# Retrieval and the Semantic Web

Alistair Miles (04137388)

Dissertation Report – Web Technologies MSc

Oxford Brookes University

Submission Date 1st September 2006

## Abstract

A primary motivation for the development of the Semantic Web has been the need for effective information retrieval systems which may be realised through vocabulary control and the use of structured metadata. The technological framework of the Web (URI, HTTP, XML) and of the Semantic Web (RDF, OWL, SPARQL) provides a platform upon which distributed data and metadata applications may be constructed, but does not in itself provide any direct support for information retrieval applications *per se*. Widely applicable Semantic Web languages that extend this basic layer and provide generic support for retrieval applications, in addition to good practice guidelines and design patterns for developing such applications, are required.

The ultimate purpose of this report is to develop a formal theory of retrieval using controlled vocabularies that have a simple and intuitive structure, to provide the necessary theoretical foundations for the development of Semantic Web languages and design patterns for distributed retrieval applications. The main body of this report is devoted to the articulation of such a theory. The theory is expressed formally through the use of mathematical notation, with the intention that this level of formality will provide the bridge between informal requirements specifications and the implementation of effective retrieval applications in computer systems.

Specifically, a theory is developed to describe the ways in which a structured vocabulary may be used to construct an index over a collection of objects and then used to express queries which may be evaluated against an index to obtain a set of results. This theory is extended to consider ways in which both the precision and recall of retrieval strategies may be improved, through the use of expansion and ranking techniques and through “coordination”. The problem of translating between controlled vocabularies is also

considered. The theory attempts to formalise, unify and extend the traditional wisdom of the library sciences regarding the use of thesauri, classification schemes, subject heading systems, taxonomies and other types of structured vocabulary, so that proven techniques and methodologies may be transferred to a Semantic Web context.

The recently chartered W3C Semantic Web Deployment Working Group has been charged with the development of the Simple Knowledge Organisation System (SKOS) to W3C Recommendation status. SKOS is a Semantic Web language specifically intended to support information retrieval applications using controlled vocabularies that have a relatively simple structure. A formal requirements specification is the first planned deliverable in the standardisation of SKOS. An immediate goal of this report is to provide a level of abstraction that can be used to perform a comparative analysis of use cases involving information retrieval systems that operate with structured vocabularies, so that the requirements of these systems with respect to Semantic Web languages such as SKOS may be clearly determined. Also, this report suggests ways in which the theory may be mapped to concrete language constructs and representation patterns in Semantic Web languages. In so doing it is hoped that the development of SKOS and similar languages may be grounded with sufficient rigour to ensure their wide applicability and consistent use.

# Table of Contents

# 1 Introduction

This report provides a theoretical foundation for the use of Semantic Web technologies for retrieval. The theory is intended to bridge the gap between the flexibility and logical power of RDF(S) [MAN04] [BRI04], OWL [SMI04] [PAT04] and SPARQL [PRU06] and the implementation of effective and usable retrieval systems.

## 1.1 Why Are Semantic Web Technologies Relevant to Retrieval?

With the maturation of the W3C's Semantic Web initiative, there has been renewed interest in the application of vocabulary control to the problems of information glut faced within organisations and on the Web. The interest is probably justified not only because the basic Semantic Web technology stack (URI [RFC05], XML [BRA06], RDF [KLY04], RDFS [BRI04], OWL [SMI04], SPARQL [PRU06]) provides powerful standardized tools for expressing and sharing vocabularies, but also because the possibility of harnessing network effects at the metadata level may re-balance the cost/benefit trade-offs traditionally associated with retrieval solutions based on vocabulary control.

Having said that, this report does not aim in any way to either prove or disprove the contention that, through the application of vocabulary control, retrieval systems can be provided whose functionality and performance is superior to that of retrieval systems based on text retrieval techniques or other methods, or that any perceivable benefit is sufficient to justify the cost of manual creation and curation of vocabularies and indexes. Rather, it aims to provide a mathematical foundation for the use of structured vocabularies in retrieval systems and a mapping from this foundation to concrete representations in Semantic Web languages, in order that the value of a particular vocabulary may be maximised in the context of a particular retrieval application.

A specific aim of this report is to support the development of standardized RDF languages and representation patterns for various aspects of retrieval data, including structured vocabularies, indexes and vocabulary mappings. It is hoped that the theoretical framework developed here may be used to perform a comparative analysis of use cases for retrieval systems and so to extract a set of common requirements that can be taken as input to current standardisation initiatives. In particular, the proposed development of the Simple Knowledge Organisation System (SKOS) specifications [MIL05A] [MIL05B] towards W3C Recommendation status, as planned within the scope of the recently chartered Semantic Web Deployment Working Group, is a primary motivation. So too is the continued development of the Dublin Core Metadata Initiative's (DCMI's) core metadata standards, including the DCMI metadata terms and the DCMI abstract model [POW05].

## 1.2 Introduction to the Theory Chapters

The main body of this report is devoted to the formal specification of a theory of retrieval using controlled vocabularies. The theory takes as its starting point a type of controlled vocabulary assumed to have a very simple structure, and derives a mathematical account of the retrieval operations that such vocabularies may reasonably support. Although the theory may easily be extended for vocabularies with more intricate structure, it is notable both how much complexity and how much functionality can be derived from the simple structures considered.

In short, the theory describes how a structured vocabulary may be used to create an index for a collection of documents and how queries constructed using the same vocabulary may be evaluated against an index to produce a list of results. Note that although for convenience I refer throughout this report to the objects of retrieval as "documents", the theory is entirely ambivalent towards the type of object being retrieved.

Each chapter of the theory is accompanied by one or more use cases, that illustrate how the theory may be used to provide a classification of the functionality of a controlled vocabulary retrieval system. Once the functionality of a retrieval system has been classified in this way, it may be directly compared with that of other systems

The first chapter of the theory (Chapter 3. “Foundations”) defines the structure of a vocabulary and the structure of an index. Chapter 3 also shows how simple "atomic" queries can be evaluated against an index to produce a set of results and how the structure of a vocabulary can be used to "expand" an index in a naive way to improve recall.

The second chapter of the theory (Chapter 4. “Composite Queries”) extends the foundations to define further types of query expression that can be used to combine atomic expressions into "composite" queries. Chapter 4 shows how composite queries may be evaluated, how they may be broken down into their "atoms", how they may be naively expanded and how the results of a composite query may be given a numeric score and so ranked in order of relevance to the query.

The third chapter of the theory (Chapter 5. “Limited Cost Expansions”) focuses on techniques for the expansion of both indexes and queries that make more sophisticated use of the structure of the vocabulary than the naive expansions considered in previous chapters. Chapter 5 shows how numeric weights derived from an expansion of an index or a query by this method can be factored into the scoring of results and therefore how recall can be improved without sacrificing perceived precision.

The fourth chapter of the theory (Chapter 6. “Coordination”) extends previous definitions to consider advanced uses of a structured vocabulary where vocabulary units are combined ("coordinated") in indexes and in queries to express more specific meanings. Both naive and limited cost expansions are defined for coordinated indexes and queries, so enabling precision to be gained without compromising recall.

The final chapter of the theory (Chapter 7. “Translation”) applies the theory developed so far to the problem of expressing a mapping between two vocabularies and exploiting such a mapping to obtain reasonable translations of either indexes or queries with minimal loss of precision and recall. Two alternative types of mapping are defined, and chapter 7 shows how either type of mapping may be used to translate a query or an index.

Note that I have chosen to present the theory without any reference to the literature from which the fundamental ideas or inspiration may have been drawn. I have done this in order that the theory itself may presented in a most clear and concise manner, with consistent use of terminology and without the confusion that may sometimes arise due to a discussion of differences in terminology or notation between similar works. Instead, a discussion of the relationships between the theory presented here and previous work is given in chapter 9.

## 1.3 Introduction to the Mapping to Semantic Web Languages

Once the theory has been established, possible strategies for mapping both in and out of representations in Semantic Web languages are presented in chapter 8 ("RDF Representations"). Options are discussed for creating representations of a structured vocabulary and an index using RDF and for the derivation of these data structures from existing RDF graphs. In particular, the consequences of the theory for the operational semantics of the Simple Knowledge Organisation System (SKOS) vocabularies are considered and concrete suggestions are made for forming operational definitions grounded in intended consequences in retrieval systems with respect to appropriate assumptions about relevance.

# 2 Mathematical Notation

This chapter briefly introduces the mathematical notation used throughout this report. The notation is consistent with the Z notation given in [SPI89], including the use of schemas to group a set of mathematical definitions.

## 2.1 Expressions

The expression $(x_1, \ldots, x_n)$ denotes an n-tuple whose components are the objects $x_1, \ldots, x_n$.

The expression $\{x_1, \ldots, x_n\}$ denotes the set whose only members are the objects $x_1, \ldots, x_n$.

The expression $\mathbb{P}\ S$ denotes the power set of $S$. I.e. The set of all subsets of $S$.

The expression $S_1 \times \ldots \times S_n$ denotes the Cartesian product of the sets $S_1, \ldots, S_n$. I.e. The set of all n-tuples $(x_1, \ldots, x_n)$ where $x_i \in S_i$ for each $i$ with $1 \leq i \leq n$.

The expression $\{D \mid P \bullet E\}$ denotes a set comprehension. I.e. The set whose members are the values of the expression $E$ when the variables introduced by $D$ take all possible values which make $P$ true.

## 2.2 Predicates

The symbols = and ∈ denote equality and set membership respectively and the symbols ≠ and ∉ denote their respective complements.

The symbols ¬, ∧, ∨, ⇒ and ⇔ denote the standard connectives of propositional logic.

The symbols ∀, ∃ and $\exists_1$ denote the standard quantifiers of predicate logic.

## 2.3 Sets

The symbols ⊆ and ⊂ denote the subset and proper subset relations respectively.

The symbol ∅ denotes the empty set.

The symbols ∩, ∪ and \ denote the ordinary set intersection, set union and set difference operators of set algebra respectively.

The expressions $\cap A$ and $\cup A$ denote respectively the generalized union and the generalised intersection of a set of sets $A$.

The names *first* and *second* denote the projection functions for ordered pairs, i.e. $first(x, y) = x$ and $second(x, y) = y$.

## 2.4 Relations

The expression $x \mapsto y$ denotes the ordered pair $(x, y)$.

