suppose that the request is riv $\circ$ poll. Intuitively, objects O_1 and O_2 would seem to be relevant whereas O_3 would not. Furthermore, imagine that this request is fed into a Disclosure Machine whose *inference mechanism is driven by the rules of inference defined in the* previous sections. Formally, the machine has the rules of strict inference $S = \left\{\text{MC}, \text{MG}, \text{MS}\right\}$ and the rules of plausible inference $P = \left\{\text{PR}, \text{PS}\right\}$. We now unleash this machine and try to derive the request from the characterizations of the three objects. Object O_1 can be shown to be relevant by application of *modus continens*:

$$\text{riv} \circ \text{poll in australia} \vdash_{\text{MC}} \text{riv} \circ \text{poll}$$

As it is not possible to strictly derive the request from the characterization of object O_2, plausible inference is brought to bear:

$$\begin{array}{rl} \text{eff of poll in riv} & \vdash_{\text{MC}} \\ \text{poll} & \mid\!\sim_{\text{PR}} \\ \text{riv} \circ \text{poll} & \end{array}$$

The above derivation involves only a single plausible inference step so it may be concluded that the probability of relevance of object O_2 to the request q is fairly high. Considering the situation, this conforms with our intuition. *However*, the following derivation also only involves one plausible inference step.

$$\begin{array}{rl} \textsf{air} \circ \textsf{poll in holland} & \vdash_{\textsc{mc}} \\ \textsf{poll} & \mid\!\sim_{\textsc{pr}} \\ \textsf{riv} \circ \textsf{poll} & \end{array}$$

This in fact means that on the basis of the above derivations the Disclosure Machine would assess the probability of relevance of the object dealing with air pollution in Holland (O_3) as being the same as that for the object dealing with the effects of pollution in rivers (O_2). Furthermore, the Disclosure Machine would assess this probability as being fairly high. In other words, the object about air pollution in Holland would be returned by the Disclosure Machine in response to the request riv ∘ poll. Clearly, the plausible inference mechanism is too blunt: it is unable to distinguish between object O_2 which deals with river pollution and object O_3 which clearly does not. Figure 3.3 schematically depicts the above derivations in terms of the underlying lithoid. The reason for the bluntness lies partially in the fact that via the *modus continens* rule much contextual information inherent in the initial axiom is thrown away. For example, in the deduction step eff of poll in riv $\vdash_{\textsc{mc}}$ poll information such as the pollution being in rivers and specifically the effects of the pollution, is discarded and thereafter cannot be used later in the derivation. For example, to distinguish between an appropriate plausible inference step (poll $\mid\!\sim_{\textsc{pr}}$ riv ∘ poll) and a inappropriate step (poll $\mid\!\sim_{\textsc{pr}}$ air ∘ poll).

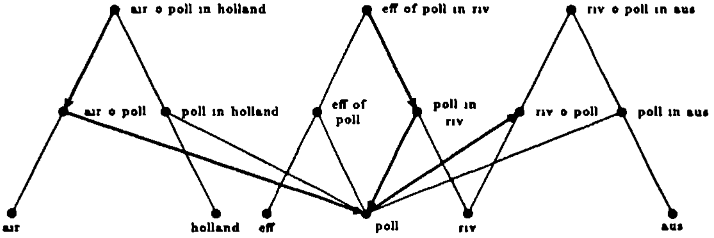

Figure 3.3: Schematic representation of two plausible derivations

3.5 The Refinement Machine: Rules of Plausible Inference

Recall that the intention is to define the inference mechanism for the Refinement Machine. So far, a strict inference mechanism has been proposed. The plausible inference mechanism is still under scrutiny. It will be evident from the previous section that the Refinement Machine should not be based on context free plausible deduction due to its bluntness.

There are a number of methods which could be employed to enhance the sensitivity of context free plausible inference. For example, a weight between the values of 0 and 1 could be associated with each plausible inference step thereby quantifying the certainty associated with the step. The strengths of the individual steps in a plausible proof could then be combined to result in a weight which reflects the certainty of the whole derivation. This is commonly referred to as the *certainty factor approach* and has some notoriety within the artificial intelligence community [vdG90]. The disadvantages of this model are well documented by several authors [vdG90][Pea88]. The basic criticism is that the model is mathematically flawed; the associated computation rules do not always accord with the axioms of probability theory. The flaw manifests itself in the form of contradictions.

This section will eventually present a Refinement Machine whose plausible inference mechanism is driven by probabilistic inference which adheres to axioms of probability theory and therefore avoids the pitfalls of the quasi-probabilistic approaches such as the certainty factor model.

This section begins by coupling plausible inference with probability theory. Consider that index expression i characterizes object O and the request q is also an index expression. Imagine that it is not possible to prove q from characterization i. If we had a joint probability distribution Pr available over the associated index expression language, then the strength of the plausible deduction $i \mathrel{|\!\sim} q$ could be equated with the conditional probability $Pr(q \mid i)$. The following definition builds on this idea by defining the notion of relevance in terms of conditional probabilities.

Definition 3.5.1

Let $\mathcal{L}(T,C)$ be a language of index expressions. Let O be an object, $\chi(O) \subseteq \mathcal{L}(T,C)$ its characterization, and $q \in \mathcal{L}(T,C)$ a request. Let Pr be a joint probability distribution defined on $\mathcal{L}(T,C)$. Then, the probability of relevance of O with respect to request q, denoted $P_{\text{Rel}}(O,q)$ is defined as:

$$P_{\text{Rel}}(O,q) \quad = \quad \mathbf{max}_{i\in\chi(O)}(Pr(q|i))$$

□

The connection between the strict inference mechanism of the Refinement Machine and the probabilistic approach indicated above is as follows: If a request q can be proven from an index expression $i \in \chi(O)$ of an object O, then the probability of q given i is maximal, that is,

$$i \vdash q \Rightarrow Pr(q|i) = 1$$

Only if none of the characterizations of O leads to an increased likelihood in the request q ($\forall_{i\in\chi(O)}[Pr(q|i) = Pr(q)]$) does the Refinement Machine conclude that O is not relevant with respect to q; all other objects are potentially relevant. Stated otherwise, the Refinement Machine does not employ a Closed World Assumption. It is a matter of tuning the Refinement Machine to decide which of the potentially relevant objects should be returned to the searcher. Typically only those objects whose probability of relevance is above some threshold value are returned. More about this in chapter 5.

The crux of the whole matter is the joint probability distribution Pr over the index expression language $\mathcal{L}(T, C)$. For the moment we will not enter into the specifics of Pr but restrict ourselves to the remark that, as such, Pr is expensive to manipulate for the purposes of probablistic inference. This is one of the main reasons why a pure probablistic approach to uncertainty reasoning was shunned by the artificial intelligence community and why the quasi-probabilistic approaches gained considerable popularity; the latter, despite their drawbacks, could in any case be efficiently implemented. Recently, however, belief networks have emerged as a compact and malleable representation of the joint probability distribution Pr [vdG90][Pea88] [Nea90]. Probabilistic inference within the realms of belief network can be reasonably efficiently realized via the propagation of probabilistic information over the network. Probabilistic inference within a belief network comprised of index expressions would seem, therefore, to be a good choice for realizing the plausible inference mechanism of the Refinement Machine. The rest of this section explores this avenue.

3.5.1 Index Expression Belief Networks

Informally[2] speaking, a belief network is a graphical representation of a problem domain depicting the probabilistic variables discerned in the said domain and their interdependencies. These interdependencies are quantified by means of conditional probabilities. Take, for example, the simple belief network depicted in figure 3.4. From the viewpoint of information disclosure, this belief network basically expresses our belief that an object is about the pollution of rivers is dependent on it being about pollution and it being about rivers. This belief is quantified via the conditional probability $Pr(\texttt{POLLUTION OF RIVERS} \mid \texttt{POLLUTION} \wedge \texttt{RIVERS})$. By associating our belief with a conditional probability in this way we have adopted a *subjectivistic* interpretation of the belief network [vdG90]. A subjectivist views the probability of an event as a measure of a person's belief in the occurrence, given the information that person has. Later in this section we will show how the subjectivistic interpretation can be complemented by a quantitative foundation derived from a *frequentist* point of departure. Wong & Yao [WY90] also combine the above two philosophical standpoints in their approach to plausible inference.

It is significant that the index expressions associated with the vertices in figure 3.4 are in upper case. The vertices should not be interpreted as index expressions, *but* as probabilistic variables over the suggested index expression and its complement. For example, POLLUTION OF RIVERS denotes a variable over the set {pollution of rivers, ¬pollution of rivers}. (For the purposes of brevity we will henceforth use P OF R to denote the probabilistic variable POLLUTION OF RIVERS, P to denote POLLUTION and R to denote RIVERS). Therefore, the

[2] The results in the balance of this chapter originates from joint work with L.C. van der Gaag. (See [BvdG93])

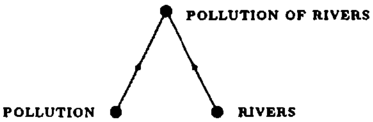

Figure 3.4: Simple Belief Network

conditional probability $Pr(\texttt{P OF R} \mid \texttt{P} \wedge \texttt{R})$ in reality encapsulates eight cases:

$$\begin{array}{l} Pr(\mathsf{p\ of\ r}|\mathsf{p} \wedge \mathsf{r}) \\ Pr(\mathsf{p\ of\ r}|\neg\mathsf{p} \wedge \mathsf{r}) \\ Pr(\mathsf{p\ of\ r}|\mathsf{p} \wedge \neg\mathsf{r}) \\ Pr(\mathsf{p\ of\ r}|\neg\mathsf{p} \wedge \neg\mathsf{r}) \end{array}$$

and the four complementary ones. Note how $\texttt{P} \wedge \texttt{R}$ generates four Boolean formulae. Such formulae are called *configurations* which are defined by a *configuration template* such as $\texttt{P} \wedge \texttt{R}$. Generally speaking, a configuration template C_V over a set V of m probabilistic variables is a conjunction of length m in which each $V_i \in V, 1 \leq i \leq m$ appears. The configuration template C_V defines 2^m configurations, a configuration being a conjunction of length m in which for each $V_i \in V$ either v_i or $\neg v_i$ occurs. The set of configurations implicated by template C_V will be denoted by $\{C_V\}$.

Our small running example demonstrates that conditional probabilities are reflected as directed edges in the belief network. The topology of an index expression belief network is in fact a directed acyclic graph (digraph), $G = (V, A)$ where the vertex set V constitutes a set of probabilistic variables over index expressions as introduced above, and the edge $(V_i, V_j) \in A$ models that variable V_j is directly influenced by variable V_i. The variable V_i is referred to as a *predecessor* of V_j; the set of predecessors of V_j is signified by $\pi(V_j)$. The topology of our small belief network is specified as follows:

$$\begin{aligned} V &= \{\texttt{P}, \texttt{R}, \texttt{P OF R}\} \\ A &= \{(\texttt{P}, \texttt{P OF R}), (\texttt{R}, \texttt{P OF R})\} \end{aligned}$$

In artificial intelligence applications the topology of a belief network is typically established manually, for example, by interviewing an expert in the given problem domain who identifies the variables and their interdependencies. Such an approach is not really acceptable for information disclosure due to the large amounts of information. If the index expression belief network is to be useful it must be generated automatically. To this end we use the lithoid. A searcher exploits a lithoid for information disclosure by moving across it to refine or enlarge the current focus. For a specific search some of the expressions in the lithoid are possibly relevant and some are not. In this sense each of the vertices of the lithoid can be interpreted as a probabilistic variable. As the index expressions in a power index expression are partially ordered, where $\subseteq_{\ast}$ is the underlying ordering relation, it follows that the corresponding probabilistic variables are partially ordered as well. Therefore, the lithoid can be taken to define the (undirected) topology of the graphical part of a belief network. The edges of the resulting undirected graph indicate the partial ordering over the discerned probabilistic variables. These edges are assigned a direction using the inverse $\subseteq_{\ast}$-relation as for the purpose of information disclosure we are interested, for example, in the probability of pollution of rivers given pollution and rivers. From $\subseteq_{\ast}$ being a partial ordering it can be concluded that the graph resulting from the transformation described above is indeed a digraph. Finally, note that in constructing the digraph from the lithoid the empty index expression may be omitted as it is not information bearing. As an example consider the lithoid depicted in figure 2.8. The corresponding digraph constructed from this lithoid is shown in Figure 3.5.

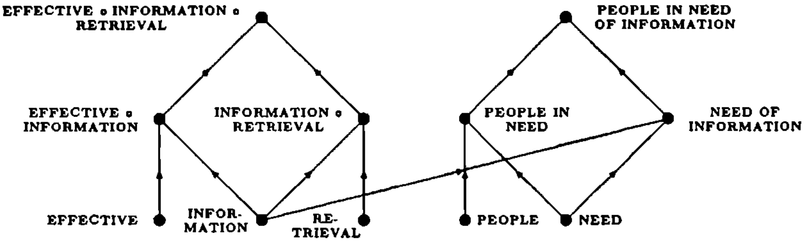

Figure 3.5: Example index expression belief network topology

An index expression belief network does not only have a topological aspect G, but also has a quantitative aspect in the form of a collection Γ of so called *conditional probability assessment functions*. That is a belief network B is a tuple (G, Γ). As mentioned in the introduction, a belief network is a compact and convenient representation of the joint probability distribution Pr. It is from this joint probability distribution that probabilities of interest can be calculated, for example, the probability of a request given an object characterization. There are varying points of departure for representing a joint probability distribution [vdG90, Pea88, Nea90]. We will confine ourselves to a rather naive introduction of this concept based on an algebraic approach. Consider a lithoid comprising the index expressions $\mathcal{I} = \left\{i_1, \ldots, i_n\right\} \subseteq \mathcal{L}(T, C)$, the empty index expression excluded. The set $\mathcal{I}$ constitutes atomic expressions for the specification of the Boolean language of index expressions $\mathcal{B}(\mathcal{I})$ in the usual way. The joint probability distribution, then, is a function $Pr : \mathcal{B}(\mathcal{I}) \rightarrow [0, 1]$ that associates with each Boolean formula a probability. For practical reasons, Pr is often specified in terms of a so called joint distribution function [Pea88]. The joint distribution function associates a probability to each configuration of index expressions such that the sum of these probabilities equals unity. Formally, if $I = \left\{I_1, \ldots, I_n\right\}$ denotes the set of probabilistic variables corresponding to $\mathcal{I}$, then the joint distribution function is a function $P : \{C_I\} \rightarrow [0,1]$ such that $\sum_{X\in\{C_I\}} P(X) = 1$. (Note that $P \subset Pr$). Since every $b \in \mathcal{B}(\mathcal{I})$ can be expressed as a disjunction of configurations and since configurations are mutually exclusive, we can always compute $Pr(b)$ using the joint distribution function and the additive axiom of probability theory ($Pr(b \vee c) = Pr(b) + Pr(c)$ if b and c are mutually exclusive) . Conditional probabilities can be computed using Bayes' well known formula $Pr(b|c) = \frac{Pr(b \wedge c)}{Pr(c)}$. In short, any probability of interest can be derived using the joint distribution function.

The joint distribution function associated with figure 3.5 would contain $2^{11} = 2048$ elements as $|\mathcal{I}| = 11$. As lithoids of practical applications will contain thousands, if not hundreds of thousands, of underlying index expressions, gives a clue as to why Pr, as such, is inefficient to manipulate. The advantage of belief networks, however, is that by supplementing the topology with conditional probability assessment functions, a more malleable representation of Pr arises. Note that only a small number of conditional probability assessment functions need to be specified because the belief network topology represents interdependencies explicitly. More specifically, each vertex $V_i \in V$ is associated with a function γ_{V_i} which describes (conditional) probabilities quantifying the influence of the values of the predecessors $\pi(V_i)$ of the vertex V_i on the values of the V_i itself. More formally, $\gamma_{V_i} : \{v_i, \neg v_i\} \times \{C_{\pi(V_i)}\} \rightarrow [0,1]$, such that $\gamma_{V_i}(\neg v_i|C_{\pi(V_i)}) = 1 - \gamma_{V_i}(v_i|C_{\pi(V_i)})$. We will shortly provide examples of how γ is defined in specific cases. The quantitative part of a belief network $B = (V, \Gamma)$ consists of a set of γ functions, one function per vertex. That is, $\Gamma = \{\gamma_{V_i} | V_i \in V_G\}$. In order to properly link the topology G of a belief network with its quantification Γ it is necessary to assign a probablistic semantics to the topology. This has much to do with defining precisely what independence between variables means. It is beyond the scope of this treatise to deal with this complicated subject. The reader is referred to [vdG90][Nea90] for more details. Before we go into the details of how the γ functions are quantified for index expression belief networks, we quote a theorem taken from [vdG90] which explicitly specifies the joint probability distribution in terms of a product of γ functions.

Theorem 3.5.1 Let $B = (G, \Gamma)$ be a belief network where $V_G = \{V_1, \ldots, V_n\}$, $n \geq 1$. Then,

$$Pr(C_{V_G}) = \prod_{V \in V_G} \gamma_V(V|C_{\pi(V)})$$

Using this theorem, the joint probability distribution of our small example can be written as follows:

$$Pr(\texttt{P} \wedge \texttt{R} \wedge \texttt{P OF R}) = \gamma_{\texttt{P}}(\texttt{P})\gamma_{\texttt{R}}(\texttt{R})\gamma_{\texttt{P OF R}}(\texttt{P OF R}|\texttt{P} \wedge \texttt{R})$$

The task now is to provide specific details of the quantification of an index expression belief network so it can be used for probabilistic inference. Consider first the vertices of an index expression belief network topology (as in figure 3.5) having no predecessors. For such variables prior probabilities on the values such a variable can take, must be specified. From the associated lithoid it can be seen that these variables correspond to unary expressions, or terms. In the context of information disclosure relative to a given set of objects $\mathcal{O}$, it is reasonable to assume that a term that occurs frequently has a higher probability of being in a relevant object than a term that occurs infrequently. The prior probabilities on the values of a term variable may therefore be computed from the occurrence frequency of the term relative to the set of objects $\mathcal{O}$. That is, for a variable T for a term t we compute the value $\gamma_T(t)$ of the assessment function γ_T associated with T using

$$\gamma_T(t) \approx \eta f(t)$$

where η is some normalizing factor. Note that the complementary function value $\gamma_T(\neg t)$ can be computed using the equality $\gamma_T(\neg t) = 1 - \gamma_T(t)$. This approach to estimating the probability of occurrence of a term is common in information disclosure [WY90].

Attention is now focussed on the vertices which correspond with variables that represent binary index expressions. These index expressions are constructed from two terms via the addition of a connector. For example, the binary index expression pollution of rivers is constructed from the terms pollution and rivers, and the connector of. We use this example to define the conditional probability assessment function for the variable P OF R. (See figure 3.4). Eight values must be specified:

$$\begin{aligned}
\gamma_{\mathtt{P\ OF\ R}}(\mathsf{p\ of\ r}|\mathsf{p} \wedge \mathsf{r}) &= w \\
\gamma_{\mathtt{P\ OF\ R}}(\mathsf{p\ of\ r}|\neg\mathsf{p} \wedge \mathsf{r}) &= x \\
\gamma_{\mathtt{P\ OF\ R}}(\mathsf{p\ of\ r}|\mathsf{p} \wedge \neg\mathsf{r}) &= y \\
\gamma_{\mathtt{P\ OF\ R}}(\mathsf{p\ of\ r}|\neg\mathsf{p} \wedge \neg\mathsf{r}) &= z
\end{aligned}$$

and the complementary ones. Now consider the function value

$$\gamma_{\mathtt{P\ OF\ R}}(\mathsf{p\ of\ r}|\mathsf{p} \wedge \mathsf{r}) = w$$

In terms of information disclosure, this function value has the following meaning: Given that we know that an object O is about pollution and we also know it is about rivers, then our belief that O is about pollution of rivers equals w. Later in this section a method for computing such belief estimates based on frequency analyses of connectors in binary expressions will be detailed. First, however, it is important to point out a consequence of harbouring maximal belief in the theorems of the strict inference mechanism. The theorem is euphemized by "No Blind Faith" because we, colloquially speaking, only attribute a non-zero belief in the affirmation of an expression when its direct predecessors are all true.

Theorem 3.5.2 No Blind Faith
Let $B = (G, \Gamma)$ be a belief network. For each triple of variables $I, J, K \in V_G$ such that $(I, K), (J, K) \in A_G$, we have $\gamma_K(k|I \wedge J) = 0$ for $I = \neg i$ or $J = \neg j$.