The expression $X \leftrightarrow Y$ denotes the set of all binary relations between $X$ and $Y$.

The expression $R : X \leftrightarrow Y$ declares that $R$ is a binary relation between $X$ and $Y$.

For any binary relation $R$, *dom* $R$ denotes the domain of $R$ and *ran* $R$ denotes the range of $R$.

For any binary relation $R$, $R^{\sim}$ denotes the inverse relation of $R$ and $R^{+}$ denotes the transitive closure of $R$.

The expression $R(\!| S |\!)$ denotes the relational image of the set $S$ through the relation $R$.

The expression $P \mathbin{⨾} R$ denotes the forward relational composition of the relations $P$ and $R$ and the expression $P \circ R$ denotes the backward relational composition.

## 2.5 Functions

The expression $X \rightarrow Y$ denotes the set of total functions from $X$ to $Y$.

The expression $f : X \rightarrow Y$ declares that $f$ is a total function from $X$ to $Y$.

The expression $f\ x$ or $f\,(x)$ denotes the application of function f to x. Application associates to the left, so $f\ x\ y$ means $(f\ x)\,y$ . I.e. When $f$ is applied to $x$ the result is a function, which is applied to $y$. Note that parentheses may be placed around any expression.

## 2.6 Numbers

The symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$ denote the set of natural numbers, the set of integers and the set of real numbers respectively.

The symbols +, -, * and ÷ denote the standard arithmetic operations of addition, subtraction, multiplication and division respectively.

The symbols <, >, ≤ and ≥ denote the standard numerical comparisons.

The expression $R^k$ denotes the $k^{\text{th}}$ iteration of the relation $R$.

The expression $a\,..b$ denotes an integer number range. I.e. the set of integers between $a$ and $b$ inclusive.

The expression $\# S$ denotes the number of members of a set $S$.

The expressions *min* $S$ and *max* $S$ denote respectively the minimum and maximum of a set of numbers $S$.

The expression $\sum S$ denotes the sum of a set of numbers $S$.

## 2.7 Sequences

The expression $\langle a_1, \ldots, a_n \rangle$ is a shorthand for the set $\{1 \mapsto a_1, \ldots, n \mapsto a_n\}$ .

The expression $seq\ X$ denotes the set of all finite sequences over *X*.

The expression $\langle\,\rangle$ denotes the empty sequence.

The expression $disjoint \langle S_1, \ldots, S_n \rangle$ states that the sets $S_1$, ..., $S_n$ are all pairwise disjoint.

## 2.8 Schemas

A set of mathematical statements may be grouped in a schema. The first row of a schema introduces a set of variables (the signature) and the second row contains propositions held to be true regarding those variables. A Schema may be open (“axiomatic” - shown without top or bottom borders) in which case the schema introduces variables and propositions whose scope is global to this report, or closed (shown with top and bottom borders with the schema name embedded in the top border) in which case the schema defines a named schema type whose variables are local to the schema. A named schema may be included in another schema by placing the name of the included schema (a schema reference) in the signature of the including schema. The effect is to combine both signatures and both sets of propositions. Schemas may also be “generic” (shown with double line top border with generic parameters inset) in which case variables and propositions are global. The use of schemas in this report is entirely consistent with the Z notation [SPI89]. An example named schema is shown below.

Schema1

$X : \mathbb{P}\,\mathbb{Z}$

$Y : \mathbb{P}\,\mathbb{Z}$

$z : \mathbb{Z}$

---

$X = \{1, 4, 7\}$

$Y = \{3, 6, 8\}$

$z = \#X + \#Y$

# 3 A Theory of Retrieval Using Structured Vocabularies: Foundations

This chapter develops the foundations of a theory of retrieval using structured vocabularies.

## 3.1 Basic Types

The following four basic types (given sets) are given as the basis of the theory.

$[CNAME, DNAME, FNAME, QUERY]$

The first of the basic types *CNAME* is the set of all "concept names". I assume that a structured vocabulary provides a set of "names" with which to index documents and with which to build queries – also known as an "indexing language". The fundamental purpose of a structured vocabulary is to establish a set of distinct meanings or "concepts" and to provide some means of referring unambiguously to these concepts via a set of identifiers or names, hence I have chosen to call this set of names "concept names". Note that the theory I present here does not rely on an exploration of what these names might *denote*, it simply assumes a set of names and derives a set of retrieval operations based on the use of these names.

The second of the basic types *DNAME* is the set of all "document names". As mentioned in the introduction, I refer through this report to the objects we are interested in retrieving as "documents", although this is purely for convenience and the theory is entirely ambivalent as to the nature of the objects being retrieved. Therefore the set of "document names" can be thought of simply as a set of identifiers for the objects being retrieved and as the basic constituents of a result set.

The third of the basic types *FNAME* is the set of all "field names". Below I develop a model of an "index" as consisting of a set of "fields" to account for the general case where an index may have "multiple fields", although in many of the common use cases an index will consist of only a single field. Where an index consists of multiple fields there must be some means by which the separate fields can be referenced from within queries and hence "field names" are introduced as this means of reference.

The fourth and final basic type *QUERY* is the set of all "query expressions", where a query expression is a statement of need expressed in terms of one or more controlled vocabularies. This chapter introduces the notion of an "atomic query" as the simplest form of query expression; subsequent chapters extend this notion to include more complicated forms of query where query expressions may be nested within other query expressions.

## 3.2 Structured Vocabularies

As mentioned above I assume that the fundamental purpose of a structured vocabulary is to establish a set of distinct meanings or “concepts” and to provide some means of referring unambiguously to those meanings, which I have called “concept names”. Therefore the basic component of a structured vocabulary is a set of concept names which can be applied as an indexing language.

In addition to a set of concept names a structured vocabulary may provide one or more binary relations on this set that describe the structure of the vocabulary, which I call “structure relations”. Here I consider a particular type of structured vocabulary where three structure relations are provided, which I call the “broadening relation”, the “narrowing relation” and the “associating relation”. Together the broadening and narrowing relations define one or more “hierarchies” (i.e. one or more trees). A structured vocabulary is “monohierarchical” if the broadening relation is functional and “polyhierarchical” if it is not.

Note that no attempt is made here to provide an independent definition of the meaning of these structure relations. Instead, their meaning is defined entirely in terms of the retrieval behaviours that they may reasonably be used to support, via assumptions about the implications of these relations for relevance and hence precision and recall.

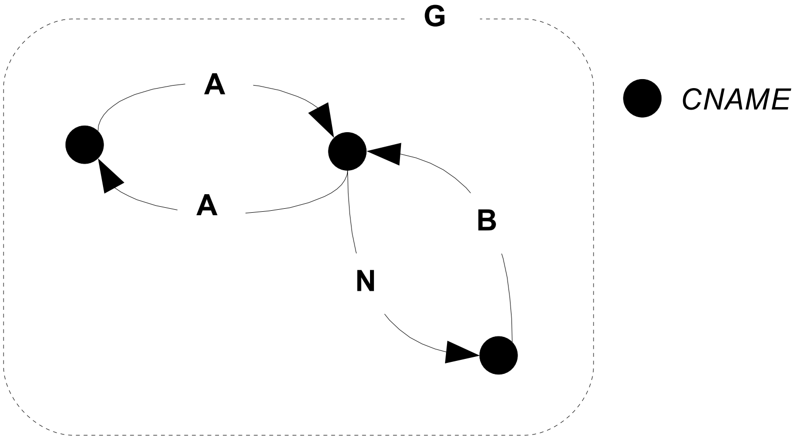
The schema below formally defines the *StructuredVocabulary* schema type.

$$\begin{array}{|l}
\text{StructuredVocabulary} \\
\hline
T : \mathbb{P}\, CNAME \\
broader : T \leftrightarrow T \\
narrower : T \leftrightarrow T \\
associated : T \leftrightarrow T \\
G : T \leftrightarrow T \\
\hline
G = broader \cup narrower \cup associated \\
broader = narrower^{\sim} \\
associated \text{ is } symmetric \\
broader^{+} \text{ is } irreflexive \\
disjoint \langle broader^{+}, narrower^{+}, associated \rangle \\
\hline
\end{array}$$

Note that some logical constraints are placed on the structure relations. Firstly it is assumed that the broadening and narrowing relations define a pair of inverse relations. Secondly it is assumed that the associating relation is symmetric. Thirdly it is assumed that the broadening relation defines a graph with no cycles, which is equivalent to the proposition that the transitive closure of this relation is irreflexive. Finally it is assumed that the associating relation, the transitive closure of the broadening relation and the transitive closure of the narrowing relation are pairwise disjoint.

The relation *G* is defined to be the union of the three structure relations and I refer to this relation as the “structure graph” or simply the “structure” of a structured vocabulary.

*Figure 1. Visualisation of the structure graph of a structured vocabulary.*



The figure above provides a visualisation of the structure graph of a controlled vocabulary for three concept names. Each arc labeled "A" represents a member of the associating relation, each arc labeled "B" represents a member of the broadening relation and each arc labeled "N" represents a member of the narrowing relation.

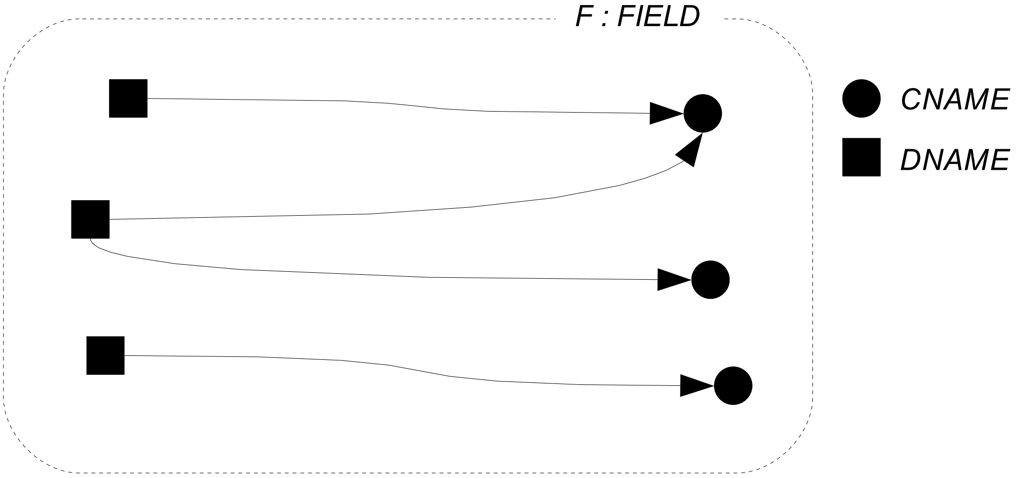
## 3.3 The Structure of an Index

As mentioned above I model an index as being comprised of a set of "fields". I define a "field" as a relation between a set of document names and a set of concept names. Because I will be reusing this notion often throughout the theoretical chapters I define a *FIELD* abbreviation below.

$$FIELD == DNAME \leftrightarrow CNAME$$

Figure 2. Above below a visualisation of the structure of a field, where each arc represents a member of the field.