Proof:
Let Pr be the joint probability distribution defined by the belief network B. From the construction of the digraph G and $(I, K) \in A_G$ we have that $i \subseteq^{*} k$. Therefore,

$k \vdash i$ which implies $Pr(i|k) = 1$. Analogously, we find $Pr(j|k) = 1$. From $Pr(i|k) = 1$ and $Pr(j|k) = 1$, we have $Pr(i \wedge j|k) = 1$. Using marginalization, $Pr(i \wedge j|k) = 1$ implies $Pr(i \wedge \neg j|k) +$ $Pr(\neg i \wedge j|k) +$ $Pr(\neg i \wedge \neg j|k) = 0$. Now observe that from $Pr(\neg i \wedge \neg j|k) = 0$ we find $Pr(k|\neg i \wedge \neg j)Pr(\neg i \wedge \neg j) = 0$. So, at least one of the probabilities $Pr(k|\neg i \wedge \neg j)$ and $Pr(\neg i \wedge \neg j)$ equals zero. We know that $Pr(\neg i \wedge \neg j) > 0$ because $Pr(\neg i \wedge \neg j) = 0$ would imply that there is no information object which is *not* about expression i and *not* about j, which clearly is not the case. Therefore, $Pr(k|\neg i \wedge \neg j) = 0$, and so $\gamma_K(k|\neg i \wedge j) = 0$. Using similar arguments, we find $\gamma_K(k|i \wedge \neg j) = \gamma_K(k|\neg i \wedge \neg j) = 0$. □

A consequence of the above theorem is that the function values x, y and z indicated above are necessarily equal to zero.

The question remains as to how w can be obtained. One method is to analyse the frequencies of occurrence of connectors in binary index expressions. We derived index expressions from the first 500 titles of the Cranfield document collection. A lithoid was constructed from the associated power index expressions. An analysis of the index expressions in the resulting lithoid revealed that approximately fifteen percent of binary expressions involve the of connector. Using this result, the value w can be approximated by 0.15, that is,

$$\gamma_{\mathtt{P\ OF\ R}}(\mathsf{p\ of\ r}|\mathsf{p} \wedge \mathsf{r}) \approx 0.15$$

For binary index expressions involving other connectors, the estimates in the table shown in figure 3.6 can be used. The values in this table are very similar to results gleaned from the same analysis using the CACM collection [Ros91].

Up to this point, we have considered variables that represent terms or binary index expressions and have defined associated conditional probability assessment functions for these variables. The remaining variables will now be considered, that is those variables representing n-ary index expressions, $n \geq 3$. From the construction of a lithoid it will be evident that an n-ary index expression is formed by combining two index expressions of degree $n-1$ that overlap in one term. For example, the ternary index expression people in need of information is constructed from the binary index expressions people in need and need of information on the basis of the term need appearing in both expressions. It is noted that if two index expressions of degree $n-1$ combine into an n-ary index expression, $n \geq 3$, they do so uniquely. So, for two index expressions i and j combining into an index expression k:

$$\gamma_K(k|i \wedge j) = 1$$

No blind faith (theorem 3.5.2) is also applicable here, so

$$\begin{aligned}\gamma_K(k|\neg i \wedge j) &= 0\\ \gamma_K(k|i \wedge \neg j) &= 0\\ \gamma_K(k|\neg i \wedge \neg j) &= 0\end{aligned}$$

The point has now been reached that for all discerned variables, a conditional probability assessment function has been specified. The index expression belief network is now in place.

connector	Probability
o	0.5366
about	0.0052
and	0.0553
are	0.0004
around	0.0017
as	0.0004
at	0.0348
behind	0.0017
between	0.0052
by	0.0061
for	0.0327
from	0.0039
in	0.0632
including	0.0017
into	0.0035
is	0.0004
of	0.1529
on	0.0370
or	0.0026
over	0.0066
through	0.0035
to	0.0170
under	0.0017
using	0.0008
with	0.0248
without	0 0004

Figure 3.6: Connector probabilities

We conclude by observing that the approach presented differs from the one proposed by Croft and Turtle [TC90], in the respect that in our approach the belief network exists purely within the realm of the descriptor language. Recently belief nets have also been investigated in conjunction with term phrases [CTL91] and term hierarchies [FC89].

3.5.2 The Index Expression Belief Network and the Logic of the Refinement Machine

Attention will now focus on describing the relationship between the index expression belief network and the inference mechanism of the Refinement Machine. Basically, information disclosure using the index expression belief network is equivalent to a Refinement Machine which is restricted to *modus continens* and a plausible inference mechanism which "guesses" connectors.

Recall from the previous section that the topology of the graphical part of an index expression belief network was obtained from the lithoid which was constructed from a core set of index expressions. In section 3.3.2 it was argued that another way of looking at the lithoid is that it constitutes all the theorems provable from this core set of index expressions via the strict inference rule *modus continens*. It follows that the Refinement Machine restricted to *modus continens* is implicitly embedded in the associated index expression belief network.

The connection between probabilistic reasoning in the index expression belief network and plausible inference in the Refinement Machine will now be addressed. Consider once again the index expression belief network depicted in figure 3.4. The belief in pollution of rivers given pollution and rivers is represented by the value of $\gamma_{\mathsf{P\ OF\ R}}(\mathsf{p\ of\ r}|\mathsf{p} \wedge \mathsf{r})$. In terms of logic, this is equivalent to the plausible inference step $\mathsf{p}, \mathsf{r} \mathrel{|\!\sim} \mathsf{p\ of\ r}$. Such a plausible inference step can be seen as a step in which the connector of is "guessed". The strength of the guess is represented by the value of the corresponding conditional probability assessment function. By generalizing this example, the plausible inference mechanism of the Refinement Machine is defines as follows.

Definition 3.5.2

Let $i_1, \ldots, i_n$ and j denote index expressions in a lithoid comprising the expressions $\mathcal{I} \subseteq \mathcal{L}(T, C)$, and let Pr be a joint probability distribution over $\mathcal{B}(\mathcal{I})$, then

$$Pr(j|i_1 \wedge \ldots \wedge i_n) > 0 \Rightarrow i_1, \ldots, i_n \mathrel{|\!\sim}_{\text{PI}} k$$

This is termed plausible deduction via probabilistic inference *and is denoted by* PI.

□

Information disclosure via the index expression belief network can be modelled by the restricted Refinement Machine $\Delta = \left\langle D, \left\{\text{MC}\right\}, \left\{\text{PI}\right\}\right\rangle$ assuming the Disclosure Structure D. The (plausible) inference mechanism of this machine is in fact driven by *evidence propagation* through an underlying index expression belief network. The actual workings of the Refinement Machine is the topic of the next section.

3.5.3 Using the Refinement Machine for Information Disclosure

The task of the Refinement Machine is to compute probabilities of relevance of objects with respect to a given request. Those with sufficiently high probabilities of relevance will be returned to the searcher. In definition 3.5.1 the probability of relevance of an object O with respect to a request q is defined in terms of the maximal conditional probability of q given a descriptor $i, i \in \chi(O)$. These conditional probabilities can be computed by the Refinement Machine in the following way: Descriptor i is entered as evidence in the underlying index expression belief network and allowed to propagate over the network resulting in an updated joint probability distribution. The probability of q can now be examined. (It is assumed that a probabalistic variable corresponding to q is in the belief network). This probability corresponds to the belief in q given the context denoted by i. Even though the conditional probability assessment functions describe the joint probability distribution locally for each vertex and its predecessors, calculation of a (revised) probability from the joint probability distribution defined by the assessment functions will generally not be restricted to performing computations which are local in terms of the graphical part of the belief network. In the literature, therefore, several less naive algorithms for computing probabilities of interest from a belief network and for propagating evidence in the network have been proposed. The most well known is the set of algorithms presented by Pearl [Pea88]. The basic idea of these algorithms is that the topology of the graph of a belief network is exploited as a computational architecture. The vertices of the graph are taken as autonomous objects having a local processor capable of performing certain probabilistic computations and a local memory in which the associated conditional probability assessment function is stored; the arcs of the graph are viewed as (bi-directional) communication channels through which the objects can send each other messages. Updating the joint probability distribution and computing local probabilities essentially entails each variable, that is, each vertex, combining its own local information with messages it receives from its neighbours providing it with further information about the joint probability distribution. The Pearl algorithms require that the belief network topology be singly connected, that is, no more than one path exists between any two nodes. Most index expression belief networks do not fulfill this requirement so in this treatise the more generally applicable algorithms devised by Lauritzen & Spiegelhalter will be used [LS88].

Although all algorithms proposed for evidence propagation are based on probability theory, they differ considerably with respect to their approach and their complexity. It should be noted that in general probabilistic inference in belief networks without any restrictions is NP-hard [Coo90]. However, only small restrictions on the topology of the graphical part of the belief network suffice to render the schemes mentioned above polynomial with respect to the number of discerned variables.

The existence of reasonably efficient evidence propagation algorithms pave the way for real life Refinement Machines which function according to a logic-based paradigm for information disclosure. The question of the potential effectiveness of such machines now surfaces. This will be pondered in chapter 5. First, however, we will digress into the world of hypermedia.

Chapter 4

Stratified Hypermedia Structures

And you see that every time I made a further division, up came more boxes on these divisions until I had a huge pyramid of boxes. Finally you see that while I was splitting the cycle up into finer and finer pieces, I was also building a structure.

This structure is formally called a hierarchy and since ancient times has been a basic structure for all Western knowledge.

M. Pirsig - Zen and the Art of Motorcycle Maintenance

4.1 The Information Systems Paradigm

The information retrieval paradigm introduced in chapter 1 (see figure 1.1) dates from the late nineteen fifties when the field of information retrieval was emerging from the realms of library science. Though useful for introducing the basic concepts in information retrieval, it is, however, becoming somewhat outdated. These days, objects need no longer be modelled as amorphous things. Emerging standards such as SGML and ODA take the structure of an information object into account. This structure is not only useful for disclosing the information contained in the object, but must also be maintained in accordance to a structural specification. Therefore, the information retrieval paradigm should explicitly make provision for structural aspects. In recent years the information retrieval paradigm is being challenged by the emergence of hypermedia systems. The primary characteristic of these systems, from an information disclosure perspective, is that the information need of the searcher is satisfied by a process of navigation (sometimes also referred to as browsing). Browsing implies that the information need is not formulated into a request, a process which is acknowledged as being difficult and error prone [Cle91]. Navigation is supported by advanced user interfaces, an aspect which is recognized as something which influences the effectiveness of disclosing the underlying information. We therefore argue that the user interface should also be given consideration within the framework of the information disclosure paradigm.

As the paradigm depicted in figure 1.1 does not cover important aspects relevant to the present day, we advocate considering information disclosure in a broader, more modern *Information System Architecture*. (See [Bub86]). Within this framework an information system is considered to have the following components (see figure 4.1):

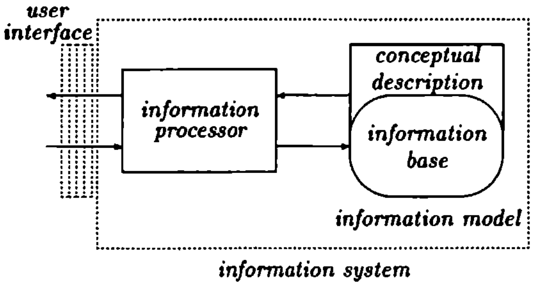

Figure 4.1: The Information System Paradigm

1. The *information model* comprising:
   (a) A *conceptual description* (specification), describing the structure of the stored information, and the rules that govern modifications of the stored information (such as constraints).
   (b) An *information base*, containing the stored information according to the conceptual description. This is usually referred to as an instantiation (population) of the conceptual description.
2. An *information processor*, that processes user requests. The information processor accepts commands from the user via a user interface, interprets them in terms of the conceptual description, and responds in accordance with the information model (both the information base and the conceptual description).

An important difference between hypermedia systems and (conventional) information systems is the concept of *associative link*, that enables the user to navigate through the information base. Furthermore, state-of-the-art hypermedia systems, in contrast to conventional information systems, feature almost no conceptual description of the stored data. The weaknesses of such an approach have been discussed by several authors ([Gar88], [SF89]). There seems to be a growing need to be able to support a conceptual description with regard to both hypermedia and traditional document information systems. The combination of structured documents with hypermedia applications looks promising. This chapter describes a stratified hypermedia architecture founded on this combination. The framework described in figure 4.1 is taken as point of departure.

4.2 Stratified Hypermedia

In[1] the literature there have been a number of papers which focus on formally defining hypermedia at a conceptual level. Several approaches can be recognized; in [Gar88] for example, a model of hypermedia is presented using first-order logic. In [Tom89] hypermedia is modelled in terms of hypergraphs. Recently, two level hypermedia architectures have been emerging [BvdW90b][Luc90][AAC+89][ACG91][GGP89]. Such architectures feature an upper level, the *hyperindex* comprising a hypertext of indexing information which indexes the lower level, the *hyperbase*. The hyperbase contains the actual information. In our approach, both layers will be organized as information models (see figure 4.1). As a result, the layers together constitute a stratified architecture.

An advantage of such architectures is that information can be viewed at different levels of abstraction. For example, the searcher can navigate within the upper level to a description of their information need, and then transfer themselves to the lower level via interlayer navigation in order to arrive at the detailed information. Retrieving information is thus reduced to a process coined *Query By Navigation* ([BvdW90b]). Furthermore, some architectures feature the possibility that a disoriented searcher in the lower level can navigate to the upper level in order to re-orient themselves. We use the term *interlayer navigation* as a generic term for traversal between layers. Relations between layers form the basis for interlayer navigation. The *stratified hypermedia architecture* consists of a number of layers and their interrelations (see also [SDBvdW91]).

A layer offers the possibility to have different views on the same underlying base of fragments. Therefore, views not only allow modularization of the information, but also allow flexibility in the form of multiple views on the same information. Both aspects are generally recognized as desirable in hypermedia systems.

Formally, a layer is introduced as follows:

Definition 4.2.1
A layer *is a structure* $L = (\mathcal{F}, \mathcal{N}, \mathcal{R}, \mathcal{V})$ *where*

- $\mathcal{F}$ *is a set of information fragments. This set is called the* Fragment Base.
- $\mathcal{N}$ *is a set of presentation units (or nodes).* $\mathcal{N}$ *is called the* Node Base.
- $\mathcal{R}$ *is a structure* (E, P)*, where* E *is a set symbols denoting structural elements, and* P *is a set of context free production rules.* $\mathcal{R}$ *is referred to as the* Schema *of the layer.*
- $\mathcal{V}$ *is a set of views, called the* Mask.

□

[1] An earlier version of the content of this section appeared in [BvdW92]

4.2.1 Information Fragments

We start from a set $\mathcal{F}$ of so-called fragments. Fragments are the elementary parts of an information object, which are not decomposed structurally into smaller components. Each fragment has associated data of a particular medium (such as text, video and audio). The criterium for judging whether a fragment is atomic is not necessarily a property of the fragment itself, but rather is dependent on the lowest level of granularity at which the information is to be considered. For example, animation can be considered as a single fragment, or as a sequence of frames.

4.2.2 Nodes

Nodes are units of presentation, and are used to present the structural components to the user. Therefore, nodes are constructed from fragments. Formally, a node is a partially ordered set of fragments and is denoted by the letter N.

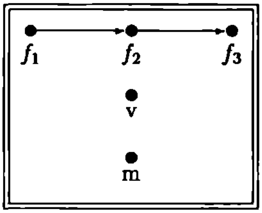

Figure 4.2: A multimedia presentation

As an example, in the node in figure 4.2 the fragments f_1, f_2 and f_3 are displayed on the screen, while at the same time the video v is played, accompanied with the audio fragment m (see figure 4.3 for hypermedia drawing conventions). This node can be represented as the following expression:

$$(f_1; f_2; f_3)\|v\|m$$

A calculus for expressions of this sort has been described in [BvdW89].

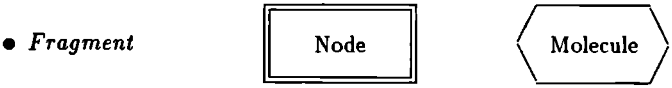

Figure 4.3: Hypermedia drawing conventions

In terms of the conceptual schema, the partial order of above example would be represented as in table 4.1.

fragment	*position*
f_1	$\langle s,1\rangle$
f_2	$\langle s,2\rangle$
f_3	$\langle s,3\rangle$
v	$\langle v,1\rangle$
m	$\langle a,1\rangle$

Table 4.1: Ordering of fragments in node

4.2.3 Rules

Usually information is structured according to some rules. For example, if the information has the form of a book, a book consists of chapters, a chapter consists of sections, etc. Context free rules are a powerful mechanism for such structural specification. A number of models have been defined using context free grammars as their basis ([GT87, TSM91]). Context free rules also have practical significance as they form the core of the Document Type Definitions of SGML ([ISO15]), a document specification language which is gaining widespread acceptance. We adopt the convention of SGML and (basically) allow context free rules only. Rules are expressed in the extended BNF format. This convention is similar to the format adopted by SGML. A rule has a left hand side, which consists of a single symbol and a right hand side, which is a series of one or more symbols, where each symbol may have one of the following occurrence indicators:

- *, the so-called Kleene star, denoting an optional repetition.
- $^+$, the so-called Kleene plus, denoting a repetition.
- ? , denoting an optional occurrence.

Example 4.2.1

The structure of a book comprising chapters comprising sections is described by:

```
book    → chapter*
chapter → section*
```

□

4.2.4 Views

In the stratified hypermedia architecture a view is defined as follows:

Definition 4.2.2

A View *is a structure* $V = (S, \omega, M, \pi, \mathcal{L})$ *where*

- $S \in E$ *is the* start symbol

- *ω is a set of parse trees generated from S using $\mathcal{R}$. ω is referred to as the* actual structure.
- *M is the set of vertices within ω. A vertex is also called a* molecule.
- *$\pi : M \rightarrow \mathcal{N}$ maps each molecule from M to a node.*
- *$\mathcal{L}$ is a set of* associative link *schemata.*

□

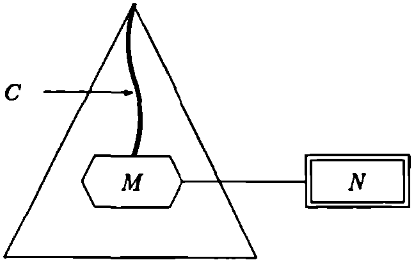

Figure 4.4: The complete context C of molecule M and its presentation by node N

Each vertex in a parse tree corresponds to an instance of a particular structural element, such as a *chapter* or *section*. Such structural elements are termed *molecules*. The *complete context* of a molecule is defined as the path from the root molecule in a parse tree to the molecule in question. By following this path the searcher becomes aware of the contextual framework in which the molecule exists. This might not be the case in an *open context*, where the searcher is only partially aware of this framework. An open, or "dangling" context, corresponds to a downward path in a parse tree, not starting from the root. The term *context* is a generic term for both complete and open contexts.

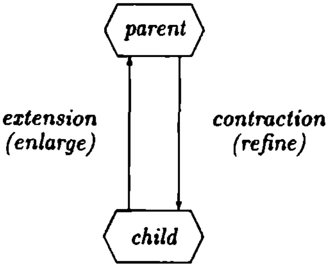

Figure 4.5: Context contraction and extension

The parse tree is useful for disclosure, as it allows the searcher to move through the information on the basis of an underlying structure. For example, moving from a chapter to a section, or from a section to a chapter. This kind of movement is termed *structural navigation*. Structural navigation features the underlying dichotomy that it either extends (enlarges) or contracts (refines) the current context (see figure 4.5).

The rules used to specify the actual structure have a context free format, however, a more liberal application of these rules is permitted than is usual in the theory of context free grammars. In particular, it is possible that a molecule occurs in more than one parse tree. This allows the possibility, for example, that a chapter be shared between different books.

Molecules as such are abstract objects, and need a mechanism to be presented. For this purpose, the function π maps molecules in the actual structure to nodes. In this way the actual structure ω is adorned with content in the form of information fragments resident in nodes. It is the task of the so called *author* to provide an actual structure with a suitable adornment.

A view also contains a set of *associative link* schemata, where a particular link scheme consists of a set of links of the same category. A link originates from a fragment in a node and leads to a fragment in another node:

$$l \in \mathcal{L} \Rightarrow l \subseteq (\mathcal{N} \times \mathcal{F}) \times (\mathcal{N} \times \mathcal{F})$$

Note that the destination node is in the same layer as the source. This restriction contributes to the layer-wise modularization of applications. It is the responsibility of the author to make the link sources, not only visible in the node, but also selectable by the searcher. By selecting a link, the searcher initiates the traversal of the associative link. This is denoted as *associative navigation*.

Ambiguity within Views

When traversing an associative link, several possibilities for system disorientation exist. First, the destination node of the link may be the presentation of more than one molecule. This is termed *presentational ambiguity*:

$$l \in \mathcal{L} \wedge \langle\langle n_1, f_1\rangle, \langle n_2, f_2\rangle\rangle \in l \wedge \pi(M_1) = \pi(M_2) = n_2 \wedge M_1 \neq M_2$$

Presentational ambiguity has to be resolved into a unique context from which the searcher can continue. One possibility is that the choice be made by the information processor. However, with this solution system disorientation then can lead to searcher disorientation. A better solution therefore is that the system provides information about the possible contexts and let the searcher make a choice. Presentational ambiguity can be avoided by employing the constraint that every molecule has a unique presentation. Formally, we require

$$\bigcup_{V \in \mathcal{V}} (\pi_V) \quad \text{is a } \textit{one to one} \text{ function}$$

Another cause of system disorientation is so called *contextual ambiguity*. Consider the situation depicted in figure 4.6. The searcher has traversed an associative link and has arrived

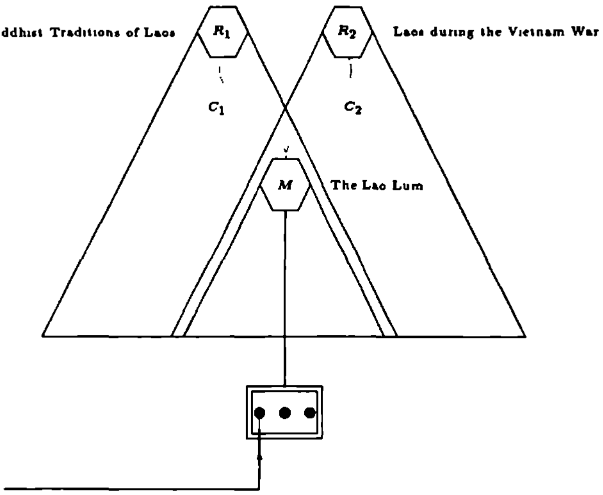

Figure 4.6: Example of contextual ambiguity

at the presentation associated with molecule M. This molecule models a structural element, for example a chapter, dealing with the *Lao Lum*, the valley people of the southeast asian country, Laos. Laos is not famous for many things, but there are two contexts within which it enjoys some notoriety: Laos has a rich Buddhist tradition and appalling things happened there during the Vietnam war. The ambiguity arises because there is more than one contextual framework with which the searcher may continue due to molecule M being shared. In such cases, the system can ask the searcher to chose one of the alternative contexts. In this example, the searcher could continue with the Lao Lum within the *Buddhist* context or the *Vietnam War* context. When a choice is made, the contextual ambiguity is resolved in a single step. Later we will demonstrate how step-wise context resolution can be used to support the searcher in the clarification of their information need.

4.2.5 The expressive power of Layers

There are a number of currenty available specification languages for documents, for example, SGML [ISO15], TEX [Knu84] and ODA [CGR87]. The underlying conceptual model of these description languages is implicit in their definition, although it is recognized that the conceptual model is important. (See, for example, [Sch89] for a conceptual description of SGML).

The concept of a layer is powerful enough to express the important aspects of these languages. For example, SGML-based documents are easily mapped into a layer in the following way: Each SGML document can be considered as a separate view whose actual structure conforms to the grammar specified in the Document Type Definition (DTD) of the document. Cross references between SGML documents are modelled as associative links between views.

A feature of ODA documents is that they can be viewed both from a logical or a layout perspective. In our architecture, this is modelled by two views based on the same underlying set of fragments (*content portions* in ODA terminology). The actual structures of each view correspond to the *specific logical structure* and *specific layout structure* respectively. The start symbol of each view identifies a set of rules which define a *document class.* Figure 4.7 illustrates another example of multiples views on the same underlying set of fragments in the context of document maintenance. The document readers view has no structure. It presents the document as a whole, so that the document can basically be read sequentially. The document maintenance view, on the other hand, takes the full structure of the document into account. This is useful when the component parts of the document are to be manipulated.

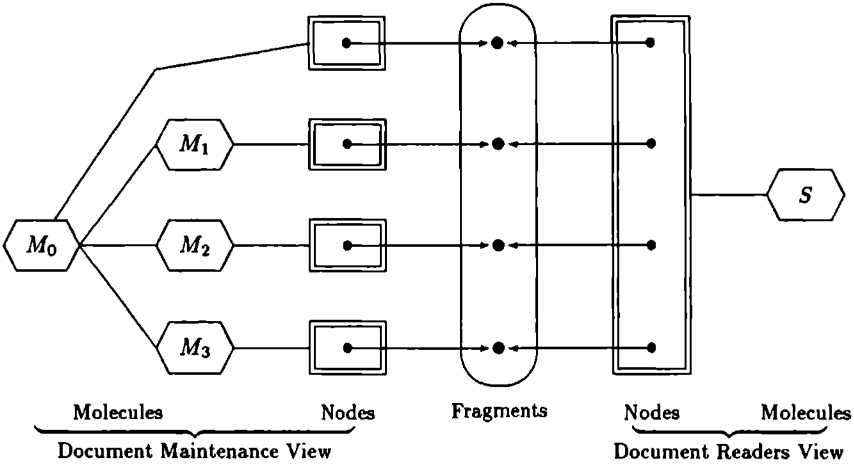

Figure 4.7: Two document views over the same fragments

The notion of layer is also sufficiently powerful to model state-of-the-art hypermedia. We refer to such hypermedia as *flat hypermedia* as they are constructed by chopping the documents into fragments and linking these fragments together to form a network structure suitable for navigation. Such hypermedia are modelled as a layer whose schema $\mathcal{R} = (\{S\}, \emptyset)$. That is, the layer has no structure (the productions, $P = \emptyset$). As a consequence, each parse tree consists of a single molecule which corresponds to the start symbol S. (See the reader's view depicted in figure 4.7). The presentation of each molecule comprises a node containing one or more fragments. The network structure is realized by imposing an associative link scheme over these presentations.

Associative link schemes offer flexibility as they impose no restrictions on how nodes can be linked together. Structural aspects in flat hypermedia are only simulated by a special link scheme, for example, the hierarchical link. This implies that there is no possibility to constrain the actual structure which is the principal reason why flat hypermedia can readily degenerate into an interweaved mess.

4.2.6 An example Hyperbase Layer

In this section we present a concrete example of a layer in the form of a hyperbase, the lower level of a two level hypermedia. We start with the following set $\mathcal{F}$ of information fragments:

f_1 : The effects of pollution on fish can be related to various aspects.

f_2 : The increased industrial capacity of most countries has led to higher concentrations of heavy metals in rivers.

f_3 : These metals have caused the destruction of the ecosystems on which the fish depend as well as killing the fish directly.

f_4 : The effects of the heavy metals remain predominant for years because they sink to the river bottom and are only very slowly flushed out by river currents.

f_5 : Many lakes in Scandinavia have been rendered lifeless by acid rain.

f_6 : This is caused principally by the coal burners in the Ruhr and industrial centres in East Germany and Poland.

f_7 : Because of the economic importance of salmon we consider the effects of river pollution on their migration.

f_8 : There is a higher concentration of heavy metals in rivers due to the increased industrial capacity of most countries.

From these fragments, the following nodes are composed:

N_1 (f_1) The effects of pollution on fish can be related to various aspects.

N_2 ($f_2 + f_3$) The increased industrial capacity of most countries has led to higher concentrations of heavy metals in rivers. These metals have caused the destruction of the ecosystems on which the fish depend as well as killing the fish directly.

N_3 (f_4) The effects of the heavy metals remain predominant for years because they sink to the river bottom and are only very slowly flushed out by river currents.

N_4 ($f_5 + f_6$) Many lakes in Scandinavia have been rendered lifeless by acid rain. This is caused principally by the coal burners in the Ruhr and industrial centres in East Germany and Poland.

N_5 $(f_7 + f_8)$ Because of the economic importance of salmon we consider the effects of river pollution on their migration. There is a higher concentration of heavy metals in rivers due to the increased industrial capacity of most countries.

A structure is imposed on this information according to the following grammar:

1. Non-terminals: $E = \left\{\mathsf{S}, \mathsf{M}\right\}$.

2. Rules: $P = \left\{\, \mathsf{S} \rightarrow \mathsf{M}^+ \,\right\}$

Three views are identified: $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$. Each view has S as start symbol, and consists of a single parse tree in the associated actual structure. This leads to (see figure 4.8):

$\mathcal{V}_1$ the *Pollution and Fish* view. This view has S_1 as start molecule. The start molecule leads to the sequence $M_1, \ldots, M_5$, presented respectively as $N_1, \ldots, N_5$.

$\mathcal{V}_2$ the *River Pollution and Salmon Migration* view, has start molecule S_2, and an underlying sequence M_2, M_3, M_5.

$\mathcal{V}_3$ the *Lake Pollution* view, consisting of start molecule S_3 refined as the sequence M_4, M_3.

Note that this example suffers from contextual ambiguity. However, there is no presentational ambiguity.

The final aspect to be considered are the associative schemata. These are assumed empty for all three views. The resulting structure is depicted in figure 4.8. Note that the presentations of the molecules S_1, S_2 and S_3 have been omitted in this figure for reasons of clarity.

In these views contain some interesting phenomena: Firstly, there is redundancy within the *River Pollution and Salmon Migration* view. This view contains *twice* a sentence about how increased industrialization has led to higher heavy metal concentrations in rivers. (See nodes 2 and 5). Secondly, the *Lake Pollution* view contains irrelevant aspects, namely node 3 is about heavy metals in rivers and has thus nothing to do with the pollution of lakes. This leads to the question of the *quality* of a view, an issue which will covered later in this chapter. First, however, we move up an abstraction level from the hyperbase layer into the hyperindex layer.

4.2.7 The Hyperindex

A hyperindex is a layer of indexing information within the stratified architecture. In such a layer, the fragment base consists of a set of descriptors. The hyperindex typically consists of a single view. Usually, this view is organized as a flat hypermedia. We present three examples of hyperindices.

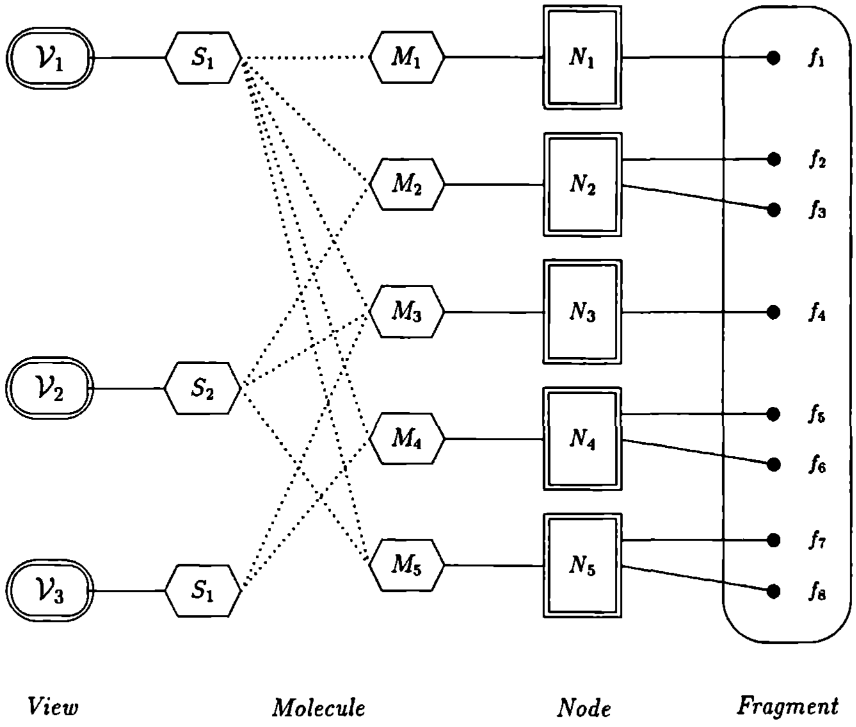

Figure 4.8: The hyperbase of the Pollution Example

4.2.7.1 A Vocabulary as Hyperindex

In this section an example of a hyperindex is described which is based on a simple set of index terms. The underlying set of fragments (the vocabulary) is:

$\mathcal{F}$= { concentration, destruction, ecodeme, economy, ecosystem, fish, heavy-metal, industry, lake, lifelessness, migration, pollution, river, riverbottom, salmon, Scandinavia, trek, waterbody }

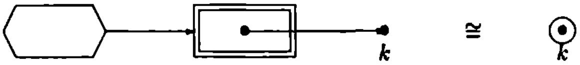

Figure 4.9: Embedding of keywords in the hyperindex

We choose not to impose structure and will therefore model the hyperindex as a flat hypermedia having the grammar $G = (\{\mathsf{E}\}, \varnothing)$. One view is introduced. For each index term a molecule is introduced, and adorned as shown in figure 4.9. This figure also gives a shorthand pictorial representation for such a structure. This enables us to depict the hyperindex as in figure 4.10.

Two schemata for associative links are introduced based on the following relationships between index terms:

isa The isa-relation expresses the categorical class of index terms. In our example, this relation only consists of:

salmon ISA fish
river ISA waterbody
lake ISA waterbody

corr The corr-relation is a symmetric relation, expressing that an index term corresponds to another index term. In our example we have:

ecodeme CORR ecosystem
trek CORR migration

The corresponding associative links are represented in figure 4.10.

4.2.7.2 A Thesaurus as Hyperindex

Iconclass is a tool for the characterization and disclosure of subjects, themes, and motifs pertaining to art of the Western world. The term *Iconclass* is derived from **Icon**ographic **class**ification. It has been developed over the last forty years by de Waal and Couprie [vdW85]. (See [Pou92] for an introductory article on Iconclass). Iconclass divides the world of art into nine main divisions which are depicted in figure 4.11. Each of these

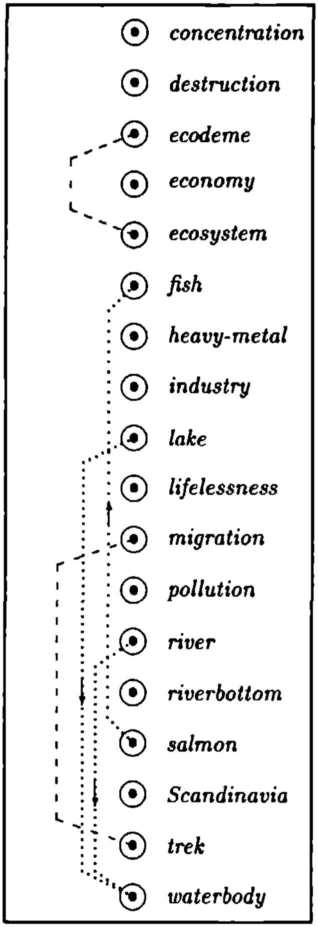

Figure 4.10: The vocabulary hyperindex

1	The Supernatural, God and Religion
2	Nature
3	Human being, man in general
4	Society, civilization, culture
5	Abstract ideas or conceptions
6	History
7	The Bible
8	Myths, legends and tales (not from classical antiquity)
9	Myths, legends and tales from classical antiquity

Figure 4.11: Main classifications of Iconclass

main classifications is further subdivided into so called primary subclassifications which are in turn divided into secondary subclassifications and so forth. For example, the primary subdivisions of The Supernatural, God and Religion are depicted in figure 4.12. From an information science perspective, Iconclass is a *faceted hierarchical thesaurus* [SD86]. Such thesauri are typified by having a small set of facets at the root of the hierarchical organization whereby each facet possesses its own subhierarchy of facets.

1.1	Christianity
1.2	Non-Christian religions
1.3	Magic and Occultism
1.4	Astrology

Figure 4.12: Primary subclassifications of The Supernatural, God and Religion

Iconclass can be modelled as a hyperindex layer in the following way. The fragment base $\mathcal{F}$ comprises terms like those depicted in the previous figures. Two possibilities exist to model the hierarchical structure of the thesaurus. The first is a structural approach: The schema $\mathcal{R}$ specifies a context free grammar which generates hierarchical structures. For example, $\mathcal{R} = \left\langle \{\mathtt{S}\}, \{\, \mathtt{S} \rightarrow \mathtt{S}^{+} \,\} \right\rangle$. A second possibility is to model the thesaurus as a flat hypermedia, in which the hierarchy is simulated with a special link scheme. For the purposes of this example, we adopt the structural approach. Only one view will be introduced. This view necessarily has S as start symbol. As the thesaurus consists of a singly hierarchy, the actual structure will contain only one parse tree. The presentation of the molecules depicts the hierarchical relationship between the classifications. Refinement and enlargement of a classification are realized by structural navigation. (In the presentation depicted in figure 4.13 context refinement is denoted by ($\bigtriangledown$) and context enlargement by ($\bigtriangleup$)). Iconclass does not feature cross references between classifications, meaning that there is no possibility for associative navigation ($\mathcal{L} = \emptyset$).

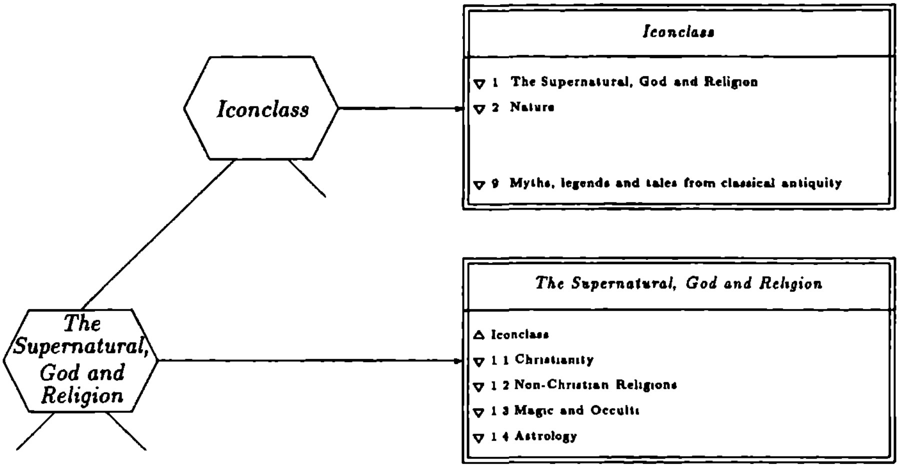
Iconclass
Iconclass
▽ 1 The Supernatural, God and Religion
▽ 2 Nature
▽ 9 Myths, legends and tales from classical antiquity
The Supernatural, God and Religion
The Supernatural, God and Religion
△ Iconclass
▽ 1 1 Christianity
▽ 1 2 Non-Christian Religions
▽ 1 3 Magic and Occult
▽ 1 4 Astrology

Figure 4.13: Actual Structure and presentation of Iconclass

4.2.7.3 A Lithoid as Hyperindex

In section 2.3 of chapter 2 it was described how a lithoid can be constructed by forming a union of power index expressions. Figure 4.14 depicts an example of such a lithoid. It is formed by the union of ℘(effective ∘ (information ∘ (retrieval)) and ℘(people in (need of (information)).

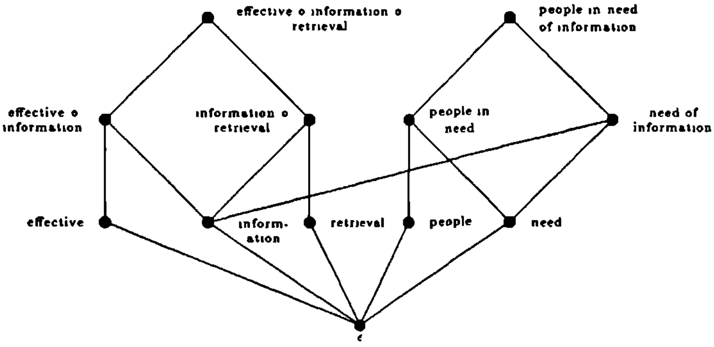

Figure 4.14: Example lithoid

The lithoid forms a useful structure for supporting Query by Navigation. If we consider every vertex as a potential focus of the searcher, then the surrounding descriptors of the focus are enlargements or refinements of the index expression denoting the current focus. The searcher can move across the lihoid by refining (making more specific) or enlarging (making more general) the current focus. Searching is thus reduced to an organized form of browsing. Figures 4.15 and 4.16 depict a refinement operation from both the user interface perspective and the underlying lithoid. In these figures, the enlargements of the focus are denoted by ($\bigtriangledown$) and refinements by ($\triangle$).