*Figure 2. Visualisation of the structure of a field.*



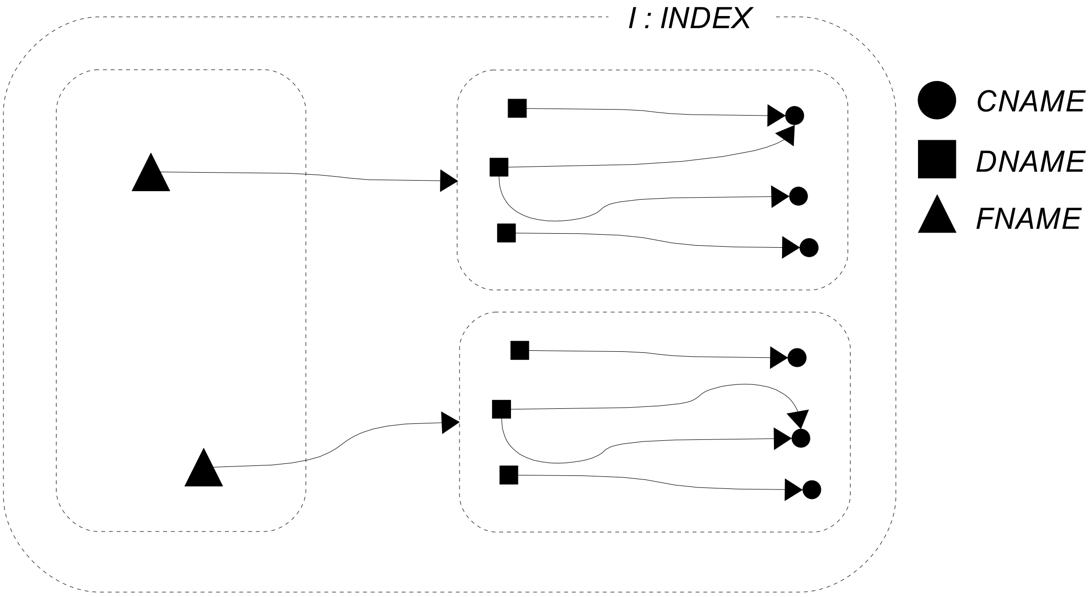
I define a "functional" field as being a field that satisfies the criterion of a function – all document names are mapped to one and only one concept name. A “relational” field I define as being a field that does not fulfill this criterion – all document names are mapped to one or more concept names.

I define an index as a mapping from a set of “field names” to a set of “fields”. Again because this notion is reused often I define an *INDEX* abbreviation below.

$$INDEX == FNAME \rightarrow FIELD$$

Figure 3 below shows a visualisation of an index with two fields.

*Figure 3. Visualisation of an index with two fields.*



## 3.4 Index Functions

Here I define several useful functions for operating on indexes. The *alldocs* function obtains the set of all documents indexed by a particular index and is defined below.

$$alldocs : INDEX \rightarrow \mathbb{P}\ DNAME$$

$$\begin{array}{l} \forall\, I : INDEX \bullet \\ \quad alldocs\ I = \bigcup \{ F : FIELD \mid F \in (ran\ I) \bullet dom\ F \} \end{array}$$

Note that for a field *F* the expression *dom F* gives the domain of the field, which is the set of documents indexed by that field. For an index *I* the expression *ran I* gives the range of the index, which is the set of fields which comprise the index. The definition of the *alldocs* function above simply gives the union of the domains of all the fields of an index, which is the set of all documents indexed by that index. Note also that it is expected that in most use cases all documents will be indexed in all fields and hence the domain of each of the fields will be the same set of document names. However this theory does not require this to be the case – different documents may be indexed in different fields.

The most common operations needed for a particular field are to obtain the set of all concept names indexing a particular document and to obtain the set of all documents indexed by a particular concept name. I refer to these functions as the forward index function (*tags*) and the inverted index function (*hits*) respectively. I have named these functions *tags* and *hits* in an attempt to provide a convenient notation that is somewhat suggestive of the meanings of the functions especially with regard to the evaluation of queries (see below). I have also chosen to use a superscript notation in an attempt to make subsequent definitions more compact. The *tags* and *hits* functions are defined below for a generic field.

$$
\begin{array}{|l}
\_^{tags} : FIELD \rightarrow ( DNAME \rightarrow \mathbb{P}\, CNAME ) \\
\_^{hits} : FIELD \rightarrow ( CNAME \rightarrow \mathbb{P}\, DNAME ) \\
\hline
\forall\, F : FIELD ; \forall\, d : DNAME ; \forall\, t : CNAME \bullet \\
\quad F^{tags}\, d = F\,(\!| \{ d \} |\!) = \{ t : CNAME \mid d \mapsto t \in F \} \land \\
\quad F^{hits}\, t = F^{\sim}\,(\!| \{ t \} |\!) = \{ d : DNAME \mid d \mapsto t \in F \}
\end{array}
$$

Note that the forward index function applied to an object *d* is equivalent to the relational image of the set {*d*} and the inverted index function applied to an object *t* is equivalent to the inverted relational image of the set {*t*}.

*Figure 4. Visualisation of the forward index function for a field F.*

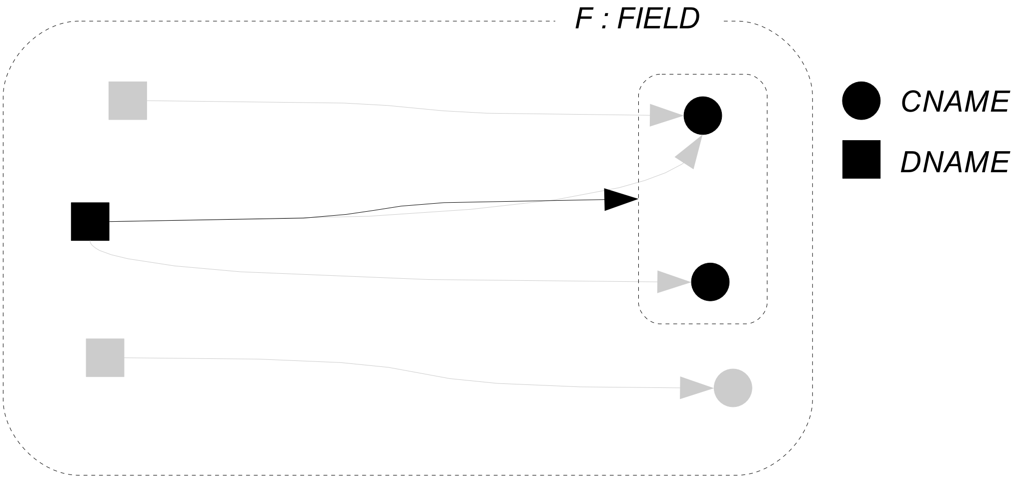
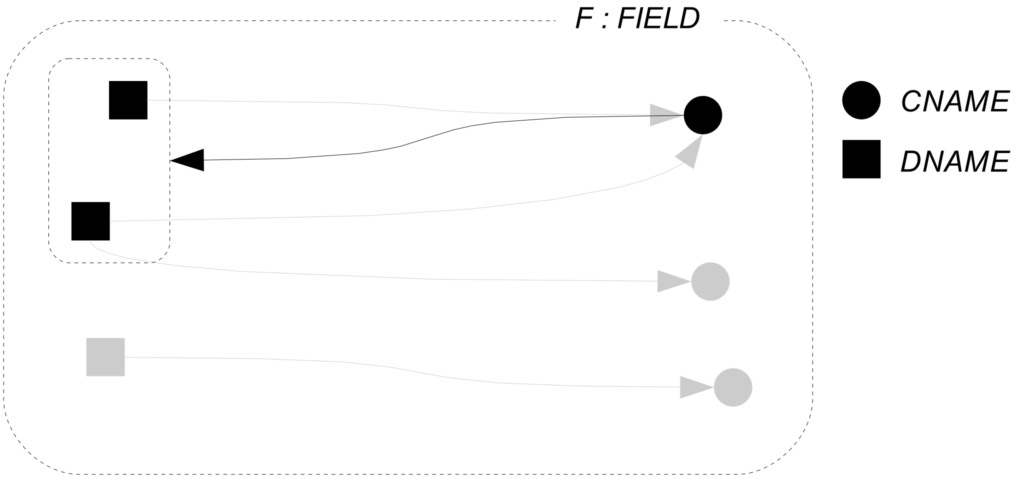


*Figure 5. Visualisation of the inverted index function for a field F.*



## 3.5 Atomic Queries

An atomic query is a query expression comprising a reference to a single field name and a single concept name. I define the *atom* function as a constructor function for query expressions of this type below.

$$atom : FNAME \times CNAME \rightarrow QUERY$$

## 3.6 Query Evaluation

A query may be "evaluated" with respect to an index. The outcome of the evaluation of a query is a set of document names, which I refer to as the "result set" or simply the "results" of a query.

I define the *results* function below, which I refer to informally as the "direct query evaluation function". This function derives a set of document names as the results of a query with respect to a particular index.

$$results : INDEX \rightarrow QUERY \rightarrow \mathbb{P}\ DNAME$$

$$\forall\, f : FNAME\,;\ \forall\, t : CNAME\,;\ \forall\, I : INDEX\ \bullet$$
$$results\ I\,(atom\,(f\,,t)) = (I\ f)^{hits}\ t$$

Note that, because an index is modeled as a mapping from a set of field names to a set of fields, for an index *I* and a field name *f* the expression (*I f*) equates to the field in *I* denoted by *f*. The expression $(I\,f)^{hits}$ then gives the inverted index function for that field and the expression $(I\,f)^{hits}\ t$ equates to the set of documents indexed in the field named *f* by the concept name *t*.