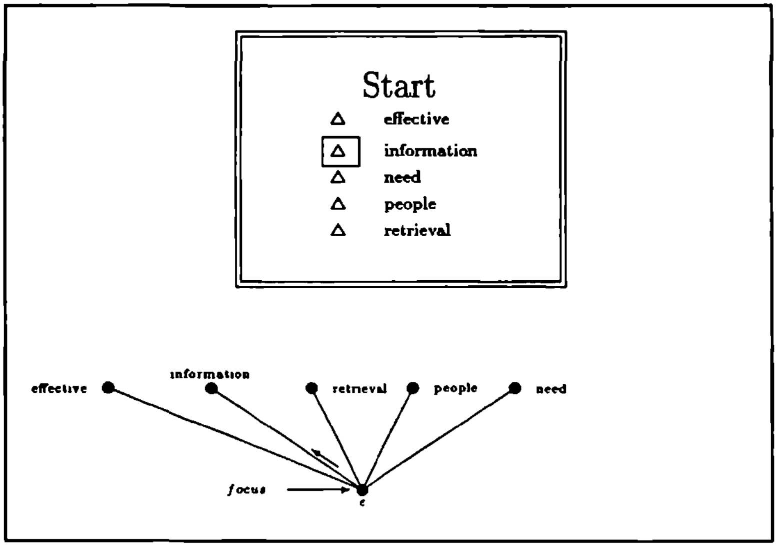

Figure 4.15: Before refinement

The lithoid-based hyperindex can be modelled as a layer in the following way. The fragment base $\mathcal{F}$ consists of the index expressions like those depicted in the previous figures. Just as was the case with the thesaurus, two possibilities exist to model the lattice structure of the index expressions. In both cases, a single view is sufficient for this.

The basic idea of the structural approach is to model each component power index expression of the lithoid by a separate "parse tree". Therefore, the schema specifies a context free grammar which generates hierarchical structures. For example, $\mathcal{R} = \left\langle \left\{ \mathsf{S} \right\}, \left\{ \mathsf{S} \rightarrow \mathsf{S}^{+} \right\} \right\rangle$. The lattice structure of the power index expression is achieved by molecule sharing. (See figure 4.17). The sharing reflects that an index expression may be a subexpression of several expressions. The resulting actual structure is an interweaving of component parse trees. To envisage such an actual structure, consider figure 4.14 in which every vertex corresponds to a molecule.

A second possibility is to model the lithoid as a flat hypermedia. In this case, each edge of figure 4.14 corresponds to an associative link. Thesaural relations, for example the *isa* relation, can be modelled via special link schemes for this purpose. Bosman *et al* have implemented a lithoid hyperindex of art history information along these lines. (See [BBB91]).

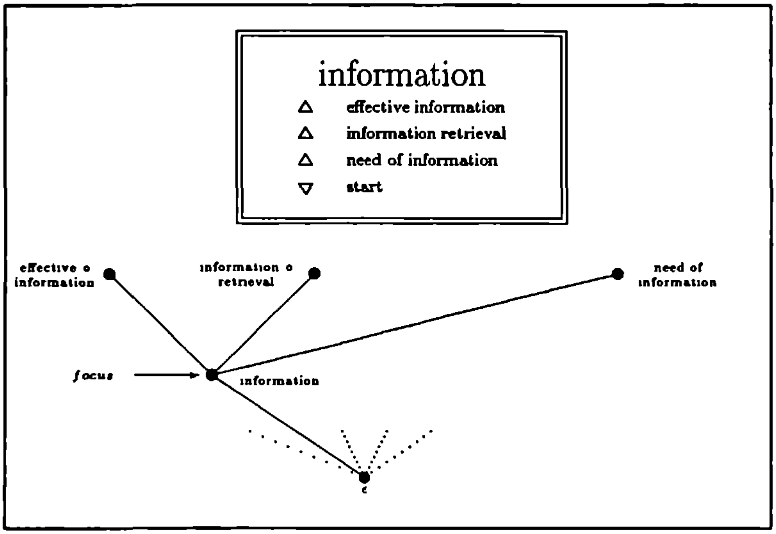

Figure 4.16: After refinement

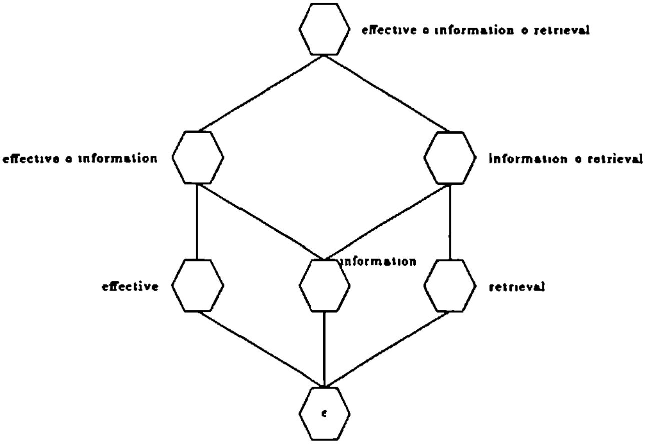

Figure 4.17: A power index expression as parse tree

The disclosure effectiveness of such a hyperindex is dealt with in Chapter 5. The reader is referred to [Bru90] for more details regarding lithoid-based hyperindices. A description of a hyperindex based on a lattice of terms can be found in [GGP89].

4.2.8 De-ambiguation of the Information Need by Step-wise Context Resolution

Previously, it was shown how contextual ambiguity can be resolved in a single step. (See figure 4.6). When a searcher enters a lithoid based hyperindex contextual ambiguity immediately arises. A one step approach to the resolution of this ambiguity is not appropriate when the searcher in the hyperindex does not have a clear idea of their information need. In a sense, the contextual ambiguity parallels the ambiguity of the searcher's information need. In this section we will detail how the information need can be de-ambiguated via a process of *stepwise context resolution.*

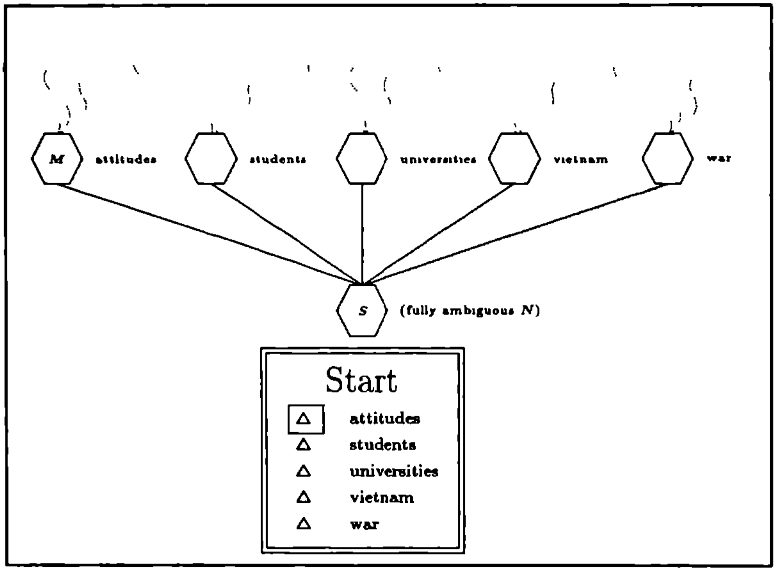

Figure 4.18: Entering the hyperindex

Consider figure 4.18. This depicts a searcher entering a lithoid based hyperindex. Note that the start molecule is a part of every complete context and can be seen as corresponding to a fully ambiguous information need. In this example, the searcher decides that (s)he wants to develop their information need further via selection of the description **attitudes**. The result is the situation depicted in figure 4.19. The searcher has now completed one step in the process of de-ambiguating their information need. The focus is now the molecule M which corresponds to the unary expression **attitudes**. This molecule resides in more than one complete context so there is still contextual ambiguity. However, a clear step has been made towards the resolution of the context because the number of contexts in which molecule **attitudes** is a part of, is a subset of the complete contexts which were applicable at the start. More formally, if $contexts(\Sigma)$ denotes the complete contexts in which molecular sequence Σ are a part of, then $contexts(\langle M, S\rangle) \subset contexts(\langle S\rangle)$. Note that $\langle S\rangle$ and $\langle M, S\rangle$ denote open contexts as mentioned earlier.

By chosing for **attitudes**, a number of contexts have been consciously *rejected*. Bruza & Van der Weide [BvdW92] have developed a calculus based on contextual characterizations which can be used on the rejected contexts to predict future searcher behaviour. These predictions are used to rank the refinement possibilities of the current focus in order of likelihood of relevance [Gro91].

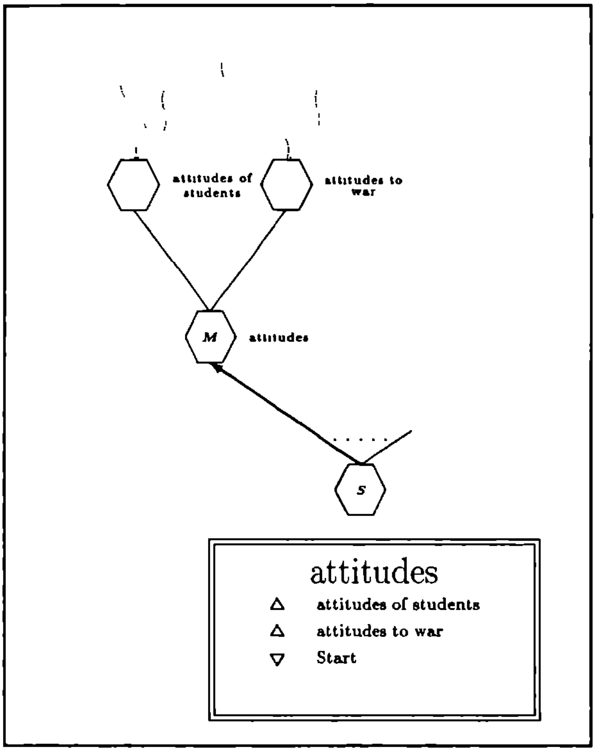

Figure 4.19: After one step in the de-ambiguation of the information need

Via the step wise contextual resolution the searcher can gradually home in on suitable description(s) of their information need. Note that it is not necessary to completely resolve contextual ambiguity, for example, the searcher could decide that the expression **attitudes to war in vietnam** fits their information need and stop searching even though the associated context may still be ambiguous.

4.3 The User Interface

In the previous section the stratified hypermedia architecture has been introduced. In this section the behaviour of such a system towards the searcher will be described in terms of operations that are to be performed by the information processor. (See figure 4.1).

The basic concept in the interaction with a searcher is a context. A context in the hyperindex represents (part of) the information need of the searcher. By operating on this context, the searcher will reach a point where no improvements are possible. For example, suppose the searcher is interested in documents that deal with the role of astrology in myths from the classical antiquity (see figure 4.13), and is searching in the hyperindex layer. First the searcher locates the context in the hyperindex that as good as possible corresponds to the first descriptor, being the classification *myths, legends and talks from classical antiquity.* From this point, no improvement is possible that results in a context that better describes the information need. The searcher will then put a hold on this context, and starts a search for the missing part of the information need. This search will lead the searcher to the classification *astrology.* As a result, the searcher has managed to describe the information need by two contexts. Via interlayer navigation, the searcher can then retrieve the relevant information objects.

Generally, a searcher will have activated a number of contexts, one of which is selected as focus for further processing. This set of activated contexts can be seen as a set of guide-cards in the current layer. For this reason the current set of activated contexts will be referred to as a *guide.* This guide can be seen as a reflection (thus far) of the information need of the searcher.

Only the context under focus (if any!) may be subject to further processing. The focus is displayed to the searcher by its presentation. The searcher now has the following possibilities:

1. *creation or change of focus*
   The searcher may select a new focus, either by activating a new context as an extension to the guide, or by choosing another context in the guide for further elaboration.

2. *interlayer navigation*
   If the searcher finds the guide a precise enough description of their information need, the searcher can use this set as a key for entering another layer. There are two special cases. Going from the hyperindex into the hyperbase is denoted as *beam down.* Going in the opposite direction is termed *beam up.*

3. *associative navigation*
   Typical for hypermedia systems is that they offer the opportunity to follow up on some issue of the current presentation, by making use of the link system of the current view. The actual links within a presentation then have to be visualized by the author, and selectable, usually via a pointing device. Traversing an associative link results in a shift of context.

4. *structural navigation*
   The searcher may enlarge (extend) or refine (contract) the context under focus by structural navigation. In this case, the searcher traverses over the structural geography of the layer. When operating on a complete context, structural navigation cannot result in contextual ambiguity, even if the current molecule is shared as substructure by more than one molecule.

5. *context resolution*
   If the current context is ambiguous, the searcher can make a step towards resolving this ambiguity. The system will present the possible parent molecules (in some appropriate order), and let the searcher choose between one of the alternatives. (See, for example, figure 4.18).

6. *context dissolution*
   The converse of context resolution is context dissolution. The searcher can make a context more general by making it (more) ambiguous. The effect of this operator is that the head molecule of the current context is pruned.

7. *query by similarity*
   When reading the current presentation, the searcher might find descriptors in the text that better characterize the information need than the current guide. The searcher may then highlight this information (usually by an appropriate mouse actions) for building a corresponding guide in the hyperindex, and making the transfer to this new guide.

4.4 The Characterization Calculus

In this section attention is directed to the characterization of the information objects (fragments, nodes, molecules) within a view. The characterizations are not only fundamental for the disclosure of the objects, but also offers the possibility to compare objects on the basis of their assumed information overlap. The latter aspect will be used later in this chapter to formally underpin criteria with which the quality of views can be judged.

4.4.1 Characterizing fragments

For the purposes of modelling characterizations, the function χ is used. This function maps the information objects (fragments, nodes, molecules) onto the fragment base of another layer. For example, the information objects of the hyperbase are characterized by the fragments of the hyperindex (the descriptors). The characterization of the fragments of a layer is the basis of the characterization ? calculus. This characterization is usually obtained by some efficient indexing algorithm.

In the context of our ongoing hypermedia example we characterize by keywords and assume the characterization of the fragments to be:

$$\chi(f_1) = \left\{\text{pollution, fish}\right\}$$

$$\begin{aligned}
\chi(f_2) &= \{\text{concentration, heavy-metal, river, industry}\}\\
\chi(f_3) &= \{\text{destruction, ecosystem, fish, heavy-metal}\}\\
\chi(f_4) &= \{\text{heavy-metal, riverbottom}\}\\
\chi(f_5) &= \{\text{lifelessness, lake, Scandinavia}\}\\
\chi(f_6) &= \{\text{pollution}\}\\
\chi(f_7) &= \{\text{economy, salmon, pollution, river, migration}\}\\
\chi(f_8) &= \{\text{concentration, heavy-metal, river, industry}\}
\end{aligned}$$

4.4.2 Characterizing nodes

The characterization of a node is derived from its underlying fragments as follows:

Definition 4.4.1

Given a node N, then the characterization *of N is defined as*

$$\chi(N) = \bigcup_{f \in N} \chi(f)$$

□

where $\cup$ is a suitable associative binary operator on characterizations. Note that the structure of the presentation, as reflected in the partial order in which the information fragments are presented, is not taken into account.

In our ongoing example the operator $\cup$ unites sets of index terms. For example:

$$\begin{aligned}
\chi(N_3) &= \chi(f_4)\\
&= \{\text{heavy-metal, riverbottom}\}\\
\chi(N_4) &= \chi(f_5) \cup \chi(f_6)\\
&= \{\text{lifelessness, lake, Scandinavia, pollution}\}
\end{aligned}$$

4.4.3 Characterizing molecules

The characterization of molecules consists of two components:

1. A characterization which can be derived from its presentation. This is denoted as the *weak* characterization χ_w.

2. A characterization which can be derived from its structural cohesion, the *strong* characterization χ_s.

The following simplification is assumed (see also [BvdW90b] and [BvdW90a]): Suppose M is a molecule, with submolecules $M_1, \ldots, M_n$, then:

$$\begin{aligned}\chi(M) &= \chi_w(M) \cup \chi_s(M)\\ \chi_w(M) &= \chi(node(M))\\ \chi_s(M) &= \bigcup_{1 \leq i \leq n} \chi(M_i)\end{aligned}$$

With respect to the ongoing example:

$$\begin{aligned}\chi(M_4) &= \chi(N_4)\\ &= \{\text{lifelessness, lake, Scandinavia, pollution}\}\\ \chi(M_3) &= \chi(N_3)\\ &= \{\text{heavy-metal, riverbottom}\}\\ \chi(S_3) &= \chi(node(S_3)) \cup \chi(M_4) \cup \chi(M_3)\\ &= \{\text{lifelessness, lake, Scandinavia, pollution,}\\ &\quad \text{heavy-metal, riverbottom}\}\end{aligned}$$

4.4.4 Comparing information objects

The characterization of information objects gives the opportunity to a quantitative comparison on the basis of their information overlap. For example, if two nodes share a large number of descriptors, then they are assumed to have a strong overlap. If, however, the intersection of their respective characterizations is empty, then the assumption is that they have no overlap. On the basis of this intuition overlap is modelled on a scale of zero to one, where a value of one means that the two objects have exactly the same characterization, and therefore are considered to contain the same information, which not necessarily implies that the two objects are identical. Conversely, a value of zero means that there is no correlation. Building on this intuition, the following definition formally establishes the notion of information overlap between objects within a layer.

Definition 4.4.2

Let $\mathcal{O} = \mathcal{F} \cup \mathcal{N} \cup \bigcup_{\mathcal{V} \in \mathcal{V}} M_{\mathcal{V}}$*, then*

$$\text{Overlap} : \mathcal{O} \times \mathcal{O} \rightarrow [0, 1]$$

is a measure of the information overlap between objects. □

The overlap between objects A and B is defined as the similarity of their descriptions $\chi(B)$ and $\chi(A)$:

Definition 4.4.3

$$\text{Overlap}(A, B) = \text{Sim}(\chi(A), \chi(B))$$

□

Example 4.4.1

If keywords are used as characterization mechanism, then the following gives a good reflection of overlap:

$$\textsc{Sim}(x,y) = \begin{cases} \frac{|x \cap y|}{|x \cup y|} & \textit{if } x \cup y \neq \emptyset \\ 1 & \textit{otherwise} \end{cases}$$

The overlap of fragments f_1 and f_7 is $\frac{1}{6} = 0.167$. *The overlap of nodes N_2 and N_5 is* $\frac{5}{12} = 0.417$. □

4.5 The Quality of Views

The stratified hypermedia architecture offers the possibility to create several views on the same underlying set of information fragments. Just like functions and procedures in programming languages, views offer the possibility to modularize the information within a layer. A good modularization will hopefully furnish effective information disclosure. Current hypermedia systems offer limited support for views. There are two aspects in this regard; view creation and view management. Consider the author who tries to create a view dealing with the life and music of Chopin, within a layer whose information consists of composers of the romantic period. The author must first attempt to locate the relevant information about Chopin, and then bring structure into it. If the information base is large, the author stands a good chance of becoming disoriented during this process. Furthermore, if the author is working with a system based on mark-up tags, for example, SGML or TEX then they are more than likely forced to work at an abstraction level comparable to assembly language. Thus, view creation is a difficult and error prone task. Examples of typical errors are; a dangling associative link, or a link to an erroneous destination presentation. Erroneous structures can also result from systems which do not (or cannot!) enforce structural integrity. As the size and sophistication of the application grows, so does the need for view management tools. Central here is the ability to judge the integrity or quality of a view. In this section two criteria are proposed for judging views. The criteria are *cohesion* and *relevance*. Redundancy and irrelevance are particular extremes of these criteria.

The properties of views can be considered in terms of its substructures. In our analysis, focus is centred on a major component within views, the *sequence*.

Sequences

A sequence occurs in a number of ways within views. The most familiar of these is what is commonly referred to as a *path*. Such sequences are typically generated by rules containing the Kleene-star or Kleene-plus operator in their right-hand side. For example,

$$\mathsf{S} \rightarrow \mathsf{M}^+$$

This grammar rule generates a sequence of one or more molecules of class M. Another variation of a sequence is a sequence of fragments *within* a node. This usually occurs in conjunction with text fragments which are to be read sequentially.

Cohesion of Sequences

Cohesion is a measure of the *connectedness* or *togetherness* of the sequence. A sequence is connected if there is sufficient information overlap between the successive objects. Cohesion can be considered as a range: At the low end of the range, we term a sequence *disjointed* when there is very little information overlap between successive objects. At the high end of the range, we term a view *redundant* if the searcher is constantly confronted with the same or similar information. We do not contend that redundancy is 'bad'. In fact, controlled redundancy is a constructive way to help the searcher stay with a particular idea or line of thought. For example, certain aspects are summarized (repeated) at different places in the sequence. In other words, redundancy is not only the repetition of information but is also dependent on the distance between the information that is repeated. If this distance is short then the repetition is probably not constructive and may in fact be annoying for the searcher.

The distance between objects in a sequence is simply the distance within the sequence. Cohesion is a function of the overlap of the objects in the sequence modified by their distance apart. For most purposes, the following is a good approximation of cohesion:

$$\textsc{Cohesion}(x,y) = \frac{\textsc{Overlap}(x,y)}{\textsc{Distance}(x,y)}$$

The cohesion of a sequence can be represented as a directed graph in which the arcs between two nodes are annotated with the cohesion value of the two associated objects.

The cohesion graph can be used to detect *disjointedness* and *redundancy* as follows: If there are many arcs that have very low values between nodes that are close together in the graph, then this is a reflection of disjointedness. On the other hand, if the arcs of nodes that are close together have relatively high values and arcs connecting nodes that are further apart have mostly values greater than zero, then this is a reflection of redundancy.

Example 4.5.1

> *Figure 4.20 is the graphical representation of the cohesion of the* River Pollution and Salmon Migration *view. (See page 97).* □

To illustrate how the values on the arcs are calculated consider the arc which connects nodes 2 and 5. Earlier calculations revealed that the overlap between these nodes is 0.417. In the graph they have a distance of 2. So their cohesion is $\frac{0.417}{2}$ which equals 0.208.

On the basis of the above graph the author may decide that the value on the arc between nodes 2 and 5 is rather high considering the distance between the nodes. This value is due to the redundant information in both nodes.

The question arises as to what can be done if the author decides that the redundancy is not acceptable. This can sometimes be alleviated by splitting the redundant information from the objects involved and forming this into a new object [Sha85].

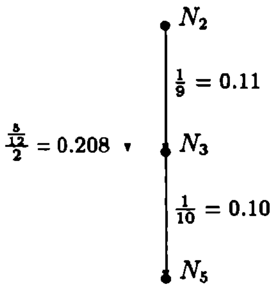

Figure 4.20: Cohesiveness Graph

Intention of Sequences

Sometimes a sequence has a specific purpose. This is normally the case with paths because these are defined by the author to support a theme or intention. For example, the intention of the nodes in the *Lake Pollution* view (see page 97) is that it should provide information about the pollution of lakes. In our model we formalize the intention of a sequence by characterizing it. That is, the intention can be represented, for example, by a set of descriptors. The characterization of an intention is denoted by $\bar{\mu}$.

The question arises as to how well the sequence reflects its intention. For example, the *Lake Pollution* sequence is not a good reflection of the intention because node N_3 of this view has nothing to do with the pollution of lakes. We say that N_3 is irrelevant with respect to the intention of the view.

The relevance of a given sequence with respect to its intention can be quantified by determining the information overlap of every object in the sequence with the intention.

Example 4.5.2

Given the intention

$$\bar{\mu} = \{\text{pollution, lake}\}$$

then the relevance of the Lake Pollution *with respect to* $\bar{\mu}$ *is:*

$$\textsc{Overlap}(N_4, \bar{\mu}) = \tfrac{2}{4} = 0.5$$

$$\textsc{Overlap}(N_3, \bar{\mu}) = \tfrac{0}{4} = 0$$

□

A sequence can be deemed as being relevant if many of the objects in the sequence have a non-zero overlap with the intention. A typical application is writing teaching material for courses covering material that has been defined by a list of chapters of different books. The intention, then, is the cumulative characterization of the material defining chapters. The overlap of the textbook with this intention then is a measure to what extend the material has been covered in the textbook.

Another application is marking of exercises that ask for the overview of some part of the material. The mark can be directly derived from the overlap between the (characterization of the) material, and the work of the student.

4.6 Beaming between Layers

Imagine that the searcher has navigated through the hypcrindex and is satisfied with the information need as expressed in the guide. (S)he now wishes to retrieve the molecules in the underlying hyperbase which fit the description constituted by the guide. Within the stratified architecture such a retrieval operation is realized by interlayer navigation in the form of a *beam down* operation. The trigger for the beam down is a request which is in part derived from the guide. This request has two components

1. A set q of descriptors from the hyperindex which approximates the information need.
2. A *scope* which specifies a syntactic filter for molecules in the hyperbase.

The descriptive part of the request is derived readily from the context(s) within the guide. The scope aspect of the request will be described in more detail shortly.

The request is passed to an Information Disclosure Machine which realizes part of the functionality of the information processor (see figure 4.1). In order to understand how a request of the above form is satisfied, it is necessary to precisely define how the stratified hypermedia relates to the disclosure structure of the Disclosure Machine. Basically, we provide a mapping from a two level hypermedia to the disclosure structure of the machine. Recall that the Disclosure Structure D is a system consisting of $\mathcal{O}$, a characterization language $\mathcal{C}$ and the indexing relation χ (see definition 3.2.1). If $\hbar$ is a hyperindex layer and $\mathcal{H}$ an underlying hyperbase, the mapping is defined as follows:

- $\mathcal{O} = \bigcup_{v \in \mathcal{H}_{\mathcal{V}}} M_v$
- $\mathcal{C} = \hbar_{\mathcal{F}}$
- $\chi \subset \mathcal{O} \times \mathcal{C}$

In other words, the objects comprise the molecules of the hyperbase, the characterization language is the fragment base of the hyperindex and χ defines a relation between the molecules in the hyperbase and the fragments in the hyperindex.

The Disclosure Machine, for example the Refinement Machine, when fed with q, evaluates the function $P_{\text{Rel}}(O, q)$ for all $O \in \mathcal{O}$. Those molecules with a sufficiently high probability of relevance are passed through the scope filter. The following strategies can then be adopted (see figure 4.21):

1. Retrieve the *most general* molecules only. This is usually done in existing disclosure systems. For example, in libraries, it is sufficient to deliver the location codes of relevant books, as books can only be taken as a whole from the shelf.

2. Retrieve the *most specific* molecules only, that is, molecules that are sufficiently relevant, but whose descendants are not relevant enough. In this approach, the information need is satisfied in its finest granularity. The advantage is the minimization of time spent perusing objects in order to find the answer to a specific question ([WD91]).

3. Retrieve by *structural element*. In this case the scope is specified by a non-terminal symbol in the hyperbase layer. For example, the system can be asked to only retrieve relevant sections.

4. Retrieve the *relevant subtree*, that is, all relevant molecules as a separate parse tree.

In order to realize the scope filter, the Disclosure Machine defined in Chapter 3 needs to be enhanced with extra functionality. For example, a function which returns a molecule's syntactic class.

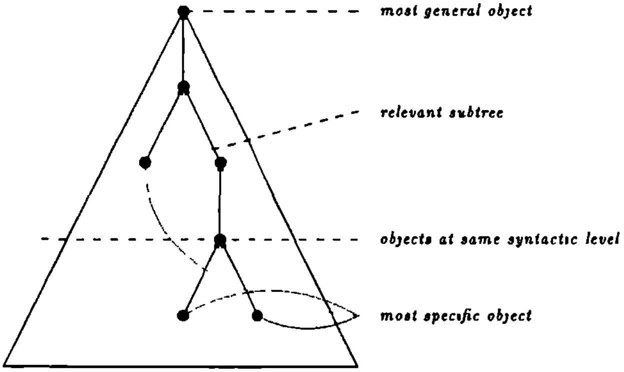

Figure 4.21: Strategies for beam down

The relevant molecules satisfying the scope condition are used as the basis of a dynamic view on the hyperbase. This view is a way of parcelling the query result and has the following characteristics. The start symbol of the view is a special symbol S, which does not take part in any grammar rule. As a consequence, each parse tree consists of a single molecule. The actual structure contains one parse tree M, which forms a stepping stone to the retrieval result. The presentation of this molecule M contains a fragment identifying each potentially relevant molecule. The fragments are ordered with respect to decreasing likelihood of relevance of the associated molecule with respect to q. Emerging from each such fragment is an associative link which runs to the presentation of the associated molecule.

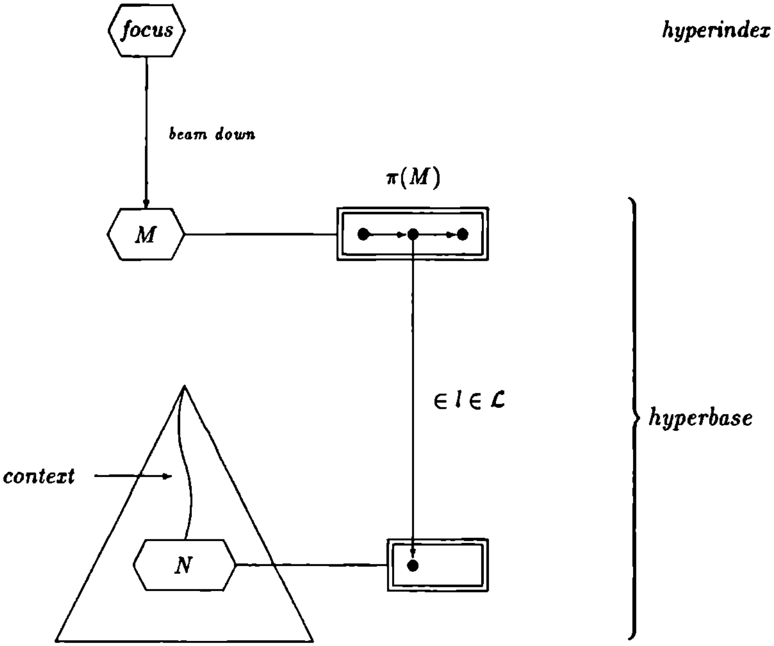

Figure 4.22: The beam down operation

Note that traversing such an associative link brings about a shift of the current context, as it lands the user into the parse tree which the molecule is a part of.

Sometimes the searcher can become lost in the hyperbase. In hypertext jargon this is commonly referred to as *lost in hyperspace.* In such situations a *beam up* operation can be performed which transfers the user from the level of the hyperbase to the level of the hyperindex. This operation can be compared to a lost wanderer in the forest who climbs a tall tree to look at the forest from above. The result of the beam up operation is a guide which is derived from the characterization of the molecule associated with the presentation which was current at the time of invocation of the beam. The most specific descriptor(s) in the characterization are used for constructing the guide as these give the most detailed reflection of the focus at the time of the beam. The searcher can use any of the contexts in the guide as a starting point for Query by Navigation. In this way the searcher can hopefully quickly reorient and return again to the hyperbase with a *beam down.*

Chapter 5

Musings on the Disclosure Effectiveness of the Stratified Architecture

> *The rule of thumb became that if strategy A was different by at least 5% to strategy B on a number of (document) collections, then we could infer that it was indeed different. The intuition was, that, had we a sophisticated statistical theory, it would undoubtedly support the conclusion. This approach to information retrieval has continued into the eighties but there are distinct signs that it has run its course.*
>
> C.J. van Rijsbergen

The previous chapter introduced the stratified information disclosure architecture. The user can search for information by browsing, for example, within the hyperbase. Another possibility is Query by Navigation. This latter approach is comparable to the matching process of the older information retrieval paradigm (see figure 1.1) in the following sense. After developing a query guide by navigating through the hyperindex layer the searcher uses this guide to trigger interlayer navigation in the guise of the the *beam down* operation. The basis of the *beam down* is a matching process driven by an Information Disclosure Machine which in turn is a part of the functionality of the information processor.

A pertinent question is whether such an architecture really does offer effective information disclosure. In this chapter we will ponder this question. Before a detailed presentation of effectiveness results, it is worthwhile to look again at the methodology traditionally employed to measure and judge information disclosure effectiveness.

In the above quote "strategy" refers to a particular mechanism of disclosure. For example, one could unleash two disclosure machines M_1 and M_2 over a number of document collections with the same queries, and compare the recall and precision results. If machine M_1

had more that five percent better average recall and precision figures, then one concludes that M_1 is a more effective disclosure machine than M_2. As mentioned earlier in chapter 1, this paradigm has gained notoriety due to the Cranfield experiments performed in the early sixties and is sometimes referred to as the *Cranfield Paradigm*. The quote suggests that this framework is losing its pertinence, the reason being that information disclosure is no longer confined to the controlled library environment, but has become an issue much wider in scale [Rij89]. It seems that new paradigma are needed to explore the issue of disclosure effectiveness. Unfortunately there is as yet no theory with which the effectiveness of two disclosure mechanisms can be compared *inductively*. For this reason, it is unfortunately not possible to present a theorem (with associated proof) that the stratified architecture offers better disclosure than, for example, the vector space model. Therefore, to study the potential effectiveness of the stratified architecture we have adopted an experimental approach according to the Cranfield Paradigm. First off the effectiveness of Query by Navigation over a lithoid-based hyperindex is discussed in the framework of traditionally organized experiment. Thereafter, the Refinement Machine is studied in an attempt to shed light on the question of its suitability as the vehicle for the *beam down* operation.

5.1 Effectiveness of Query By Navigation over an Art History Lithoid

The slides library (*slibrary*) of the Department of Art History at the University of Nijmegen is a manual catalogue of 90,000 slides that has been built up over a period of forty years. During this period different students and co-workers have been involved in cataloging slides depicting pieces of art or architecture. A user can search for a slide(s) via a number of different avenues. Some searches occur via a single, or combination of, extensional characteristics associated with an art object. For example, the name of the artist or the technique used to create the art object. These searches are typically straightforward because the information need is clear cut. More difficult searches are directed at the subject or intention of the art object. For example, the retrieval of all slides of paintings depicting babies. In order to satisfy such information needs the hierarchical thesaurus *Iconclass* is used (see section 4.13). The goal of organizing the thesaurus in the form of a hierarchy is to create a concept space which is easily browsable by the searcher.

Imagine a slide depicting a painting of a man walking through a field with his dog and a flock of sheep. Which classifications should be used to characterize this slide? `25F (animals)`, `25F2 (mammals)`, `25H (landscapes)`, `34 (man and animal)`, `34A (taming and training of animals)`, `34B11 (dog)`, `34C (protection of animals)`, `43B1 (enjoying nature)`, `46A14 - (farmers)`, `47I221 (herding)`, or `73B25 (of the shepherds)`? It is clear that for both the librarian, who characterizes the slides, and the searcher, a choice of classification is not always easy to make.

Another problem with using Iconclass as a disclosure mechanism stems from its hierarchical structure. While moving down the hierarchy it is sometimes not clear to the searcher if the desired target facet will be a subfacet of the current one. For example, when trying to find all slides of artwork depicting babies the searcher is immediately presented with the following dilemma. Is baby a subclassification of 3 (Human being, man in general), or 4 (Society, civilization, culture) (see figure 4.11)? It turns out that it is a subclassification of the latter, which some searchers do not expect.

Due to the above problems searchers were sometimes encountering difficulties in retrieving relevant slides. In some cases they eventually resorted to costly sequential searches through the catalogue in the hope to find relevant slides. In order to provide better disclosure of slides it was decided to use their title descriptions as a basis of a disclosure mechanism. This is because a fair amount of the information about the subject of the particular art object can be regularly found in the slide's title description. Typical examples of title descriptions are Mary with Child and Vlucht naar Egypte. As the last title demonstrates, the descriptions are not always in English. The slibrary contains descriptions in French, German, Italian, English and Dutch. Clearly any disclosure system based on the title descriptions would have to deal with multi-lingual aspects. The disclosure mechanism chosen was a lithoid based hyperindex over which the searcher could navigate in the way depicted in section 4.2.7.3 of chapter 4. The lithoid was constructed as described in section 2.3.2 of chapter 2. That is, index expressions were derived from the title descriptions. Thereafter, the power index expressions were generated and united to form the lithoid. The question was whether such a lithoid would offer any improvement in the disclosure of the information in the catalogue.

5.1.1 Effectiveness Criteria for Navigable Disclosure Systems

Both Iconclass and the lithoid are examples of navigable disclosure systems. As such they can readily be realized in the form of hyperindices (see section 4.2.7). The question arises as to how the disclosure effectiveness of such systems can be judged.

Recall and precision are measures of effectiveness which are employable once the searcher is satisfied with his or her query guide and wants to retrieve the relevant objects by performing a beam down operation. However, before this point was reached there was navigation involved. For example, in the case of the lithoid, the searcher has traversed a certain path to arrive at a target expression. Similarly, when using Iconclass a path was followed through the classification hierarchy to arrive at a particular target classification. There must be criteria to judge this process since if much navigation was involved then this could mean that the searcher had difficulty in finding his or her way through the disclosure system at hand.

Research on the effectiveness of navigation within a disclosure system has its roots in studies of the disclosure effectiveness of printed subject indexes [Kee77] [Kee78]. In these studies criteria such as search time, number of pages turned, were used to assess the browsability of subject indexes. These criteria were further developed in Craven's research on on-line subject indexes [Cra86]. Craven defines the notion of the *predictability* of a disclosure system as being dependent on the extent to which a searcher can predict where relevant index entries can found. Although Craven defined predictability as a criterium for judging on-line subject indexes, it can nevertheless applied to navigable disclosure systems, such as hyperindices, by formally expressing predictability in terms of *logical decisions* [Bla90].

This notion can be explained in terms of a lithoid hyperindex: At each focus the searcher can de-ambiguate the contexts denoted by the focus by refining it, or conversely, the pertinent contexts can be made more ambiguous by enlarging the said focus. A refinement or enlargement operation means that the searcher has chosen to transform the focus in such a way that they think the resulting index expression is a better description of their information need. Such a choice is an example of a logical decision. Logical decisions are also involved in browsing thesauri [CD90]: While searching a faceted hierarchical thesaurus such as Iconclass, a choice must be made between the available subfacets.

When a large number of logical decisions are typically involved while searching a given navigable disclosure system, this probably suggests that searchers cannot predict how to navigate to relevant descriptors. They are becoming "lost" within the disclosure system. Therefore, the *predictability* of a navigable disclosure system is defined as being inversely proportional to the number of logical decisions necessary to find a relevant descriptor.

Once a searcher has found a relevant descriptor, the question arises as to whether the descriptor found is most relevant with respect to the searcher's information need. In other words, there could be other, more relevant descriptor(s) in the hyperindex of which the searcher is not aware. It is therefore important that relevant descriptors be near each other in the disclosure system so that when the searcher finds a relevant descriptor, other relevant descriptors are also readily found. This aspect of a disclosure system is referred to its *collocation*. Collocation is informally introduced by Craven [Cra86] and a formalization of this criterium can be found in [Bru90]. Note that high collocation is desirable for exhaustive searches, that is, searches in which as many as possible of the relevant objects must be found.

Now that criteria are available with which a navigable disclosure system can be judged, the task now is to employ them suitably in an experimental setting.

5.1.2 The Experimental Setting

The objective of the experiment was to test how effectively searchers could satisfy predefined information needs using either Iconclass or the lithoid-based hyperindices. For this purpose a two level hypermedia was implemented. A hyperbase of 437 catalogue cards of slides depicting various topics within art history was set up as a flat hypermedia with an empty associative link scheme. As a consequence, it was not possible to navigate within the hyperbase. As Iconclass officially contains twenty three thousand facets, it was decided only to implement a subset. The subset was realized by restricting the depth of hierarchy. The resulting facets were used to implement a flat hyperindex featuring a single hierarchical link type which was used to simulate the hierarchical structure of Iconclass. Using the 437 title descriptions, a lithoid was automatically constructed. It contained 2434 expressions, of which 650 were terms. The lithoid was also implemented as a flat hyperindex. The 650 terms implied that the start node (see figure 4.15) comprised 650 entries. To facilitate perusal in this large node, a *finder* was implemented using the "word wheel" technique: A searcher is presented with a window within which they can type-in a term. After typing the first letter a pop-up window appears showing all terms that begin with that letter arranged in alphabetical order. As the searcher continues to type, the entries in the pop up window are adjusted accordingly. At any time the searcher can choose a term in the pop up window, thus avoiding further typing. This choice becomes the initial focus for Query by Navigation within the lithoid-based hyperindex.

Eight queries were devised describing information needs that could all be answered using either hyperindex (see figure 5.1). The motivation behind most queries was to test recall. For example, query 8 seems simple enough, but in art history *Maria* appears in may different contexts. Thus it is difficult to achieve good recall. Other queries were specifically devised to promote navigation over the disclosure system in question. For example, query 1 forces Iconclass searchers to look deep within the Iconclass hierarchy and query 2 forces searchers of both systems to navigate until a desired number of slides have been found. For each query, a set of relevant objects was established by the librarian so that recall and precision measures could be calculated.