## 3.7 Vocabulary Expansion

The significant value of a structured vocabulary beyond the benefit gained from the reduction in ambiguity lies in the assumptions about relevance that can be made based on the structure of the vocabulary. This is discussed further below, but first I defined a pair of functions *bexp* and *nexp* that can be used to “expand” a concept name in the context of a particular structured vocabulary. I refer informally to the first of these functions as the “broadening vocabulary expansion function” and the second as the “narrowing vocabulary expansion function”. These functions make use of the *broader* and *narrower* relations of a structured vocabulary to derived set of concept names as the “expansion” of a single concept name. The functions are defined formally below.

$$bexp\, , nexp : StructuredVocabulary \rightarrow CNAME \rightarrow \mathbb{P}\, CNAME$$

$$\begin{aligned} &\forall\, V : StructuredVocabulary ;\ \forall\, t : CNAME \bullet \\ &\quad bexp\ V\ t = \{t\} \cup \{x : CNAME \mid t \mapsto x \in V.broader^{+}\} \ \wedge \\ &\quad nexp\ V\ t = \{t\} \cup \{x : CNAME \mid t \mapsto x \in V.narrower^{+}\} \end{aligned}$$

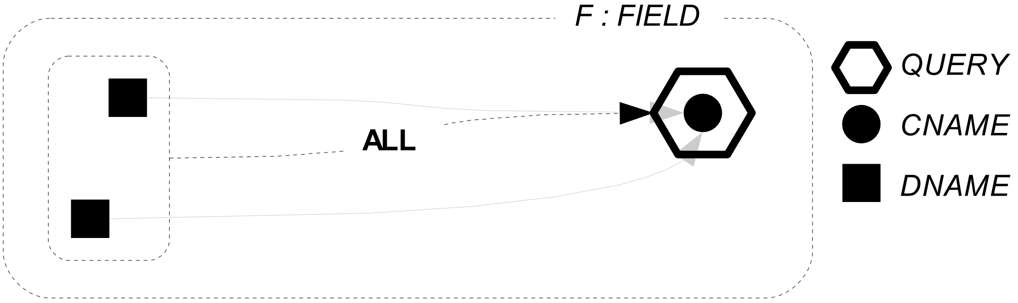
Note that a concept name is defined to occur in its own expansion.

## 3.8 Naïve Relevance Assumptions

If we start from the point of view that a document is either relevant or not relevant with respect to a query, we can derive some general assumptions regarding the implications of the structure of a controlled vocabulary for relevance and therefore recall and precision.

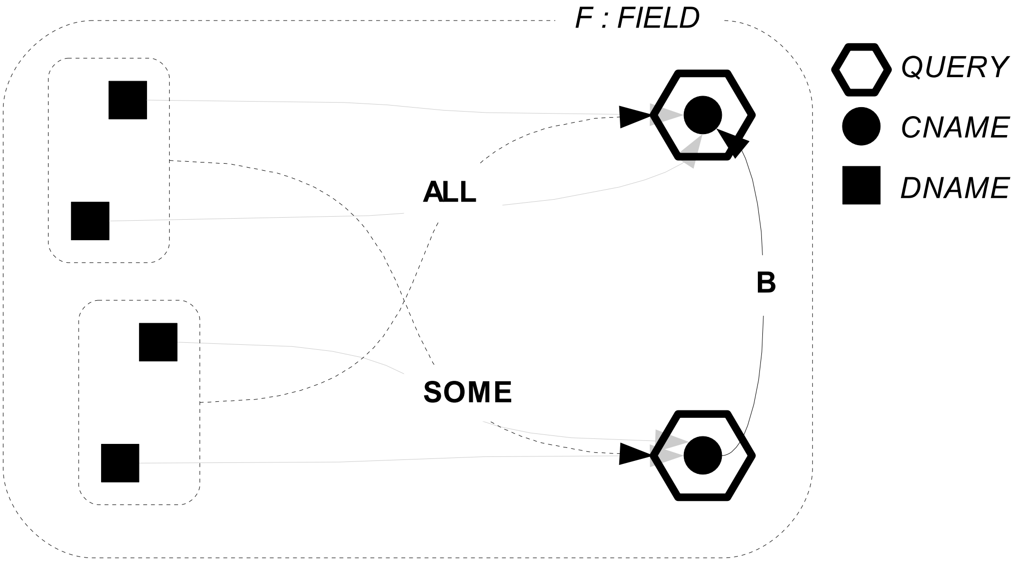
Firstly we require what I call the “naive assumption of ideal indexing”, which can be stated informally as, “**all** documents indexed with a particular concept name in a particular field are relevant to an atomic query for that concept name in that field”. This assumption is illustrated pictorially by the figure below, where the dashed arrow labeled “ALL” is intended to indicate that all documents at the root of the arrow are assumed relevant to the query at the tip of the arrow.

*Figure 6. Depiction of the “assumption of ideal indexing”.*



Given the assumption of ideal indexing, we can now define what I call the “naive assumption of broadening relevance”, which can be stated informally as, “if *y* is broader than *x*, then **all** documents relevant to a query for *x* in some field are also relevant to a query for *y* in the same field and **some** documents relevant to *y* in some field are also relevant to a query for *x* in the same field." This assumption is illustrated by the figure below, where the arrow labeled "SOME" is intended to indicate that some documents at the root of the arrow are relevant to the query at the tip of the arrow.

*Figure 7. Depiction of the "naive assumption of broadening relevance".*



Note that the assumption of broadening relevance can be applied iteratively over any number of steps in the broadening relation. E.g. If *y* is broader than x and *z* is broader than *y*, then by the naïve assumption of broadening relevance, all documents assumed relevant to a query for *x* in some field may also be assumed relevant to a query for *z* in the same field.

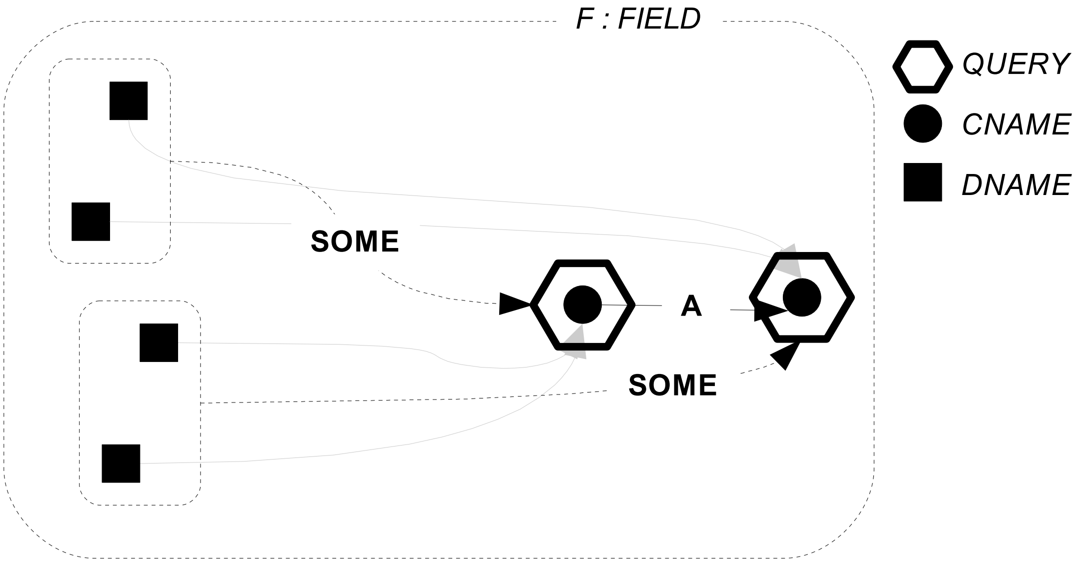
The naïve assumption of broadening relevance can also be stated as, "if *y* is broader than *x* then the set of all documents relevant to a query for *x* in some field is a **subset** of the set of all documents relevant to a query for *y* in the same field".

To explain why the naïve assumption of broadening relevance might generally be seen to hold, consider for example that all books about political history would generally be considered relevant to a search for books on history, or that all books about animal behaviour would generally be considered relevant to a search for books on zoology. If we capture the concepts of political history, history, animal behaviour and zoology in a structured vocabulary and we capture the relationships between these concepts in the broadening structure relation of that vocabulary, then the naïve assumption of broadening relevance would generally hold for an index that uses this vocabulary.

The assumption of broadening relevance is of course a first approximation and is unlikely to hold universally, especially where a structured vocabulary has a very "deep" broadening relation. In chapter 5 I refine this assumption, however here I begin to frame a definition for the intended semantics of a broadening relation in terms of this assumption, by stating that the broadening relation implies the naïve assumption of broadening relevance to a first approximation. A key characteristic of the naïve assumption of broadening relevance is its asymmetry. It is this feature that fundamentally distinguishes the broadening/narrowing relations from the associating relation.

I also define the "naive assumption of associating relevance" informally as, "if x is associated to y, then **some** documents relevant to a query for *x* in some field are also relevant to a query for *y* in the same field and vice versa". The assumption of associating relevance is illustrated by the figure below. I begin to frame a definition of the intended semantics of the associating relation by stating that an associating relation implies the naïve assumption of associating relevance to a first approximation.

*Figure 8. Depiction of the "naive assumption of associating relevance".*



By the naïve assumption of broadening relevance, because **all** documents relevant to a narrower query are assumed relevant to a broader query in the same field, we are then justified in including those documents indexed with narrower concept names in the results of the broader query. In this way relevant results are included in the result set that otherwise would not be and, if the naïve assumption holds, recall will be improved without loss of precision. However, because only **some** documents relevant to a broader query may be assumed relevant to a narrower query, although recall may be marginally improved by including documents indexed with broader concept names in a narrower query, precision will drop because a significant number of non-relevant items will also be included. The same applies with respect to the associating relation.

## 3.9 Naïve Index Expansion

The previous section introduced the naïve assumptions of broadening and associative relevance and stated that the assumption of broadening relevance justifies the inclusion of additional results in a result set. This may be achieved by expanding an index in accordance with the assumptions and then directly evaluating queries with respect to the expanded index.