Q1	*Find slides depicting the Annunciation*
Q2	*Find three slides depicting flowers*
Q3	*Find slides depicting French city-views by Van Gogh*
Q4	*Find a slide depicting an equestrian statue of Aurelius*
Q5	*Find slides with harvest scenes*
Q6	*Find slides depicting Greek gods*
Q7	*Find slides depicting the passion of Christ*
Q8	*Find slides depicting Maria*

Figure 5.1: The queries for Iconclass and the art-history lithoid

The queries were tested on a group of fourteen people. Ten of these had a background in art history, the other four were computer scientists with little or no knowledge in this area. The first group was divided into two groups of five, one group being tested on the lithoid, the other on Iconclass. The motivation for involving computer scientists was twofold. First, we wanted to test the effectiveness of the lithoid in regard to *naive*(in the art history sense) searchers. Secondly, as the computer scientists have knowledge of structures such as lattices, we wanted to test if they would exploit the lattice-like structure of the lithoid more effectively than the art historici.

Although both Iconclass and the lithoid accessed the same underlying hyperbase, they were not integrated into one system at the time of the experiment. Searchers could therefore only use the disclosure mechanism assigned to them to answer the queries. In order to determine the predictability logging was used to record actions corresponding to a logical decision of the searcher. The log file for each user of the lithoid hyperindex consisted of recordings of the following events: Selecting a term in the term finder, refining or enlarging the current focus or a *beam down*. For Iconclass users the recorded events were: Each classification selected or a *beam down*.

Query	*Recall*	*Precision*	*Logical decisions*
1	0.56	1 00	14.0
2	0.30	0 70	15.2
3	0.34	0.60	14.7
4	0.00	0.00	∞
5	0.55	1.00	13.8
6	0.77	0.98	8.2
7	1.00	1.00	6.8
8	0.26	0.98	11.2
Average	0.47	0.78	13.0

Figure 5.2: Effectiveness of Art Historici using Iconclass

Query	*Recall*	*Precision*	*Logical decisions*
1	1 00	1.00	3.4
2	0.65	1.00	15.8
3	0.71	0.61	9.3
4	1.00	1 00	4.0
5	0.80	1.00	5.6
6	0.43	0.92	12.8
7	0.30	0.94	14.4
8	0 54	0.84	6.8
Average	0.68	0.91	9.0

Figure 5.3: Effectiveness of Art Historici using the lithoid

5.1.3 Summary of Experimental Results

The average *recall* and *precision* results in figures 5.2 and 5.3 show no significant (Wilcoxon test, $\alpha = 0.05$) improvement of the lithoid over Iconclass with regard to these criteria. Furthermore, it cannot be concluded that the lithoid is more predictable, even though the average number of logical decisions required was less.

Precision

As the searcher moves down the classification hierarchy or performs refinement operations the focus becomes more and more descriptive or specific. Such descriptors tend to have higher precision and lower recall [Sal89]. It is therefore not surprising that the precision of both disclosure systems was generally high.

Recall

Problems occurred with both systems regarding recall. With regard to Iconclass queries 2,3,4 and 8 demonstrated low average recall. This can be attributed to the following:

- Searchers are sometimes confused as to which classification is relevant to the information need. For example, the slide depicting the equestrian statue of Aurelius (Query 4) is classified under `traffic on land`. This was so obscure none of the Iconclass searchers could find it.

- The collocation is poor in some cases. For example, in query 8 most searchers found the classification `Virgin Mary` under the main classification `Religion and Magic` and then stopped looking. There were, however, many other relevant classifications such as Mary's coronation and the Birth and youth of Christ. Searchers were not prompted to look for them because these subclassifications exist under *another* main classification.

The poor recall with regard to the lithoid can also sometimes be attributed to poor collocation. For example, once the searcher has found Maria (Query 8), it is also desirable that Madonna would also be found. This is also true for queries 6 and 7. When trying to find Greek gods (Query 6), searchers using the lithoid hyperindex typically began with the term gods in the finder, but only a few slides were characterized by this term. It was necessary to navigate to Zeus, Hercules etc., but this was not possible because in order to establish a relationship between gods and Zeus, these terms must appear together in a slide title. This suggested that the terms (the simplest index expressions) be sometimes further structured via thesaural relations.

Predictability

The Iconclass hyperindex can in some cases be judged as unpredictable because large numbers of logical decisions were necessary to fulfill a particular information need. This was basically because the searchers do not know what subfacets they can expect within a facet which is currently the focus. For example, for religious subjects it is not clear whether to look under Religion and Magic or The Bible. When a wrong choice has been made the searcher is forced to backtrack up the hierarchy and then try an alternative subfacet thereby resulting in more logical decisions. The extreme case of this was query 4, in which no Iconclass searcher succeeded. (The average searcher gave up after 20 logical decisions, meaning ∞ in figure 5.2 is 20).

Poor collocation also adversely affects predictability. Even, if the searcher is aware of relevant facets they are sometimes far from each other. For example, in trying to answer query 8, a minimal of 5 logical decisions are necessary to navigate from Virgin Mary to Mary's coronation.

The lithoid-based hyperindex lacked predictability due to the following reasons:

- Some searchers did not realize that refining the focus does not produce more relevant slides, thus resulting in more logical decisions.
- Disorientation. Searchers forgot where they had been in the hyperindex.
- Searchers often did not choose the enlargements of the current focus: Seeing a relevant term in an index expression, they escaped back to the finder and began refining this term.
- Due to the lack of thesaural relations between terms, the searchers were sometimes forced to browse aimlessly with the finder in the hope of finding relevant terms. (The large number of logical decisions associated with query 6 demonstrated this clearly).

Performance of the Computer Scientists

Figure 5.4 summarizes the effectiveness of computer scientists using the lithoid-based hyperindex. In comparing these results with those of the art historici (see figure 5.3) it can be concluded that computer scientists were as effective as the art historici with regards to recall and precision (Wilcoxon test, $\alpha = 0.05$).

The navigation of the computer scientists involved, on average, less logical decisions than the art historici (Wilcoxon test, $\alpha = 0.05$). Furthermore, it seems that they could exploit the lattice-like structure of the lithoid-based hyperindex. An example of this was query 3. Here all computer scientists began the search with the term Paris because they lacked the relevant knowledge to begin with place names such as Arles (where Van Gogh produced many of the paintings in his French period). From the focus Paris it was possible to navigate to Arles by refining Paris to view of Paris and enlarging this to view and then refining again to view over Arles. This last focus could the be enlarged to Arles, which characterized quite a few of the relevant slides.

The art historici, on the other hand, typically only chose refinements of the current focus, not realizing that additional contexts could be found by choosing and enlargement. This may suggest that the art historici did not understand fully the underlying structure of the lithoid.

Query	*Recall*	*Precision*	*Logical decisions*
1	1 00	1 00	2 6
2	0 75	1 00	13 7
3	0 82	0 56	9 3
4	1 00	1 00	4 3
5	0 88	1 00	6 5
6	0 44	0 69	12 3
7	0 16	1 00	12 5
8	0 35	0 93	5 9
Average	0 68	0 90	8 4

Figure 5.4: Effectiveness of Computer scientists using the lithoid

Overall Remarks

Possibly one of the main advantages of the lithoid-based hyperindices is that non-expert searchers seem to be able to effectively satisfy their information needs in a domain that is foreign to them. This contrasts with a disclosure system such as Iconclass by which knowledge of the domain is often necessary in order to effectively use the system. (This is the reason that we could not test the effectiveness of computer scientists with the Iconclass hyperindex). Furthermore, the lithoid-based hyperindex can contain multi-lingual index expressions, whereas the Iconclass facets are restricted to English.

The fact that the lithoid can be constructed automatically can be considered a big advantage. However, the disadvantage is that the index expression Announcement of birth of Christ will never be associated with Maria, even though they are highly related. This is because the index expression transducer is a syntactic analyzer and cannot make the connection with Maria. A trained librarian would make this relationship, this being the strength of manual classification.

The recall of lithoid-based hyperindices does benefit by introducing thesaural relations between terms [BB91]. For example, a cross reference relation, which allows terms such as Maria and Madonna to be related. The ISA relation can also be employed. For example, Hercules ISA god. This allows the searcher in the lithoid-based hyperindex to refine the focus god to Hercules.

Another conclusion of the experiment is that the lithoid-based hyperindex and the Iconclass-based one complement each other. For some queries it is more sensible to use a Iconclass. For example, to find many slides dealing with the passion of Christ one need only navigate to the relevant facet in Iconclass. On the other hand, for the searcher who wants to know something about Maria but is not sure what, can better use the lithoid-based hyperindex. By selecting the term Maria in the finder, the contexts in which Maria occur are displayed as refinements. A useful system would be one which incorporates both of these. The integration can be modelled as a hyperindex layer in which the lithoid and Iconclass exist as separate views. A system is currently functional at the Art History department of the University of Nijmegen which was constructed along these lines. It has proven successful enough to be marketed as a product [Sie91].

Finally, more research is needed to better understand the weaknesses and strengths of lithoid-based hyerindices.

5.2 The Experimental Effectiveness of the Refinement Machine

5.2.1 A Small Experiment

As a starting point for discussion of the effectiveness of the Refinement Machine, consider once again the example described in section 3.4 of chapter 3. This example was used to show how a Disclosure Machine with a context-free plausible inference mechanism would assign the same probability of relevance to a document about air pollution in Holland as to one about the effects of pollution of rivers in response to a request riv o poll.

The same problem will now be presented to the restricted Refinement Machine as defined in section 3.5 of chapter 3. This entails that an index expression belief network is constructed from the core set of index expressions,

$$\begin{aligned}\chi(O_1) &= \{\textsf{riv o poll in Australia}\}\\ \chi(O_2) &= \{\textsf{eff of poll in riv}\}\\ \chi(O_3) &= \{\textsf{air o poll in Holland}\}\end{aligned}$$

The topology of the resulting belief network is depicted in figure 5.5. Remember that the probability of a term was estimated using normalized occurrence frequencies. In this example, the term poll occurs three times, once in each document. This leads to a probability

$Pr(\text{poll}) = 0.3333$. The table shown in figure 5.6 summarizes the term probabilities for this example. The conditional probability assessment functions for variables representing binary index expressions are defined using the normalized connector occurrence frequencies shown in figure 3.6. For example, the function value $\gamma_{\texttt{POLL IN RIV}}(\text{poll in riv}|\text{poll} \wedge \text{riv})$ of the conditional probability assessment function $\gamma_{\texttt{POLL IN RIV}}$ for the variable POLL IN RIV is estimated by the probability that the in connector associates the two terms: in this case the value is 0.0632.

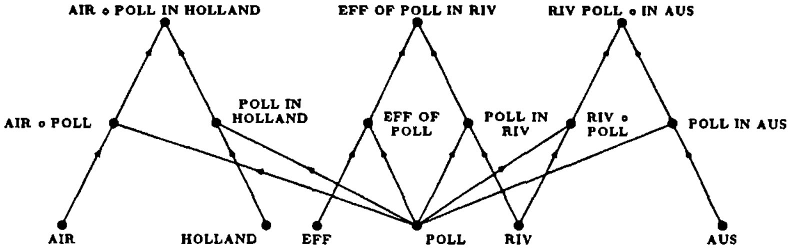

Figure 5.5: The pollution index expression belief network

t	$f(t)$	$Pr(t)$
poll	3	0.33
riv	2	0.22
eff	1	0.11
aus	1	0.11
air	1	0.11
holland	1	0.11

Figure 5.6: Term probabilities in the Pollution Example

The belief network was automatically constructed in thc following way: A C program indexes titles resulting in a core set of index expressions. Another program uses this core to generate the lithoid which is stored in a relational database. From this database a belief network specification is generated which can be fed to the IDEAL system [SB90]. IDEAL is an environment for building and manipulating belief networks. A number of evidence propagation algorithms are supported by IDEAL; in our case, the Lauritzen & Spiegelhalter [LS88] algorithm was employed. In the first phase a so called *join tree* is derived from the belief network. This is an equivalent representation of the belief network in which the direction of the edges have been omitted and certain edges have been added to short cut large cycles. From this representation small subgraphs can be identified and organized into a tree. Due to the manipulations of the network the first phase is computationally expensive (polynomial in the height of the graph), but it need only be performed once. In the second phase, evidence propagation can proceed efficiently through the tree of subgraphs via a chain reaction. The experiment was conducted using a version of IDEAL running in compiled LISP on a SUN 4 computer. Each characterization was separately entered as evidence into the belief network and subsequently propagated; thereafter the probability of the request was computed from the network. This probability corresponds to our belief in the request in the light of the given characterization. The results of this experiment for each of the objects are summarized in the table shown in figure 5.7. These probabilities can be translated directly into relevance judgements using definition 3.5.1.

Evidence	Pr(riv o poll|*Evidence*)
eff of poll in riv	0.55
riv o poll in aus	1
air o poll in holland	0.12

Figure 5.7: Characterizations as Evidence

The results are encouraging. In contrast with a Disclosure Machine with a context free plausible inference mechanism, the Refinement Machine shows a substantial differentiation between Pr(riv o poll | eff of poll in riv) and Pr(riv o poll | air o poll in Holland). The Pr(riv o poll | riv o poll in Australia)= 1 as the request riv o poll is strictly deducible from the characterization riv o poll in Australia via *modus continens.*

5.2.2 A Larger Experiment

The motivation behind the experiment described in this section was to give an indication of the feasibility of the Refinement Machine in a larger practical setting. An index expression belief network was constructed using the first twenty-five document titles and the first three queries of the Cranfield collection [Fox90]. The titles of these documents and the associated requests are depicted in figures 5.8 and 5.9. Note that in figure 5.9 the characterizations of the requests are also given. Why the requests are characterized by ternary expressions will become clear shortly.

The same basic approach was taken to realize the index expression belief network for this experiment as for the smaller experiment described in the previous section. In this case, however, an attempt was made to use more realistic probabilities for the terms. For this purpose, term probabilities were calculated using the first 500 titles of the Cranfield collection. These probabilities ranged between 0.000264 (= Pr(accuracy)) and 0.0397 (= Pr(flow)). In order to lessen the impact of the computational expense involved in the construction of the join tree, the height of the lithoid was restricted. Only index expressions of three terms or less were used. The effect of this measure can be envisioned as follows. A lithoid basically resembles a mountain range, each peak corresponding to a power index expression. Restriction to ternary index expressions or smaller has the effect of lopping of the peaks so that only the common base remains. This in fact captures the most interesting part of the lithoid as this area contains roughly 95% of shared index expressions (see figure 2.24 in chapter 2). The shared index expressions are crucial in distributing evidence through the associated belief network. The twenty-five titles and three queries resulted in a restricted

	Title
1	experimental investigation of the aerodynamics of a wing in a slipstream
2	simple shear flow past a flat plate in an incompressible fluid of small viscosity
3	the boundary layer in simple shear flow past a flat plate
4	approximate solutions of the incompressible laminar boundary layer equations for a plate in shear flow
5	one-dimensional transient heat conduction into a double-layer slab subjected to a linear heat input for a small time internal
6	one-dimensional transient heat flow in a multilayer slab
7	the effect of controlled three-dimensional roughness on boundary layer transition at supersonic speeds
8	measurements of the effect of two-dimensional and three-dimensional roughness elements on boundary layer transition
9	transition studies and skin friction measurements on an insulated flat plate at a mach number of 5.8
10	the theory of the impact tube at low pressure
11	similar solutions in compressible laminar free mixing problems
12	some structural and aerelastic considerations of high speed flight
13	similarity laws for stressing heated wings
14	piston theory - a new aerodynamic tool for the aeroelastician
15	on two-dimensional panel flutter
16	transformation of the compressible turbulent boundarylayer
17	remarks on the eddy viscosity in compressible mixing flows
18	the flow field in the diffuser of a radial compressor
19	an investigation of the pressure distribution on conical bodies in hypersonic flows
20	generalised-newtonian theory
21	on heat transfer in slip flow
22	on slip-flow heat transfer to a flat plate
23	skin-friction and heat transfer characteristics of a laminar boundary layer on a cylinder in axial incompressible flow
24	theory of stagnation point heat transfer in dissociated air
25	inviscid hypersonic flow over blunt-nosed slender bodies

Figure 5.8: The first 25 Cranfield titles

	Request	$\chi(q)$
q_1	similarity laws in construction of aeroelastic models of heated high speed aircraft	similarity ∘ laws in construction laws in construction of aeroelastic construction of aeroelastic ∘ models aeroelastic ∘ models of heated models of heated ∘ high heated ∘ high ∘ speed high ∘ speed ∘ aircraft
q_2	structural and aeroelastic problems with flight of high speed aircraft	structural and aeroelastic ∘ problems aeroelastic ∘ problems with flight problems with flight of high flight of high ∘ speed
q_3	solved problems of heat conduction in composite slabs	solved ∘ problems in composite solved ∘ problems of heat problems of heat in composite problems in composite ∘ slabs problems of heat ∘ conduction

Figure 5.9: The Three Requests and their characterizations

lithoid of 393 index expressions. The corresponding unrestricted lithoid comprises 1007 expressions. Of the 120 index expressions that were shared, 114 of them were ternary index expressions or smaller.

Instead of giving the document characterization i as evidence and examining the belief in the request, the inverse was tried, namely the request q was entered as evidence [1] , propagated, and the belief in characterizations ($Pr(i|q)$) was inspected. This approach was taken for efficiency as it involves only one evidence propagation phase through the belief network for each request. Compare this to the small experiment of the previous section which involved one propagation phase for each characterization. In order to determine probabilities of relevance (see definition 3.5.1) it is necessary to determine $Pr(q|i)$ for each characterization i of an object O. Note that the strategy just described delivers $Pr(i|q)$ not the required $Pr(q|i)$. Using Bayesian inversion, however, the desired result can be calculated as follows:

$$Pr(q|i) = \frac{Pr(i|q)Pr(q)}{Pr(i)}$$

The value $Pr(i)$ signifies the prior probability of characterization i; such prior probabilities can be extracted from the IDEAL system by propagating null evidence, and thereafter examining the belief in i. The prior probability $Pr(q)$ can be arrived at via a single computation using theorem 3.5.1. A consequence of this theorem together with the theorem No Blind Faith is that q's prior probability is determined purely by the subexpressions of q. Plugging these as evidence into theorem 3.5.1 results in q's prior probability.

Taking a look at the above equation, it can be seen that the relationship $Pr(i|q) > Pr(i)$ is crucial for determining potentially relevant objects. The reason for this is that if $Pr(i|q) = Pr(i)$, then $Pr(q|i) = Pr(q)$ meaning that the document characterization i

[1] As we use a restricted lithoid q corresponds to a set $\chi(q)$ of ternary index expressions Propagation of q in fact entailed entering each of these expressions as evidence and propagating them in parallel In belief network terminology this is referred to as *multiple evidence propagation*

does not influence our belief in q. (Note that the case $Pr(i|q) < Pr(i)$ cannot occur as we propagate only the affirmations of expressions, never their negations). Formulated differently, $Pr(i|q) > Pr(i) \Rightarrow Pr(q|i) > Pr(q)$ and any characterization that increases our belief in the q identifies a potentially relevant document. So, when viewing the results that are presented in the next section it is important to note those characterizations for which our belief increased in the light of request q.

Summary of Experimental Results

One of the first impressions given by results given in figures 5.10, 5.11 and 5.12 is that some of the characterizations of the documents and queries seem strange in a natural language sense. For example, effect of controlled at supersonic. This due to restricting the underlying lithoid to ternary expressions or smaller. In a way, such expressions can be viewed as intermediate steps in the plausible inference process. When the plausible inference mechanism uses the full lithoid, these intermediate results acquire beliefs which are propagated upwards to expressions that in general do make sense.

Another remarkable aspect is that many beliefs are zero, and most of the non-zero beliefs are very small. The zero beliefs are in fact a product of IDEAL's imprecision. Due to the small prior probabilities associated with variables corresponding to unary index expressions, propagation of belief to higher order expressions involved such small numbers which were beyond IDEAL's level of precision[2]. Only one prior probability of a ternary expression was non-zero: Pr(laminar $\circ$ boundary $\circ$ layer) =*2.1E-6* (See $\chi(O_4)$ and $\chi(O_{23})$). The reason for this was that the expression boundary $\circ$ layer occurs frequently enough to push the prior probability of laminar $\circ$ boundary $\circ$ layer above the IDEAL's precision level. This value can be considered as a "background value" for the prior probabilities of other ternary expressions, in the sense that the these prior probabilities must be less than this value.

To shed some light on the potential effectiveness of the Refinement Machine we now examine the beliefs in document characterizations resulting from the propagation of individual queries. As a yardstick the relevance judgements displayed in figure 5.13 are used. These relevance judgements were extracted from the judgements used in the original Cranfield experiments [Fox90].