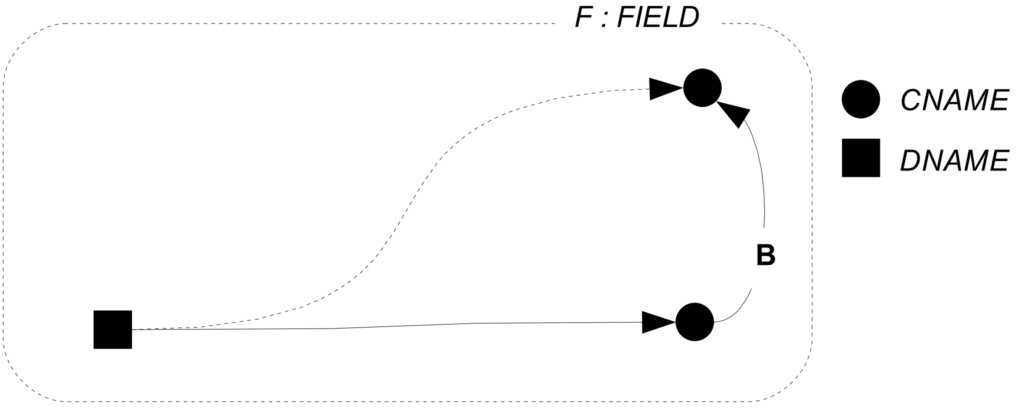
This section provides a formal definition for the expansion of an index in accordance with the naïve assumption of broadening relevance, which I call the naïve expansion of an index. The function *fexp* defined below gives the naïve expansion of a field and the function *iexp* derives the naïve expansion of an index with respect to a structured vocabulary from *fexp*.

$$
\begin{array}{|l}
fexp : StructuredVocabulary \rightarrow FIELD \rightarrow FIELD \\
iexp : StructuredVocabulary \rightarrow INDEX \rightarrow INDEX \\
\hline
\forall\, V : StructuredVocabulary ;\ \forall\, I : INDEX ;\ \forall\, F : FIELD \bullet \\
\quad fexp\ V\ F = \{ d : DNAME ;\ x, t : CNAME \mid d \mapsto t \in F \land x \in (bexp\ V\ t) \bullet d \mapsto x \} \land \\
\quad iexp\ V\ I = \{ f : FNAME \mid f \in dom\ I \bullet f \mapsto fexp\ V\ (I\ f) \}
\end{array}
$$

Note that the expression (*bexp V t*) gives the broadening vocabulary expansion for the concept name *t* with respect to the structured vocabulary *V*. Note also that the expression (*I f*) equates to the field in *I* denoted by *f* and that therefore the expression *fexp V* (*I f*) equates to the naïve expansion of the field in *I* denoted by *f* with respect to the vocabulary *V*.

The figure below illustrates the naïve expansion of a field by the above method, where the dashed arc indicates a member of the expanded field.

*Figure 9. Depiction of the naïve expansion of a field.*



## 3.10 Summary

This chapter has laid the foundations for a theory of retrieval using structured vocabularies. A “structured vocabulary” was defined as comprising a set of “concept names” to be applied as an indexing language and one or more binary relations on this set. An “index” was defined as a mapping from a set of “field names” to a set of “fields”, where a field was defined as a relation between a set of “document names” and a set of “concept names”. An “atomic query” was defined as a query expression encapsulating a reference to a single field name and a single concept name. The direct evaluation of atomic queries with respect to an index was defined by reference to the “inverted index function”. Finally the notions of vocabulary expansion and index expansion were formally defined and the underlying assumptions regarding inferences that can drawn from the structure of a vocabulary about the relevance of indexed documents were discussed.

## 3.11 Use Case A - “Blogpress” Web-log Application

### *3.11.1 Informal Description*

“BlogPress” is a web-log publication tool. Laura maintains a web-log using the BlogPress tool.

The BlogPress tool allows a user to maintain a set of "categories". Each category can have a "category parent". Laura has defined three categories, labeled "General", "Politics" and "Travel" respectively. The category labeled "General" is the category parent of the other two categories.

The BlogPress tool allows a user to publish "posts" and to associated one or more categories with each post. Laura has published two posts, entitled "What I think about politics" and "My trip to Havana". The first of these is posted to the "Politics" category, the second is posted to the "Travel" category.

Laura's three categories are displayed as hyperlinks arranged as a hierarchy on the front page of her web-log, as shown below.

```
General
  Politics
  Travel
```

Anyone visiting her web-log can click on one of these links. When a visitor clicks on the "Politics" link, the post entitled "What I think about politics" is displayed. When a visitor clicks on the "Travel" link, the post entitled "My trip to Havana" is displayed. When a visitor clicks on the "General" link, both posts are displayed.

### *3.11.2 Classification*

**Vocabulary structure**: monohierarchical.

**Index structure**: single-field, relational fields, no coordination.

**Query capability**: atomic queries only.

**Query evaluation strategy**: direct evaluation using naïve expansion of either index or query.

# 4 A Theory of Retrieval Using Structured Vocabularies: Composite Queries

This chapter extends the theory developed in the previous chapter to consider query expressions that allow several atomic expressions to be combined into a more complicated statement of need. Specifically four types of query expression are defined, being "required-optional-prohibited" (ROP) expressions, "and" expressions, "or" expressions and "not" expressions. Each of these types of "composite" query expression may have a number of child expressions and expressions may be arbitrarily nested.

A theory is developed for how these queries may be evaluated, how they may be decomposed into their constituent atoms, how the results of composite queries may be given a numeric score and hence ranked in order of relevance to the query and how queries may be expanded according to the assumption of broadening relevance.

## 4.1 Composite Query Expressions

I define four types of composite query expression. The "required-optional-prohibited" query expression (*rop*) is a function of three disjoint sets of query expressions, being the "required", "optional" and "prohibited" components of the expression. The "and" (*and*) and "or" (*or*) query expression are both functions of a single set of query expressions, being the components of the expression. The "not" (*not*) query expression is a function of a single query expression, being the component of the expression. The components of any type of composite query expression may be query expressions of any type, which is to say that query expressions may be arbitrarily nested.

$$\begin{array}{|l}
rop : (\mathbb{P}\,QUERY \times \mathbb{P}\,QUERY \times \mathbb{P}\,QUERY) \rightarrow QUERY \\
and : \mathbb{P}\,QUERY \rightarrow QUERY \\
or : \mathbb{P}\,QUERY \rightarrow QUERY \\
not : QUERY \rightarrow QUERY
\end{array}$$

## 4.2 Evaluation of Composite Queries

The schema below extends the definition of the direct query evaluation function given in the previous chapter to provide a theory of the evaluation of composite query expressions. Arbitrarily composed query expressions may be evaluated by a recursive application of the formulas.

The evaluation of *and*, *or* and *not* expression types is intended to be in agreement with the intuitive understanding of the words "and" "or" and "not" in the context of Boolean logic. I.e. The results of an *and* expression are obtained by taking the set intersection of the results of all component expressions, the results of an *or* expression are obtained by taking the set union of the results of all component expressions and the results of a *not* expression are obtained by taking the set difference between the totality of the indexed collection and the results of the component expression.

Note that the presence of “optional” components in an *rop* expression does not have any influence on the outcome of the query evaluation – this is intended to be consistent with the intuitive understanding of the entirely optional nature of these components. However the presence of optional components may of course influence the scoring of results (see below). In practice a retrieval system may choose to loosen the interpretation of “optional” components in the absence of any “required” components and require that results match at least on “optional” component.

Note also in the definitions below the use of the relational image operator over sets of query expressions – this is simply to provide a more compact presentation of the definitions.

$$results : INDEX \rightarrow QUERY \rightarrow \mathbb{P}\, DNAME$$

$$\begin{array}{l}
\forall R, O, P : \mathbb{P}\, QUERY;\ \forall I : INDEX \bullet \\
\quad results\ I\,(rop(\varnothing, O, P)) = (alldocs\ I) \setminus \cup(results\ I\, \llparenthesis P \rrparenthesis) \wedge \\
\quad R \neq \varnothing \Rightarrow results\ I\,(rop(R, O, P)) = \cap(results\ I\, \llparenthesis R \rrparenthesis) \setminus \cup(results\ I\, \llparenthesis P \rrparenthesis) \\
\forall E : \mathbb{P}\, QUERY;\ \forall I : INDEX \bullet \\
\quad results\ I\,(and\ E) = \cap(results\ I\, \llparenthesis E \rrparenthesis) \wedge \\
\quad results\ I\,(or\ E) = \cup(results\ I\, \llparenthesis E \rrparenthesis) \\
\forall e : QUERY;\ \forall I : INDEX \bullet \\
\quad results\ I\,(not\ e) = (alldocs\ I) \setminus (results\ I\ e)
\end{array}$$

## 4.3 Query Decomposition

As a prelude to establishing methods for deriving numeric scores for the results of a query I first define functions to derive the atomic components of a query. The *posatoms* function returns the set of all query atoms that make a “positive” contribution to the query (either by causing the inclusion of matching documents or by improving the score of already matching documents) and the *negatoms* function derives the set of all query atoms that make a “negative” contribution to a query (by causing the exclusion of matching documents). The definitions of the *posatoms* and *negatoms* functions are mutually dependent. Note that the value of these functions for an arbitrarily composed query expression is derived by a recursive application of the formulas to the components of the expression.

$$\begin{array}{l}
posatoms : QUERY \rightarrow \mathbb{P}\, QUERY \\
negatoms : QUERY \rightarrow \mathbb{P}\, QUERY \\
\hline
\forall R, O, P : \mathbb{P}\, QUERY \bullet \\
\quad posatoms(rop(R, O, P)) = (\bigcup posatoms \llparenthesis R \cup O \rrparenthesis) \cup (\bigcup negatoms \llparenthesis P \rrparenthesis) \wedge \\
\quad negatoms(rop(R, O, P)) = (\bigcup negatoms \llparenthesis R \cup O \rrparenthesis) \cup (\bigcup posatoms \llparenthesis P \rrparenthesis) \\
\forall E : \mathbb{P}\, QUERY \bullet \\
\quad posatoms(and\ E) = \bigcup posatoms \llparenthesis E \rrparenthesis \wedge \\
\quad negatoms(and\ E) = \bigcup negatoms \llparenthesis E \rrparenthesis \wedge \\
\quad posatoms(or\ E) = \bigcup posatoms \llparenthesis E \rrparenthesis \wedge \\
\quad negatoms(or\ E) = \bigcup negatoms \llparenthesis E \rrparenthesis \\
\forall e : QUERY \bullet \\
\quad posatoms(not\ e) = negatoms\ e \wedge \\
\quad negatoms(not\ e) = posatoms\ e \\
\forall f : FNAME ; \forall t : CNAME \bullet \\
\quad posatoms(atom(f, t)) = \{atom(f, t)\} \wedge \\
\quad negatoms(atom(f, t)) = \emptyset
\end{array}$$

## 4.4 Scoring Results

There is no way of differentiating between the results of an atomic query because documents either do or do not match the query expression. However with composite queries there is the possibility that documents in the result set will match different combinations of query atoms, if either *rop* or *or* expressions have been used (because then documents may be included in the results without having to match all query atoms). In these cases we may derive a numeric score to reflect the degree to which a member of a result set matches the query and present the results in a ranked order of greatest relevance to the query, assuming that the numeric score we derive accurately reflects relative differences in relevance between results.

The simplest and most intuitive scoring method is simply to count the number of positive atoms for which a document matches and return that number as the score. This method is defined formally in the schema below by the *unweightedscore* function. It can be argued that this number reflects relative relevance because for example a document that matches both atoms of a simple two-atom *or* expression is intuitively a “better match” than documents that only match one or the other atom.

$$\begin{array}{l}
matchingposatoms : INDEX \rightarrow (QUERY \times DNAME) \rightarrow \mathbb{P}\, QUERY \\
unweightedscore : INDEX \rightarrow (QUERY \times DNAME) \rightarrow \mathbb{Z} \\
\hline
\forall I : INDEX ; \forall q : QUERY ; \forall d : DNAME \bullet \\
\quad matchingposatoms\ I\ (q, d) = \{e : QUERY \mid e \in posatoms(q) \wedge d \in (results\ I\ e)\} \wedge \\
\quad unweightedscore\ I\ (q, d) = \#\, matchingposatoms\ I\ (q, d)
\end{array}$$

Commonly in text retrieval systems a greater significance is placed on “terms” that occur less frequently in an index – because they occur less frequently they have a greater discriminatory power when used within queries. A numerical representation may be given to this frequency of occurrence, typically called either the “inverse document frequency weight” (IDF) or “collection frequency weight” (CFW). Although the IDF weight metric is seldom if ever applied to indexes involving controlled vocabularies, there may be reason to suggest that such a metric could improve the performance of search and therefore I provide a definition consistent with the literature below, simplified of course because there is no need to take account of within-document frequency or document length.

$$
\begin{array}{|l}
idf : FIELD \times CNAME \rightarrow \mathbb{R} \\
\hline
\forall\, F : FIELD ;\ \forall\, t : CNAME \bullet \\
\quad idf\,(F\,, t) = \log \#\,(dom\, F\,) - \log \#\,(F^{hits}\ t)
\end{array}
$$

I have chosen to model the inverse document frequency as being specific to each separate field – this is because fields may be queried independently and are essentially orthogonal axes of retrieval. The inverse document frequency of a concept name with respect to a particular field is given by taking the logarithm (to any base) of the size of the domain of the field (given by the expression #(*dom F*) – the number of members of the domain of the field) and subtracting the logarithm of the number of documents indexed with that concept name in the given field (given by the expression #($F^{hits}$ *t*)).

The IDF metric may be incorporated into a result score as defined by the schema below where the function *idfscore* multiplies the value of the “field boost” (*fb f*) by the inverse document frequency and sums this value over all matching positive atoms of the query. The field boost is an arbitrary function that can be used to assign a greater importance to particular fields in an index.

$$
\begin{array}{|l}
idfscore : INDEX \rightarrow (QUERY \times DNAME\,) \rightarrow \mathbb{R} \\
\hline
\forall\, I : INDEX\,;\ \forall\, q : QUERY\,;\ \forall\, d : DNAME\ \bullet \\
\quad idfscore\ I\,(\,q\,, d\,) = \\
\sum \{\, f : FNAME\,; t : CNAME \mid atom(\,f\,, t) \in matchingposatoms(\,q\,, d\,) \bullet (\,fb\ f\,) * idf\,(\,(\,I\ f\,)\,, t)\}
\end{array}
$$

## 4.5 Naïve Query Expansion

The previous chapter defined the naïve assumption of broadening relevance and described its application to the expansion of an index. The same assumption may be used to expand a query and these two approaches are mathematically equivalent, although they are not computationally equivalent and hence there is value in exploring both alternatives. They are not computationally equivalent because the expansion of an index is a one-time operation which can be computed prior to the evaluation of queries, whereas the expansion of a query must be computed in real-time and hence the time taken to compute the expansion may influence the responsiveness of a user-interface. The value of query expansion over index expansion is in the potential to deliver interesting dynamic functionality within query systems, however this potential cannot be explored until I have developed the more sophisticated principles of limited cost expansions in the next chapter. Prior to that the schema below establishes a basic definition for the naïve expansion of an arbitrarily composed query expression.

$$qexp : StructuredVocabulary \rightarrow QUERY \rightarrow QUERY$$

$$\begin{array}{l}
\forall\, V : StructuredVocabulary;\ \forall\, R, O, P : \mathbb{P}\, QUERY \bullet \\
\quad qexp\ V\ (rop\ (R, O, P)) = rop\ (qexp\ V \llparenthesis R \rrparenthesis, qexp\ V \llparenthesis O \rrparenthesis, qexp\ V \llparenthesis P \rrparenthesis) \\
\forall\, V : StructuredVocabulary;\ \forall\, E : \mathbb{P}\, QUERY \bullet \\
\quad qexp\ V\ (and\ E) = and\ (qexp\ V \llparenthesis E \rrparenthesis)\ \wedge \\
\quad qexp\ V\ (or\ E) = or\ (qexp\ V \llparenthesis E \rrparenthesis) \\
\forall\, V : StructuredVocabulary;\ \forall\, e : QUERY \bullet \\
\quad qexp\ V\ (not\ e) = not\ (qexp\ V\ e) \\
\forall\, V : StructuredVocabulary;\ \forall\, f : FNAME;\ \forall\, t : CNAME \bullet \\
\quad qexp\ V\ (atom\ (f, t)) = or\ (\{ x : V.T \mid x \in (nexp\ V\ t) \bullet atom\ (f, x) \})
\end{array}$$

Note that composite query expressions essentially remain unaffected by the query expansion. The effect of the expansion is to replace each atom in the query with an *or* expression composing atoms derived from the narrowing expansion of the concept name in the original atom.

## 4.6 Summary

In this chapter I have defined four types of composite query expression, being “required-optional-prohibited” expressions, “and” expressions, “or” expressions and “not” expressions. These types of query expression allow several atomic expressions to be composed and also allow composite expressions to be arbitrarily nested. This chapter has also extended the definition of the direct query evaluation function for composite queries, where the evaluation of “and”, “or” and “not” expressions is defined formally as set operations on the results of expression components. Functions were defined for the decomposition of an arbitrarily nested composite query into its constituent atoms, which is required for the definition of both unweighted and inverse document frequency weighted systems of scoring results. Finally the naïve expansion of a query consistent with the naïve assumption of broadening relevance was defined.

## 4.7 Use Case B – “Local Government” Document Management System

### *4.7.1 Informal Description*

A local government department uses a document management system to manage the reports it produces on a variety of topics. When a new report is added to the system, some metadata about the report is captured. The metadata includes a "subject" field, in which one or more descriptors from a thesaurus must be entered.

The table below gives the title and subject fields of metadata for some of the reports in the system, with multiple descriptors in the subject field delimited by semi-colons.

| Title | Subject |
|---|---|
| Townscape Heritage Initiative Report | Urban conservation; Historic buildings |
| Historic Parks and Gardens Report | Urban conservation; Parks and gardens |
| Outdoor Play Facilities Report | Parks and gardens; Playgrounds |

Below is an extract from the thesaurus used by the department, as it appears in its printed form, with only BT relationships shown for brevity.

```
Urban conservation
  BT Built environment

Historic buildings
  BT Heritage

Parks and gardens
  BT Leisure and culture

Playgrounds
  BT Sports and recreation facilities
```

The document management system provides basic and advanced searching interfaces.

The basic interface allows a user to browse the thesaurus, and select a descriptor. Once a descriptor is selected, a list of results is displayed. A list of additional descriptors that may be used to "refine" the search is also displayed. The user may then select a descriptor from this additional list, which has the effect of refining the previous search. This process of refinement may be iterated until no additional descriptors are available.

The advanced interface allows users familiar with the thesaurus to build search expressions using descriptors and the keywords 'AND', 'OR' and 'NOT'.

A user, Steve, interacts with the basic user interface. He first selects the descriptor "Built environment", and is presented with the two reports entitled "Townscape Heritage Initiative Report" and "Historic Parks and Gardens Report". He then selects the descriptor "Parks and gardens" from the additional list, and only the report entitled "Historic Parks and Gardens Report" is displayed.

Another user, Joan, interacts with the advanced user interface. The table below shows the search expressions she enters, with the list of results she is presented for each.

| | Advanced Search Expression | Results List |
|---|---|---|
| 1 | "Built environment OR Leisure and culture" | 1. Historic Parks and Gardens Report<br>2. Townscape Heritage Initiative Report<br>3. Outdoor Play Facilities Report |
| 2 | "Built environment AND Leisure and culture" | 1. Historic Parks and Gardens Report |
| 3 | "Built environment NOT Leisure and culture" | 1. Townscape Heritage Initiative Report |

### *4.7.2 Classification*

**Vocabulary structure**: monohierarchical, associative.

**Index structure**: single-field, relational fields, no coordination.

**Query capability**: composite queries.

**Query evaluation strategy**: direct evaluation using naïve expansion of either index or query.

## 4.8 Use Case C - "National Environmental Directory"

### *4.8.1 Informal Description*

The National Environmental Directory is an on-line directory of organisations whose business relates in some way to the natural environment. The following features of an organisation are captured for each entry in the directory: "Topic of Interest"; "Organisation Type"; "Operational Area". A different controlled vocabulary is used to describe each of these features.