Query 1

With regard to this request, documents 12, 13, 14 and 15 are deemed relevant (see figure 5.13). We see from the result tables that documents O_5, O_{12}, O_{13} are revealed as being potentially relevant as they contain descriptors i such that $Pr(i|q) > Pr(i)$. Note that the non-zero beliefs in generalised-newtonian $\circ$ theory in $\chi(O_{20})$ and piston $\circ$ theory in $\chi(O_{14})$ signify prior probabilities of the respective expressions and are therefore not interesting.

Consider the beliefs in the characterizations of document 12. The belief in characterizations containing high $\circ$ speed are non-zero because high $\circ$ speed $\circ$ aircraft $\in \chi(q_1)$ (see figure 5.9)

[2] *precision*, not in the information disclosure sense of the word, rather in the computational sense

	x	$Pr(x\|q_1)$	$Pr(x\|q_2)$	$Pr(x\|q_3)$
$\chi(O_1)$	*investigation of aerodynamics in slipstream*			
	experimental ∘ investigation of aerodynamics			
	aerodynamics of wing in slipstream			
	investigation of aerodynamics of wing			
$\chi(O_2)$	*past ∘ flat ∘ plate*			
	flow ∘ past ∘ flat			
	shear ∘ flow ∘ past			
	simple ∘ shear ∘ flow			
	fluid of small ∘ viscosity			
	incompressible ∘ fluid of small			
	plate in incompressible ∘ fluid			
	flat ∘ plate in incompressible			
$\chi(O_3)$	*boundary ∘ layer in simple*			
	layer in simple ∘ shear			
	past ∘ flat ∘ plate			
	flow ∘ past ∘ flat			
	shear ∘ flow ∘ past			
	simple ∘ shear ∘ flow			
$\chi(O_4)$	*boundary ∘ layer ∘ equations*			
	laminar ∘ boundary ∘ layer	2.1E-6	2.1E-6	2.1E-6
	equations in shear ∘ flow			
	equations for plate in shear			
	layer ∘ equations for plate			
	layer ∘ equations in shear			
	incompressible ∘ laminar ∘ boundary			
	solutions of incompressible ∘ laminar			
	approximate ∘ solutions of incompressible			
$\chi(O_5)$	*double-layer ∘ slab ∘ subjected*			
	small ∘ time ∘ internal			
	input for small ∘ time			
	heat ∘ input for small			
	linear ∘ heat ∘ input			
	conduction into double-layer ∘ slab			
	conduction to linear ∘ heat	7.3E-6		7.3E-6
	conduction into double-layer to linear			
	heat ∘ conduction into double-layer			
	heat ∘ conduction to linear			1 3E-5
	transient ∘ heat ∘ conduction			0 0009
	one-dimensional ∘ transient ∘ heat			
$\chi(O_6)$	*flow in multilayer ∘ slab*			
	heat ∘ flow in multilayer			
	transient ∘ heat ∘ flow			1 8E-5
	one-dimensional ∘ transient ∘ heat			
$\chi(O_7)$	*boundary ∘ layer ∘ transition*			
	effect on boundary ∘ layer			
	controlled ∘ three-dimensional ∘ roughness			
	effect of controlled ∘ three-dimensional			
	effect at supersonic ∘ speeds			
	effect at supersonic on boundary			
	effect of controlled at supersonic			
	effect of controlled on boundary			
$\chi(O_8)$	*three-dimensional ∘ roughness ∘ elements*			
	two-dimensional and three-dimensional ∘ roughness			
	two-dimensional on boundary ∘ layer			
	two-dimensional and three-dimensional on boundary			

Figure 5.10: Results

	x	$Pr(x \mid q_1)$	$Pr(x \mid q_2)$	$Pr(x \mid q_3)$
	effect of two-dimensional and three-dimension			
	effect of two-dimensional on boundary			
	measurements of effect of two-dimensional			
	boundary ∘ layer ∘ transition			
$\chi(O_9)$	*insulated ∘ flat ∘ plate*			
	skin ∘ friction ∘ measurements			
	mach ∘ number of 5.8			
	studies and skin ∘ friction			
	studies at mach ∘ number			
	studies on insulated ∘ flat			
	studies at mach on insulated			
	studies and skin at mach			
	studies and skin on insulated			
	transition ∘ studies and skin			
	transition ∘ studies at mach			
	transition ∘ studies on insulated			
$\chi(O_{10})$	*theory of impact ∘ tube*			
	theory at low ∘ pressure			
	theory of impact at low			
$\chi(O_{11})$	**free ∘ mixing ∘ problems**		*3.4E-6*	*3.4E-6*
	laminar ∘ free ∘ mixing			
	compressible ∘ laminar ∘ free			
	solutions in compressible ∘ laminar			
	similar ∘ solutions in compressible			
$\chi(O_{12})$	**high ∘ speed ∘ flight**	*0.001*	*0.5514*	
	considerations of high ∘ speed	*8.1E-5*	*8 1E-5*	
	aerelastic ∘ considerations of high			
	structural and aerelastic ∘ considerations			
	some ∘ structural and aerelastic			
$\chi(O_{13})$	*stressing ∘ heated ∘ wings*			
	laws for stressing ∘ heated	*4.6e-6*		
	similarity ∘ laws for stressing	*8.4e-6*		
$\chi(O_{14})$	**piston ∘ theory**	*2.6E-6*	*2.6E-6*	*2.6E-6*
	aerodynamic ∘ tool for aeroelastician			
	new ∘ aerodynamic ∘ tool			
$\chi(O_{15})$	*two-dimensional ∘ panel ∘ flutter*			
$\chi(O_{16})$	*turbulent ∘ boundary ∘ layer*			
	compressible ∘ turbulent ∘ boundary			
	transformation of compressible ∘ turbulent			
$\chi(O_{17})$	*compressible ∘ mixing ∘ flows*			
	remarks on eddy ∘ viscosity			
	remarks in compressible ∘ mixing			
	remarks on eddy in compressible			
$\chi(O_{18})$	*diffuser of radial ∘ compressor*			
	field in diffuser of radial			
	flow ∘ field in diffuser			
$\chi(O_{19})$	*investigation of pressure ∘ distribution*			
	investigation in hypersonic ∘ flows			
	investigation on conical ∘ bodies			
	investigation in hypersonic on conical			
	investigation of pressure in hypersonic			
	investigation of pressure on conical			
$\chi(O_{20})$	**generalised-newtonian ∘ theory**	*1.3E-6*	*1.3E-6*	*1.3E-6*

Figure 5.11: Results (continued)

	x	$Pr(x\|q_1)$	$Pr(x\|q_2)$	$Pr(x\|q_3)$
$\chi(O_{21})$	*transfer in slip ∘ flow*			
	heat ∘ transfer in slip			
$\chi(O_{22})$	*transfer to flat ∘ plate*			
	heat ∘ transfer to flat			
	slip-flow ∘ heat ∘ transfer			
$\chi(O_{23})$	*axial ∘ incompressible ∘ flow*			
	laminar ∘ boundary ∘ layer	*2 1E-6*	*2.1E-6*	*2.1E-6*
	characteristics of laminar ∘ boundary			
	characteristics in axial ∘ incompressible			
	characteristics in axial on cylinder			
	characteristics of laminar in axial			
	characteristics of laminar on cylinder			
	transfer ∘ characteristics of laminar			
	transfer ∘ characteristics in axial			
	transfer ∘ characteristics on cylinder			
	heat ∘ transfer ∘ characteristics			*4.3E-6*
	skin-friction and heat ∘ transfer			
$\chi(O_{24})$	*theory in dissociated ∘ air*			
	theory of stagnation in dissociated			
	point ∘ heat ∘ transfer			*1 4E-5*
	stagnation ∘ point ∘ heat			*4.9E-6*
	theory of stagnation ∘ point			
$\chi(O_{25})$	*blunt-nosed ∘ slender ∘ bodies*			
	flow over blunt-nosed ∘ slender			
	hypersonic ∘ flow over blunt-nosed			
	inviscid ∘ hypersonic ∘ flow			

Figure 5.12: Results (continued)

Request	*Document*
q_1	$O_{12}, O_{13}, O_{14}, O_{15}$
q_2	O_{12}, O_{14}, O_{15}
q_3	O_5, O_6

Figure 5 13: Relevance Judgements

and **high ∘ speed ∘ aircraft** $\vdash_{MC}$ **high ∘ speed**, therefore rendering maximal belief in **high ∘ speed**. Note that characterizations containing **aerelastic** have a zero belief factor due to the misspelling of this term in the original title. Dealing with spelling variations is a common problem in information disclosure, and the Refinement Machine apparently could do nothing to get around it.

The belief in **high ∘ speed ∘ flight** is 0.001. (See $\chi(O_{12})$). Even though this value is very small in an absolute sense it is nevertheless thousands of times greater than its prior probability. It has been argued that relative changes in belief values can often be more significant than absolute values [GW].

Two beliefs in characterizations of document 13 are non-zero, thereby making it interesting. Consider the expression **similarity ∘ laws** which has a maximal belief because it is strictly deducible from an expression in $\chi(q_1)$. However the plausibility of

$$\textsf{similarity} \circ \textsf{laws}, \textsf{laws for stressing} \mathrel{|\!\!\sim}_{PI} \textsf{similarity} \circ \textsf{laws for stressing}$$

(the latter being in $\chi(O_{13})$) is minuscule because the belief that can be attributed to **laws for stressing** is minuscule. As we don't have the prior probabilities available, it is impossible to say by what factor the belief in **similarity ∘ laws for stressing** increased in relation to its prior probability. Note, however, that the non-zero beliefs associated with $\chi(O_{13})$ are only 4 times larger than the background value. This may mean that the increase in belief would not be significant enough for the Refinement Machine to deem document 13 as being relevant enough to return.

Documents 14 and 15 would not be detected as being potentially relevant as the beliefs in the descriptors of the characterizations of these objects are still zero after the request was propagated. The Refinement Machine restricted to *modus continens* requires that there be some overlap between the request and object characterization. The evidence propagation uses this overlap to increase the beliefs involved. A Refinement Machine that could bring *modus substituens* and *modus generans* to bear will engender better recall because these rules foster the introduction of new elements into an expression. These new elements promote recall at the possible expense of precision.

In the case of document 15, there is no similarity whatsoever between its characterization and $\chi(q_1)$. This demonstrates the inherent insufficiency of only using document titles as basis for object characterization. Apparently the relevance judgement here was based on the abstract associated with this document.

Query 2

The relevant documents for query 2 are 12, 14 and 15. The Refinement Machine attributed a belief of 0.5514 to one of the descriptors in characterization of document 12. This is encouraging. For the same reasons as for query 1, documents 14 and 15 would not be returned by the Refinement Machine. The descriptor **free ∘ mixing ∘ problems** (see $\chi(O_{11})$) has a minuscule non-zero belief due to this expression sharing the term **problems** with the query.

Query 3

The relevant documents for this query are 5 and 6. Potentially interesting documents after propagation of the request are O_5, O_6, O_{11}, O_{23} and O_{24}. The characterization of document 5 feature three expressions of non-zero belief because these characterizations contain the expression heat o conduction which has a maximal belief as it is strictly deducible from an expression in $\chi(q_3)$. Unfortunately, the machine is unable to infer that document 5 is about slabs, whereas it can trivially be proven that it is about a slab via

$$\text{conduction into double-layer} \circ \text{slab} \vdash_{\text{MC}} \text{slab}$$

Once again the restriction of the Refinement Machine to *modus continens* is shown. If the thesaural relation slabs $_{\text{ISA}}$ slab was available in the presence of *modus substituens* and *modus generans* then problems in composite o SLAB could be strictly deduced from problems in composite o SLABS. With this result,

$$\text{composite} \circ \text{slab} \vdash_{\text{MC}} \text{slab}$$

hence,

$$\text{composite} \circ \text{slab} \mathrel{\vdash\!\!\sim}_{\text{PI}} \text{double-layer} \circ \text{slab}$$

This leads the way to promoting recall.

Descriptors in $\chi(O_6), \chi(O_{11}), \chi(O_{23})$ and $\chi(O_{24})$ which have a non-zero belief value are due to these sharing the term heat with the query. In the case of $\chi(O_6)$, the belief in transient o heat o flow is an order of magnitude higher than the other beliefs due to the high prior probability of the term flow.

5.2.3 Overall Remarks regarding the Refinement Machine

On the basis of the above experiments it is not possible to produce a detailed analysis of the effectiveness of the Refinement machine. For such an analysis, experiments of a much larger scale are necessary. In particular such experiments would allow the tuning of the Refinement Machine due to the increased understanding of how much the difference between $Pr(i|q)$ and $Pr(i)$ should weigh in the relevance judgements. Furthermore, further research will shed light on the appropriateness of the quantification of the index expression belief network. In this research values for probability assessment functions associated with variables based on binary expressions were derived by a simple frequency analysis over all connectors, for example, 15% of the binary expressions involve an of connector. However, $\gamma_{\texttt{P OF R}}(\texttt{P OF R}|\texttt{P} \wedge \texttt{R}) = 0.15$ is a blatant underestimate because expressions such as pollution is rivers, pollution over rivers and pollution with rivers do not (normally) arise.

Nevertheless, the experiments do provide food for thought. A big question mark is whether the Refinement Machine is scalable to real life information disclosure applications. Even with the restricted lithoid as basis is took approximately one and half hours for IDEAL to build the join tree. Propagation of a request needed approximately forty minutes. The stark reality is that real life applications would be based on an index expression belief network containing thousands, if not hundreds of thousands, of vertices. Such networks are far greater in size than are currently being studied within artificial intelligence. Despite these magnitudes we do believe that an efficient Refinement Machine is realizable due to the following reasons:

- The IDEAL system is implemented in LISP which explains, in part, its sluggishness[3].
- Other slowness is largely due to the construction of the join tree. The big culprit here is the polynomial-in-the-height-of the-graph algorithm that detects and short cuts cyles of length 4 or more. It will be evident that in the restricted lithoid no such cycles exist, so this costly preprocessing of the belief network need not be performed.
- It is highly likely that an evidence propagation algorithm can be developed which operates on only the relevant part of the lithoid. The basis for determining this sub-network would be all descriptors i whose closure $\mathcal{K}(i)$ under strict inference overlaps with $\mathcal{K}(q)$. Furthermore, the topology and quantification of the sub-network can be determined automatically.

It seems that the Refinement Machine is a reasonably sensitive information disclosure mechanism. This precision is not only due to the expressiveness of the index expressions; context free plausible inference demonstrated a blunt disclosure mechanism defined on these expressions. It is the combination of the index expressions and belief networks which realizes a sensitive plausible inference mechanism. On the other hand, the larger experiment shows that the Refinement Machine restricted to *modus continens* does lack with regard to recall. An interesting avenue for further pursuit is to enhance the Refinement Machine with *modus generans* and a context sensitive *modus substituens* strict inference rules (see chapter 3, section 3.3). These rules allow the derivation of more useful characterizations, thereby potentially improving recall. By way of illustration, via context sensitive substitution it is possible to strictly deduce heat ∘ flow in SLAB from heat ∘ flow in MULTI-LAYER ∘ SLAB. This would allow an object about heat flow in multi-layer slabs to be characterized by the former expression. In the present setup the fact that the "heat flow" is in a "slab" is not directly represented.

Encouraging is the fact that the machine can attribute substantial belief to the following plausible inference. (See query 2, document 12).

$$\text{flight of high} \circ \text{speed} \circ \text{aircraft} \mid\!\sim \text{high} \circ \text{speed} \circ \text{flight}$$

This is a positive result because this belief is not simply based, for example, on the size of the overlap of the terms of these expressions, but is founded on probabilistic inference over a net of expressions whose topology reflects the mutual contextual relationships between the expressions. As a final remark; the experiments do demonstrate that is is possible to implement a logic-based disclosure machine which is based on a non-trivial, mathematically sound plausible inference mechanism. The use of belief networks as implementation vehicle for such machines should be investigated further.

[3] A system called HUGIN recently developed by the University of Aalberg, Denmark is implemented in C and is much faster

Chapter 6

Summary and Conclusions

This treatise is about improving the disclosure of information, an issue whose import need not be argued as information floods towards us in ever increasing tides. In a nut shell, the combination of the following is argued as promoting effective information disclosure:

- Index expressions as a more powerful characterization language (Chapter 2).
- The Information Disclosure Machine which is driven by logic-based principles (Chapter 3).
- A layered architecture which synthesizes the better aspects of hypermedia and traditional approaches to information retrieval (Chapter 4).

Improved object characterization via index expressions

In order to disclose information it is necessary to characterize the objects that carry the information. Information carriers, for example, documents, are typically characterized by keywords simply because efficient indexing algorithms exist which derive keywords automatically. This practical aspect has overweighed the fact that keyword-based characterizations form very incomplete reflections of the content of the associated information carrier. As a result, information disclosure mechanisms that use such characterizations can be expected to be ineffective. The development of a more powerful characterization language basically boils down to a trade off between the expressive power of the language against the cost of the indexing process.

This treatise features the language of index expressions as a characterization mechanism. An index expression is a structure in which not only keywords are modelled, but also the relationships between keywords. Such relationships provide a way of capturing the contextual framework in which the terms occur, thus opening the door to a more precise information disclosure mechanism. On the indexing side, it is shown that index expressions can be efficiently derived from the titles (or captions) of structural elements within an information carrier. This results in incomplete characterizations, but is partially offset by titles often being compact content-revealing reflections of the associated structural element. The restriction to titles is for pragmatic reasons; titles and captions are often in a form which permits a ready transformation to index expressions. The automatic derivation of index expressions from general text requires highly advanced parsing techniques; a problem which is beyond the scope of this dissertation.

Logic-based Information Disclosure Machines

Research in information disclosure seems to have reached a barrier; current empirically based mechanisms are not realizing significant improvements in information disclosure effectiveness. In order to force a breakthrough it is generally felt that document semantics should be brought into play. One school in information disclosure research advocates a logic-based approach in this regard.

The Information Disclosure Machine presented in this thesis embodies the logic-based information disclosure paradigm. A particular disclosure machine is dealt with at length - the Refinement Machine, whose inference mechanism is defined over the previously mentioned index expressions. As requests are often vague representations of the associated information need and objects characterizations are incomplete, being able to effectively reason with uncertainty is an important aspect of logic-based information disclosure. Index expression belief networks provide an elegant formalism into which both the strict and plausible inference rules can be embedded. Plausible reasoning with the Refinement Machine thereby becomes probablistic reasoning within a belief network. As a belief network adheres to the axioms of probability theory, it is free of the contradictions which result from quasi-probabilistic approaches to plausible reasoning. The Refinement Machine attempts to achieve good recall because it does not function according to a Closed World Assumption, while at the same time maintaining precision. The precision aspect is promoted by the expressiveness of the index expressions combined with probablistic inference within a network whose topology reflects mutual contextual relationships between expressions. Initial tests with a (restricted) Refinement Machine show it to be potentially precise though there is a long road ahead before it can be scaled up to handle real life applications.

Synthesizing Hypermedia and Information Retrieval

The information retrieval area was spawned in the libraries of the late fifties and has since provided several well founded disclosure mechanisms. A problem, however, with the information retrieval paradigm, is that the formulation of the information need is a difficult hurdle for the searcher. This often results in erroneous requests being fed to information disclosure mechanisms. One of the big advantages of hypermedia is that information disclosure is realized by browsing, which is a more natural form of searching behaviour. Hypermedia systems, however, typically offer little or no support for requests in the information retrieval sense. This can be irritating for the searcher with a clear cut information need. Furthermore, one of the characteristics of state-of-the-art hypermedia systems is the lack of conceptual description governing the associated databases. Some researchers applaud the resulting flexibility. We do not. Lack of conceptual description allows the degeneration of the database into an interweaved mess. This contributes to searcher disorientation commonly referred to as *lost in hyperspace.*

This dissertation presents a generalization of two level hypermedia into a stratified hypermedia containing any number of layers. The architecture aims at synthesizing the better aspects of information retrieval and hypermedia into an information disclosure framework of the next generation. The layers essentially allow the information to be considered at different levels of abstraction. Each layer in the stratified architecture is governed by a set of rules which fully prescribes the allowable hypermedia structures within that layer. The structure can not only be exploited for the purposes of information disclosure, but forms the bulwark for quality control within hypermedia applications. Just as functions and procedures modularize software, the notion of a view allows modularization of hypermedia within a layer. The stratified hypermedia supports information disclosure through both browsing and requests. The formulation of the information need into a request is supported by the notion of Query by Navigation within a special hyperindex layer: The searcher first browses through the hyperindex developing and concretizing his or her information need (that is, typical hypermedia interaction) and when satisfied, performs a *beam down* operation, the intention of which is to disclose relevant objects in the underlying hyperbase layer (that is, typical information retrieval). In short, Query by Navigation is a hypermedia notion cloaked as an operation which can be realized using techniques from information retrieval. With regard to information disclosure within two level hypermedia the following can be said:

- The results of the art history experiment (chapter 5) promote the use of lithoid-based hyperindices because they are constructed automatically and seem to be as effective as thesaurus-based hyperindices. Furthermore, searchers in an unfamilar subject area seem to be able to effectively search with lithoid-based hyperindices. A hyperindex combining the two appears to be a very effective tool for information disclosure.