The table below shows the entries held by the directory (multiple values are delimited by a semi-colon).

| Organisation Name | Topic of Interest | Organisation Type | Operational Area |
|---|---|---|---|
| Barn Owl Trust | Animal Welfare; Bird Species | Registered Charity | United Kingdom |
| Society for Environmental Exploration (SEE) | Wild Animals (Welfare of) | NGO | Worldwide |
| rECOrd | Wild Animals (Welfare of) | Registered Charity | Cheshire |

The three controlled vocabularies used are each organised as a hierarchy. Below is an extract from the "Topic of Interest" vocabulary, as it appears on the website.

```
Animal Welfare
   Wild Animals (Welfare of)
Species
   Bird Species
```

Below is an extract from the “Organisation Type” vocabulary, as it appears on the website.

```
Not For Profit
   NGO
   Charitable
      Registered Charity
```

Below is an extract from the “Operational Area” vocabulary, as it appears on the website.

```
Worldwide
  United Kingdom
    England
      North West England
        Cheshire
```

A visitor to the directory website is first presented with a list of the top level options for each feature. Upon selecting an option, the visitor is immediately presented with a list of matching results. The visitor may then refine the search, by either selecting an additional option from any feature, or by selecting a more specific option from the currently selected feature. This process of refinement may be iterated until no further options are available.

A visitor, Heather, first selects the option "Animal Welfare" from the "Topic of Interest" feature. She is presented with a list of three results. She then refines her search by adding the option "Bird Species" from the "Topic of Interest" feature, and is presented with a single result, being the entry for the Barn Owl Trust.

Another visitor, Michael, first selects the option "Not For Profit" from the "Organisation Type" feature. He is presented with a list of three results. He then refines his search by adding the option "Worldwide" from the "Operational Area" feature, and is presented with a single result, being the entry for the Society for Environmental Exploration. He then removes the "Worldwide" option from the search, and adds the option "United Kingdom" from the "Operational Area" feature, and is presented with a list of two results, being the entries for the Barn Owl Trust, and rECOrd.

### *4.8.2 Classification*

**Vocabulary structure**: monohierarchical, associative.

**Index structure**: multiple-field, relational fields, no coordination.

**Query capability**: composite queries (*and* expressions only).

**Query evaluation strategy**: direct evaluation using naïve expansion of either query or index.

# 5 A Theory of Retrieval Using Structured Vocabularies: Limited Cost Expansions

The previous two chapters have developed a basic theory for the use of structured vocabularies for retrieval. I have provided formal definitions for the components of a structured vocabulary, for the structure of an index and of composite queries, for the direct evaluation of arbitrary composite queries with respect to an index and for the expansion of either an index or a query in agreement with the naïve assumptions of broadening and associating relevance. These last definitions are critical because, if the naïve assumption of broadening relevance holds for a given index and a given vocabulary, by expanding result sets in agreement with this assumption relevant results will be included that otherwise would not be. In other words the naïve relevance assumptions lead to improved *recall* of searches.

This chapter refines these basic assumptions about relevance by considering inferences regarding the *relative probability* of relevance drawn from vocabulary structures. By modeling the relative probability of relevance as a numerical function, a query evaluation strategy may exploit the structure of a vocabulary to a greater extent than as previously discussed, thereby further improving the recall of search results, whilst also ranking expanded results in a detailed manner such that result sets are unlikely to be obscured by non-relevant results, effectively increasing the precision of search results also.

## 5.1 Quantified Relevance Assumptions

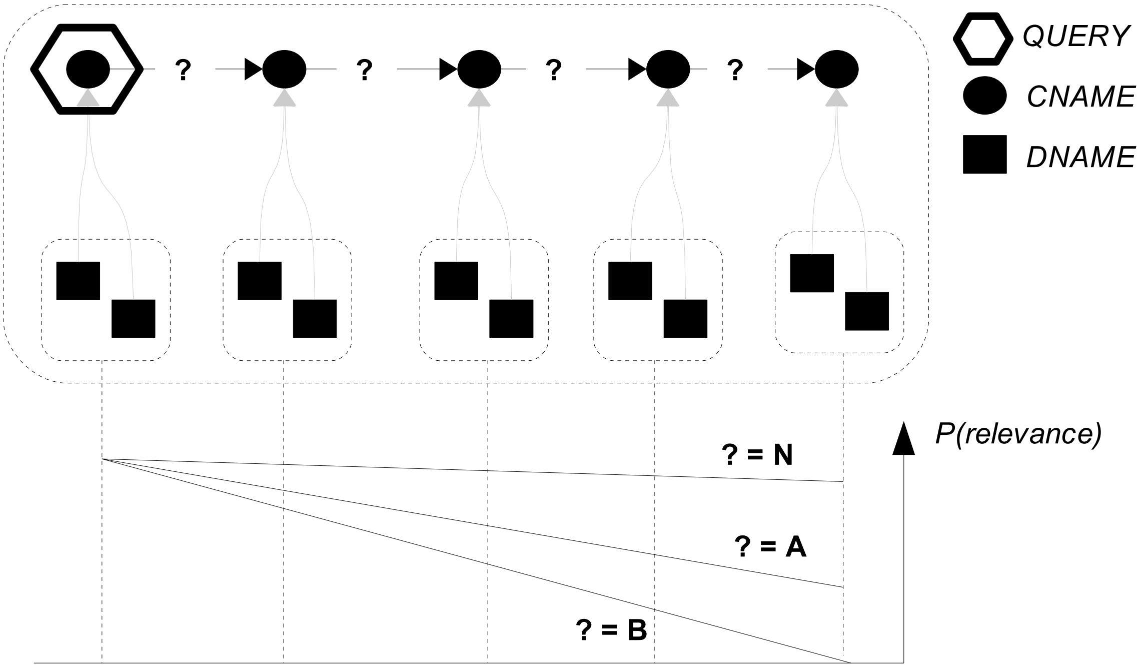
Here I state quantified versions of the assumptions of ideal indexing and broadening and associating relevance so that numerical estimates may be made as to the likelihood of relative relevance of particular results based on the structure of a vocabulary.

The "naive assumption of ideal indexing" was stated in chapter 3 as, "**all** documents indexed with a particular concept name in a particular field are relevant to an atomic query for that concept name in that field". This assumption provides the philosophical justification for the direct query evaluation function. As a basis for further definitions, I state the "quantified assumption of ideal indexing" informally as, "the probability that a document indexed with a particular concept name in a particular field is relevant to a query for the same concept name in the same field is approaching unity."

The "naive assumption of broadening relevance" was stated in chapter 3 as, "if *y* is broader than *x*, then **all** documents relevant to a query for *x* in some field are also relevant to a query for *y* in the same field and **some** documents relevant to *y* in some field are also relevant to a query for *x* in the same field." The first part of this definition provides the justification for the naïve expansions of either indexes or queries. However, it was noted that the naïve assumption of broadening relevance is likely to break down under certain circumstances, in particular where a vocabulary has a "deep" broadening relation. Also the naïve assumptions noted that **some** relevant results can be inferred from the associating relation, however with no way of indicating that the likelihood of relevance is less here than for the broadening relation the inclusion of results by association is likely to obscure result sets with irrelevant results.

The basic principle of the "quantified assumption of relative relevance" is that the probability of relevance diminishes according to some numerical function as we expand outward from the focus of the query. This idea is illustrated pictorially in figure nnn below. Beginning from a particular query and the assumption of ideal indexing, as we move away from the query focus via the structure relations of the indexing vocabulary we can describe a function to estimate how the probability of relevance of the correspondingly indexed documents diminishes.

*Figure 10. Depiction of the quantified assumption of relative relevance.*



The mathematical components of the quantified assumption of relative relevance are then (i) the form of the numerical function chosen to model the diminishing relevance (the figure above illustrates a linear function but there is no *a priori* reason for choosing only a function of this form) and (ii) the values of the parameters chosen to reflect the different effects of the different structure relations. Generally speaking it will be assumed that, under query expansion, the loss of relevance due to traversing a narrowing relationship is small, and that the loss of relevance due to other relationship types is larger. However the purpose of this chapter is not to provide suggestions for particular parameter values but rather to provide the mathematical framework that can be used to explore the application of these quantified assumptions. It is anticipated that effective parameter values will need to be established empirically and may vary significantly between different indexes and different vocabularies.

I have chosen to use the name “limited cost expansions” to refer to the mathematical framework for modeling the application of these ideas. This is because the loss of relevance due to expansion along a set of structure relationships can be modeled as a numerical “cost” of traversing a path in a directed graph. A limited cost expansion is therefore an expansion that proceeds outwards until a predefined cost limit is reached. The value of the computed cost can then be inverted to give an “expansion weight” which gives a numerical estimate of the likelihood of relevance and this value can be factored into result scores to give a ranking that is sensitive to detail in the structure of the indexing vocabulary.

## 5.2 Generic Functions of Graphs and Paths

I first establish basic mathematical tools for manipulating directed graphs and paths in those graphs. I assume that a directed graph (hereafter simply “graph”) is a set of ordered pairs. I.e. A graph is a binary relation (a set of arcs) on some set of objects (nodes). I model a path as a sequence of nodes.

The *ispath* function defined below simply establishes whether a particular sequence of nodes is indeed a path in some graph. In order to be a path an ordered pair must exist in the graph for every adjacent pair of objects in the sequence and a node cannot be revisited which is to say that it may appear only once in the sequence.

[X]

$$ispath : (X \leftrightarrow X) \rightarrow seq\ X \rightarrow \{true, false\}$$

$$\begin{array}{l} \forall G : X \leftrightarrow X;\ \forall p : seq\ X \bullet \\ \quad ispath\ G\ p \Leftrightarrow \#p > 1 \wedge \forall i, j : 1..(\#p - 1) \bullet p\,i \mapsto p(i+1) \in G \wedge i \neq j \Rightarrow p\,i \neq p\,j \end{array}$$

Note that because *p* is a sequence the expression *p i* gives the $i^{th}$ element in the sequence and *p*(*i*+1) gives the $(i+1)^{th}$ element in the sequence.

The *paths* function derives the set of all paths from one node to another in a given graph.