- For implementing the *beam down* operation, we advocate the use of the logic based information disclosure paradigm. This approach is viewed by some researchers as being extremely hopeful in forcing a breakthrough in information disclosure effectiveness. Although the results documented in this treatise do not prove a breakthrough as having taken place, it does show that logic-based information disclosure is not only a theory but also implementable.

Bibliography

[AAC+89] M. Agosti, A. Archi, R Colotti, R.M. Di Giorgi, G. Gradenigo, B. Inghirami, P. Matiello, R. Nannuci, and M. Ragona. New prospectives in information retrieval techniques: a hypertext prototype in environmental law. In *Online Management 89, Proceedings 13th International Online Information Meeting, London, England*, pages 483–494, 1989.

[ACG91] M. Agosti, R. Colotti, and G. Gradenigo. A two-level hypertext retrieval model for legal data. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 316-325, 1991.

[BB91] R. Bosman and R. Bouwman. The Automation and Disclosure of a Slides Library. Master's thesis, University of Nijmegen, 1991.

[BBB91] R. Bosman, R. Bouwman, and P.D. Bruza. The Effectiveness of Navigable Information Disclosure Systems. In G.A.M. Kempen, editor, *Proceedings of the Informatiewetenschap 1991 conference*, pages 55–69, 1991.

[BC89] C. Berrut and Y. Chiaramella. Indexing Medical Reports in a Multimedia Environment: the RIME experimental approach. In *Proceedings of the Twelfth ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86, 1989.

[Bla90] D.C. Blair. *Language and Representation in Information Retrieval.* Elsevier, 1990.

[Bru90] P D. Bruza. Hyperindices: A Novel Aid for Searching in Hypermedia. In A.Rizk, N.Streitz, and J.Andre, editors, *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 109–122. Cambridge University Press, 1990.

[Bub86] J.A. Bubenko. Information system methodologies - a research view. In T.W. Olle, H.G. Sol, and A A. Verrijn Stuart, editors, *Information System Design Methodologies· Improving the Practice*, pages 289–318. North-Holland, 1986.

[BvdG93] P.D. Bruza and L.C. van der Gaag. Index Expression Belief Networks for Information Disclosure. *International Journal of Expert Systems*, 1993. (To appear).

[BvdW89] P.D. Bruza and T.P. van der Weide. The semantics of data flow diagrams. In *Proceedings of the International Conference on Management of Data*, pages 66–78, 1989. Hyderabad, India.

[BvdW90a] P.D. Bruza and T.P. van der Weide. Assessing the Quality of Hypertext Views. *ACM SIGIR FORUM (Refereed Section)*, 24(3):6–25, 1990.

[BvdW90b] P.D. Bruza and T.P. van der Weide. Two Level Hypermedia - An Improved Architecture for Hypertext. In A.M.Tjoa and R.Wagner, editors, *Proceedings of the Data Base and Expert System Applications Conference (DEXA 90)*, pages 76–83. Springer Verlag, 1990.

[BvdW91a] P.D. Bruza and T.P. van der Weide. Deducing Relevant Information using the Information Disclosure Machine. In A.J. van de Goor, editor, *Proceedings of the Computing Science in the Netherlands Conference (CSN91)*, pages 135–149, 1991.

[BvdW91b] P.D. Bruza and T.P. van der Weide. The Modelling and Retrieval of Documents using Index Expressions. *ACM SIGIR FORUM (Refereed Section)*, 25(2), 1991.

[BvdW92] P.D. Bruza and T.P. van der Weide. Stratified Hypermedia Structures for Information Disclosure. *The Computer Journal*, 35(3):208–220, 1992.

[BY89] R.A Baeza-Yates. Algorithms for String Searching: A Survey. *ACM SIGIR FORUM*, 23(3,4):34–58, 1989.

[CD90] H. Chen and V. Dhar. Online Query Refinement on Information Retrieval Systems: A Process Model of Searcher/System Interactions. In J. Vidick, editor, *Proceedings of the Thirteenth ACM SIGIR Conference*, pages 115–134, 1990.

[CGR87] I.R. Campbell-Grant and P.J. Robinson. An Introduction to ISO DIS 8613 - Office Document Architecture - and its Application to Computer Graphics. *Computer and Graphics*, 11(4):325–341, 1987.

[Cle91] C.W. Cleverdon. The Significance of the Cranfield Tests on Index Languages. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1991.

[CN90] Y. Chiaramella and J. Nie. A Retrieval Model based on an Extended Modal Logic and its Application to the RIME Experimental Approach. In *Proceedings of the 13th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–43, 1990.

[Coo71] W.S. Cooper. A Definition of Relevance for Information Retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.

[Coo90] G.F. Cooper. The computational complexity of probabalistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[Cra78] T.C. Craven. Linked phrase indexing. *Information Processing and Management*, 14(6):469–476, 1978.

[Cra86] T.C. Craven. *String Indexing*. Academic Press, Inc, 1986.

[Cra88] T.C. Craven. Adapting of string indexing systems for retrieval using proximity operators. *Information Processing and Management*, 24(2):133–140, 1988.

[CTL91] W. Bruce Croft, H.R. Turtle, and D.D. Lewis. The Use of Phrases and Structured Queries in Information Retrieval. In A. Bookstein, Y. Chiaramella, G. Salton, and V.V. Raghavan, editors, *Proceedings of the 14th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32-45, 1991.

[Far80a] J. Farradane. Relational Indexing Part I. *Journal of Information Science*, 1(5):267-276, 1980.

[Far80b] J. Farradane. Relational indexing part II. *Journal of Information Science*, 1(6):313–324, 1980.

[FC89] M.E. Frisse and S.B. Cousins. Information Retrieval from Hypertext: Update on the Dynamic Medical Handbook Project. In *Hypertext 89 Proceedings*, pages 199–212, 1989.

[Fox90] E.A. Fox. Virginia Disc One, 1990. Virginia Polytechnic Institute and State University, Viginia U.S.A.

[Fox90] C. Fox. A Stop List for General Text. *ACM SIGIR FORUM*, 24(1-2):19–35, 1989/90.

[Gar88] P. Garg. Abstraction mechanisms in hypertext. *Communications ACM*, 31(7):863–870, July 1988.

[GC90] L.S Gay and W. Bruce Croft. Interpreting Nominal Compunds for Information Retrieval. *Information Processing and Management*, 26(1):21–38, 1990.

[GGP89] R. Godin, J. Gecsei, and C. Pichet. Design of a Browsing Interface for Information Retrieval. In N.J. Belkin and C.J. van Rijsbergen, editors, *Proceedings of the Twelfth ACM SIGIR Conference*, pages 32–37, 1989.

[Gro91] R. Grotens. An Investigation to Hyperindex Navigation Aid. Master's thesis, University of Nijmegen, 1991.

[GT87] G. Gonnet and F. Tompa. Mind Your Grammar: a New Approach to Modelling Text. In *Proceedings of the Thirteenth VLDB Conference*, pages 339–346, 1987.

[GW] L.C. van der Gaag and M.L. Wessels. A Two Layered Belief Network for Control and Selective Gathering of Evidence. (Forthcoming).

[Hei91] P.J. Heise. Technieken voor ontsluiten van ongestructureerde data. Computable, September 1991. (In Dutch).

[ISO15] ISO8879. Information Processing - Text and Office Systems - Standard General Markup Language (SGML), 1986-10-15.

[Kee77] E. Michael Keen. On the generation and searching of entries in printed subject indexes. *Journal of Documentation*, 33(1), 1977.

[Kee78] E. Michael Keen. On the performance of nine printed subject index entry types. Technical report, Department of Information and Library Studies, The University College of Wales, Aberystwyth, 1978. Research report.

[Knu84] D.E. Knuth. *The TEXbook.* Addison Wesly, reading, Massachusetts, 1984.

[LG91] P.J.F. Lucas and L.C. van der Gaag. *Principles of Expert Systems.* Addison Wesley, 1991.

[LS88] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *The Journal of the Royal Statistical Society*, 50:157–224, 1988.

[Luc90] D. Lucarella. A Model for Hypertext-Based Information Retrieval. In *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 81–94. Cambridge University Press, 1990.

[Mar77] M.E. Maron. On Indexing, Retrieval and the Meaning of About. *Journal of the American Society for Information Science*, 28(1):38–43, 1977.

[Nea90] R.E. Neapolitan. *Probabalistic Reasoning in Expert Systems.* John Wiley & Sons, 1990.

[Nie86] J. Nie. An Outline of a General Model for Information Retrieval Systems. In *Proceedings of the Ninth ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 495–506, 1986.

[NvJ91] W. Nottroth and F. van Jole. Spoorzoeken in tekstbestanden. Computable, September 1991. (In Dutch).

[Pea88] J. Pearl. *Probabalistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman Publishers, Palo Alto, 1988.

[Por80] M.F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.

[Pou92] D. Pountain. Browsing Art the Windows Way. BYTE, April 1992.

[Rij86a] C.J. van Rijsbergen. A New Theoretical Framework for Information Retrieval. In *Proceedings of the 9th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 194–200, 1986.

[Rij86b] C.J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481–485, 1986.

[Rij89] C.J. van Rijsbergen. Towards an Information Logic. In *Proceedings of the Twelfth ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86, 1989.

[Ros91] A. Rosing. An Evaluation of the Hyperindex Machine. Master's thesis, University of Nijmegen, 1991.

[Sal83] G. Salton. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, 1983.

[Sal89] G. Salton. *Automatic Text Processing: The Translation, Analysis and Retrieval of Information by Computer.* Addison-Wesley Publishing Company, 1989.

[SB90] S. Srinivas and J. Breese. IDEAL: Influence Diagram Evaluation and Analysis in Lisp, Documentation and Users Guide. Technical Memorandum no. 23, 1990. Rockwell International Science Center, Palo Alto.

[Sch89] H. Schouten. SGML*CASE: The Storage of Documents in Databases. Technical Report 03-11, TFDL/ECIT, PO 356, 6700 AL Wageningen, The Netherlands, 1989. Internal Report.

[SD86] G.W. Strong and M. Carl Drott. A thesaurus for end-user indexing and retrieval. *Information Processing & Management*, 22(6), 1986.

[SDBvdW91] P.L. van der Spiegel, J.T.W. Driessen, P.D. Bruza, and T.P. van der Weide. A Transaction Model for Hypertext. In *Proceedings of the Data Base and Expert System Applications Conference (DEXA 91)*, pages 281–286. Springer Verlag, 1991.

[SF89] P. Stotts and R. Furuta. Petri-Net-Based Hypertext: Document Structure with Browsing Semantics. *ACM Transactions on Information Systems*, 7(1):3–29, 1989.

[Sha85] D. Shasha. Netbook- a data model to support knowledge exploration. In *Proceedings of the Eleventh International Conference on Very Large Data Bases*, pages 418–425, 1985.

[Sie91] P. Siebers. Je vraagt ook geen patent aan op alfabetische volgorde van de telefoongids. *K.U. Nieuws (University of Nijmegen)*, 21(11):7–7, 1991. (In Dutch).

[Sme90] A.F. Smeaton. Indexing and Text Representation. In *Proceedings of the European Summer School in Information Retrieval*, pages 171–235, 1990.

[Sta90] H. Staveleu. De kortste weg naar elk woord is maar drie letters lang. KIJK, November 1990. (In Dutch).

[SvR90] T.M.T. Sembok and C.J. van Rijsbergen. SILOL: A Simple Logical-Linguistic Document Retrieval System. *Information Processing and Management*, 26(1):111–134, 1990.

[TC90] H.R. Turtle and W. Bruce Croft. Inference Networks for Document Retrieval. In J.L. Vidick, editor, *Proceedings of the 13th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, 1990.

[Tom89] F. Tompa. A Data Model for Flexible Hypertext Database Systems. *ACM Transactions on Information Systems*, 7(1):85–100, January 1989.

[TS88] B. Teufel and S. Schmidt. Full text retrieval based on syntactic similarities. *Information Systems*, 13(1):65–70, 1988.

[TSM91] J. Tague, A. Salminen, and C. McClellan. A Complete Model for Information Retrieval Systems. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 14–20, 1991.

[vdG90] L.C. van der Gaag. *Probability-Based Models for Plausible Reasoning.* PhD thesis, University of Amsterdam, 1990.

[vdW85] H. van de Waal. An Iconclass Classification System, 1985. Completed and edited by L.D.Couprie, E.Tolen and G.Vellenkoop.

[WD91] E.B. Wendlandt and J.R. Driscoll. Incorporating a Semantic Analysis into a Document Retrieval Strategy. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 270–279, 1991.

[Wea88] M.T. Weaver. A frame-based language in information retrieval. Technical Report TR88-25, Virginia Polytechnic, 1988.

[WF89] M.F. Wyle and H.P. Frei. Retrieving Highly Dynamic, Widely Distributed Information. In *Proceedings of the 12th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 108–115, 1989.

[WY90] S.K.M. Wong and Y.Y. Yao. A probabilistic inference model for information retrieval. *Information Systems*, 15(3):301–321, 1990.

Index

Author Index

Samenvatting

Om ons heen blijven de stapels informatiedragers onophoudelijk groeien. In deze massa wordt het steeds meer een probleem de gewenste informatie snel en doeltreffend te vinden.

In een bibliotheek gaat het zoekproces bijvoorbeeld als volgt: de informatiebehoefte wordt gespecificeerd in de vorm van een vraag die uit een kleine verzameling van trefwoorden bestaat. Ook de inhoud van een informatiedrager wordt gekarakteriseerd door middel van trefwoorden. Het ontsluitingssysteem definieert een informatiedrager als zijnde relevant zodra er een sterke overeenkomst bestaat tussen de vraag en de karakterisering van de bijbehorende informatiedrager. Zoeken op deze wijze kent echter twee beperkingen. Ten eerste: het blijkt moeilijk te zijn door middel van een concrete vraagstelling de informatiebehoefte precies te omschrijven. Ten tweede is de effectiviteit van dergelijke ontsluitingssystemen vaak teleurstellend omdat de karakteriseringen slechts een zwakke weerspiegeling zijn van de informatie, die in de drager opgesloten zit. Al veertig jaar vindt onderzoek plaats om, binnen deze beperkingen, de meest effectieve ontsluitingssystemen te realiseren. Verdere verbeteringen van dergelijke systemen lijken inmiddels uitgesloten.

Een andere, meer natuurlijke wijze van zoeken is het *browsen* door de informatie. Om deze manier van ontsluiting mogelijk te maken, worden de informatiedragers 'in stukken gehakt' en voorzien van velerlei dwarsverbanden. Het resultaat is een netwerk waarin de gebruiker zijn eigen pad kan kiezen. Een dergelijk netwerk vormt het fundament van de ontsluiting in zogenaamde hypermedia systemen. Het voordeel van deze ontsluiting is, dat de gebruiker nooit zijn informatiebehoefte hoeft te specificeren. Het nadeel is dat hij of zij 'de weg kwijt kan raken': een probleem dat recht evenredig toeneemt met de omvang van het netwerk. Bovendien gaat de oorspronkelijk structuur van de informatiedragers verloren doordat zij in stukken gehakt worden.

Dit proefschrift presenteert een algemene architectuur (*stratified architecture*) voor het ontsluiten van informatie. Op drie manieren poogt deze architectuur effectieve ontsluiting te bevorderen:

1. Het gebruik van Index Expressies als krachtig karakteriseringsmechanisme.

2. De Informatie Ontsluitingsmachine die volgens logische principes functioneert.

3. Synthese van de voordelen van beide, hierboven geschetste ontsluitingsmethoden.

De *stratified architecture* bestaat uit lagen waarmee de informatie door middel van zogenaamde *views* gemodulariseerd kan worden. Views bevatten structuren waarin en waartussen een gebruiker kan browsen. Zij moeten bovendien voldoen aan een conceptuele specificatie, die de structurele integriteit van de informatie waarborgt.

De gebruiker kan ook in zijn informatie behoefte voorzien door tussen de verschillende lagen te navigeren. Eén laag, de zogenaamde *hyperindex*, biedt de gebruiker *query by navigation*, dat wil zeggen: de mogelijkheid om te browsen naar beschrijving(en) van de informatie behoefte. Deze beschrijvingen dienen als vraag voor de Informatie Ontsluitingsmachine.

Deze machine heeft als doel om de relevante objecten in de onderliggende laag (de *hyperbase*) te vinden. Hij functioneert volgens een op de logica gebaseerd ontsluitingsparadigma, waarbij de relevantie van een informatie-object alleen maar vastgesteld wordt, als de vraag bewezen kan worden uit de karakteriseringen (axiomas) van het desbetreffende object. Als dit niet lukt, wordt een probabiliteit van relevantie van het object berekend door 'onzekere inferentie' toe te passen.

In dit proefschrift wordt de *Refinement Machine* geïntroduceerd. Zij is een specifiek voorbeeld van een informatie-ontsluitingsmachine. De Refinement Machine gebruikt een logica over de index expressies om de relevantie van een informatie-object te bewijzen. Wanneer een dergelijk bewijs niet lukt kan de machine onzekere inferenties over de mogelijke relevantie maken door probalistische uitspraken te putten uit een onderliggend *belief network* opgebouwd uit index expressie karakteriseringen. De werking van de Refinement Machine en de effectiviteit van de *stratified architecture* worden toegelicht via resultaten van enkele concrete experimenten.

Curriculum Vitae

The author was born on the fifth of january, 1962 in Brisbane, Australia, where he attended both school and university. On graduation from the University of Queensland, in 1982, with a Bachelor of Science degree (major in computer science), the author worked for a number of years as a database systems analyst and developer. At the beginning of 1987, he re-entered the university world at the Department of Information Systems, University of Nijmegen, The Netherlands. In march of 1989 he graduated as *Doctorandus* (cum laude) and thereafter assumed a position as researcher in the ESPRIT II project *APPED*. On completion of the project in february 1992, the author became a lecturer at the Department of Information Systems, University of Nijmegen. Since august 1992, the author holds the same position at the

Department of Computer Science
University of Utrecht
P.O. Box 80.089
3508 TB Utrecht
The Netherlands
peterb@cs.ruu.nl

Stellingen horende bij het proefschrift
STRATIFIED INFORMATION DISCLOSURE
van

Peter D. Bruza

1. For practical reasons, the characterization of an information object is typically incomplete. Furthermore, searchers do not always know exactly what they are looking for. Therefore, in order to be effective, an information disclosure mechanism must have the capability to reason with uncertainty.

2. A complete and sound Information Disclosure Machine is insufficient to solve the information retrieval problem.

3. It is important that a framework be developed within which the effectiveness of Information Disclosure Machines can be compared *inductively*, instead of experimentally.

4. The *No Blind Faith* theorem (Chapter 3) implies that the probability of an index expression I can be determined via probabilistic inference involving only the theorems derivable from I.

5. In order to make Index Expression Belief Networks feasible for driving plausible inference in real-life information disclosure applications, much faster evidence propagation algorithms need to be developed.

6. The divisions between hypermedia, (structured) document and traditional databases can, and should be overcome.

7. Lithoid-based hyperindices are at least as effective as thesaurus-based ones and, furthermore, lend themselves better for use by non-expert searchers (Chapter 5).

8. A formalism can be a two edged sword; when wielded appropriately it yields lucidity and conciseness. It can, however, deceive its exponent to lose touch with reality, at which point the formalism has the potential to beget itself.

9. *The Dutch Theorems:*

   (a) Many Dutch have large hands.

   (b) The notion of "going out for lunch" never really took off in The Netherlands as most Dutch take their lunch with them.

   (c) *De Ziektewet* is sometimes the rubbish bin of personnel departments.

   (d) No meeting, gathering or party in The Netherlands can begin without coffee.

   (e) The variation of the wares in a Dutch bakery is proportional to the softness of the baker's "g".

   (f) The failures of the Dutch soccer team are related to the negativity and extreme criticism of the Dutch press.

   (g) When the antipathy within the Dutch collective unconscious towards Germans has been dealt with, a step will have been made to a true united Europe.