[X]

$$paths : (X \leftrightarrow X) \rightarrow X \times X \rightarrow seq\ X$$

$$\begin{array}{l} \forall G : X \leftrightarrow X;\ \forall x, y : X \bullet \\ \quad paths\ G\ (x, y) = \{ p : seq\ X \mid first\ p = x \wedge last\ p = y \wedge (ispath\ G\ p) \} \end{array}$$

The *pathsfrom* function derives the set of all terminating paths starting from a given node in a given graph. I.e. It gives the set of all paths starting from the given node such that the path may not be extended by the given graph.

$$
\begin{array}{l}
[X] \\
\hline
\mathit{pathsfrom} : (X \leftrightarrow X) \rightarrow X \rightarrow \mathrm{seq}\, X \\
\hline
\forall G : X \leftrightarrow X ;\ \forall x : X \bullet \\
\quad \mathit{pathsfrom}\, G\, x = \{ p : \mathrm{seq}\, X \mid \mathit{first}\ p = x \land (\mathit{ispath}\, G\ p) \land \forall y : X \bullet \mathit{last}\ p \mapsto y \notin G \}
\end{array}
$$

## 5.3 Broadening and Depth

I define the “depth” of a concept name in some structured vocabulary as being the length of the shortest terminating path starting from that node in the *broader* relation. I.e. If the vocabulary is polyhierarchical then the depth is equal to the number of levels above the given node in the shortest route to the top of the hierarchy.

$$
\begin{array}{l}
\mathit{depth} : \mathit{StructuredVocabulary} \rightarrow \mathit{CNAME} \rightarrow \mathbb{Z} \\
\hline
\forall V : \mathit{StructuredVocabulary} \bullet \forall t : V.T \bullet \\
\quad \mathit{depth}\, V\ t = \mathit{min}(\{ p : \mathrm{seq}\, V.T \mid p \in (\mathit{pathsfrom}\, V.\mathit{broader}\ t) \bullet \#\ p \})
\end{array}
$$

## 5.4 Arc Weighting

In order to derive a metric for the “distance” between two nodes in the structure graph of a vocabulary (or more generally speaking the “cost” of moving from one node to another) we first require a function to derive a “weight” for different types of arc. A general definition is given below for the form of an arc weight function.

$$
\mathit{arcweight} : \mathit{StructuredVocabulary} \rightarrow (\mathit{CNAME} \times \mathit{CNAME}) \rightarrow \mathbb{R}
$$

We have assumed that the broadening, narrowing and associating relations have different implications for the diminishing of relevance as we expand along paths in these relations and therefore typically we will want to assign a weight to an arc discretely on the basis of which of these relations the arc belongs to. The schema below provides a definition for such a function, where $w_b$ $w_n$ and $w_a$ are the arc weights for arcs in the broadening narrowing and associating relations respectively.

$$
\begin{array}{l}
\mathrm{DiscreteArcWeight} \\
\hline
w_b, w_n, w_a : \mathbb{R} \\
\hline
\forall V : \mathit{StructuredVocabulary} \bullet \forall x, y : V.T \bullet \\
\quad x \mapsto y \in V.\mathit{broader} \Rightarrow \mathit{arcweight}\, V\,(x, y) = w_b \ \land \\
\quad x \mapsto y \in V.\mathit{narrower} \Rightarrow \mathit{arcweight}\, V\,(x, y) = w_n \ \land \\
\quad x \mapsto y \in V.\mathit{associated} \Rightarrow \mathit{arcweight}\, V\,(x, y) = w_a \ \land \\
\quad x \mapsto y \notin V.G \Rightarrow \mathit{arcweight}\, V\,(x, y) = \infty
\end{array}
$$

## 5.5 The Cost of Traversing a Path

As mentioned above the underlying assumption is that as we expand along a path in the structure graph of a vocabulary away from the focus of a query then the relevance of the correspondingly indexed document will diminish according to some numerical function. Here I model this diminishing of relevance by first defining a function that computes a numerical "cost" of traversing a particular path in a given graph, where the "cost" could be understood as the cost in terms of loss of probability of relevance. I.e. The greater the accumulated cost, the less the likelihood of relevance. The shape of the cost function chosen provides a numerical definition for our assumption about the shape of the diminishing relevance function, although the cost function will need to be inverted (see below).

Other authors have restricted themselves to a discussion of "distance" however I see the notion of "cost" as a generalisation of the notion of distance because a distance function is a special case of a cost function where the cost increase is linear.

Before I can provide a definition of a function for deriving the cost of traversing an entire path I first give a definition of a function for calculating the cost of traversing a single step in a path. The general form of the function is given below.

$$stepcost : StructuredVocabulary \rightarrow (seq\ CNAME \times \mathbb{Z}) \rightarrow \mathbb{R}$$

An expression *stepcost V* (*p, i*) therefore gives the cost of traversing the $i^{th}$ step in the path *p* relative to the structured vocabulary *V*.

The schema below provides a definition for a step cost function that is independent of the depth of the nodes in the context of the given vocabulary.

DepthIndependentStepCost

$$k, m : \mathbb{R}$$

$$\forall V : StructuredVocabulary \bullet \forall p : seq\ V.T \bullet \forall i : 1 .. (\# p - 1) \bullet$$
$$stepcost\ V\ (p, i) = k * i^m * arcweight\ V\ (p\ i, p(i+1))$$

Note that by including the $i^m$ term in the right hand side of the equation above I have introduced a dependency of the cost on the number of steps taken so far. This allows the cost function to take shapes other than a linear form (the linear form is achieved by setting *m*=0). Whether the shape of the cost function has a significant impact on retrieval performance is an interesting area for empirical investigation.

Some authors have suggested that the depth of the nodes must be taken into account when calculating the "distance" between two nodes and that the "distance" is relatively less at greater depth. The schema below provides a definition of a step cost function that also factors in the depth at which the nodes in the step occur.

DepthDedependentStepCost

$$j\,,k\,,m : \mathbb{R}$$

$$\forall\, V : StructuredVocabulary \bullet \forall\, p : seq\, V.T \ \bullet \forall\, i : 1\,..(\#\, p - 1) \bullet$$
$$stepcost\; V\,(p\,,i) = \frac{k * i^{m} * arcweight\; V\,(p\,i\,,p(i+1))}{j + (depth\; V\; p\,(i))}$$

We are now in a position to define a function to derive the total cost of traversing a path in the structure graph of a vocabulary. This function simply sums the costs of all the steps in the path.

$$cost : StructuredVocabulary \longrightarrow seq\, CNAME \longrightarrow \mathbb{R}$$

$$\forall\, V : StructuredVocabulary \bullet \forall\, p : seq\; V.T \bullet$$
$$cost\; V\; p = \sum_{i=1}^{\#\, p-1} stepcost\; V\,(p\,,i)$$

## 5.6 Minimum Path Cost

Given that there may be more than one path between any two nodes in the structure graph of a controlled vocabulary, we require a function that returns the cost of the “cheapest” path for any two nodes, because it is cost of the “cheapest” path that reflects the highest probability of relevance. The schema below defines a function *mincost* which derives the cost of the “cheapest” path according to some path cost function.

$$mincost : StructuredVocabulary \longrightarrow (T \times T) \longrightarrow \mathbb{R}$$

$$\forall\, V : StructuredVocabulary \bullet \forall\, x\,,y : V.T \bullet$$
$$x \neq y \;\Rightarrow\; mincost\; V(x\,,y) = min(\{p : seq\, V.T \mid p \in (paths\; V.G(x\,,y)) \bullet cost\; V\; p\}) \;\wedge$$
$$x = y \;\Rightarrow mincost\; V\,(x\,,y) = 0$$

Note the function is qualified so that the minimum cost for a node with respect to itself is zero. This allows a concept name to be included in its own limited cost vocabulary expansion (see below).

## 5.7 Limited Cost Vocabulary Expansion

The minimum path cost function is the basic function used to derive the limited cost expansion of a concept name with respect to a structured vocabulary. Quite simply the limited cost vocabulary expansion of a particular concept name is given by the set of concept names that can be reached within a specified cost limit according to a given path cost function.

Algorithmically the limited cost expansion can be obtained by searching the graph starting from the given node and returning the set of all nodes visited, terminating the search whenever the accumulated cost exceeds the predefined cost limit.

$$
\begin{array}{|l}
limvexp : StructuredVocabulary \rightarrow CNAME \rightarrow \mathbb{P}\, CNAME \\
LIMIT : \mathbb{R} \\
\hline
\forall\, V : StructuredVocabulary \bullet \forall\, t : V.T \bullet \\
\quad limvexp\ V\ t = \{ x : V.T \mid mincost\ V\, (t, x) < LIMIT \}
\end{array}
$$

## 5.8 Normalised Expansion Weight

The general cost functions described above allow a numeric calculation to be made reflecting the cost of traversing a path in the structure graph of a structured vocabulary, in terms of the extent to which relevance is assumed to have diminished. We have assumed that this cost value provides a number that can be used to estimate the probability of relevance of a set of indexed documents with respect to a query, therefore we need to invert and to normalise the cost value to obtain a value between zero and unity for which a lower value is indicative of a lower probability of relevance. The schema below provides a definition of the *expw* function which I refer to as the normalised expansion weight function with respect to a given cost limit.

$$
\begin{array}{|l}
expw : StructuredVocabulary \rightarrow CNAME \times CNAME \rightarrow \mathbb{R} \\
LIMIT : \mathbb{R} \\
\hline
\forall\, V : StructuredVocabulary \bullet \forall\, x, y : V.T \bullet \\
\quad mincost\ V\, (x, y) < LIMIT \Rightarrow expw\ V\, (x, y) = 1 - \dfrac{mincost\ V\, (x, y)}{LIMIT} \quad \wedge \\
\quad mincost\ V\, (x, y) \geq LIMIT \Rightarrow expw\ V\, (x, y) = 0
\end{array}
$$

## 5.9 Limited Cost Index Expansion

A limited cost vocabulary expansion function can be used to derive an expanded index in a manner directly analogous to the use of the broadening vocabulary expansion as given in chapter 3. A limited cost index expansion function is given by the schema below.

$$
\begin{array}{|l}
limfexp : StructuredVocabulary \rightarrow FIELD \rightarrow FIELD \\
limiexp : StructuredVocabulary \rightarrow INDEX \rightarrow INDEX \\
\hline
\forall\, V : StructuredVocabulary ;\ \forall\, I : INDEX ;\ \forall\, F : FIELD \bullet \\
\quad limfexp\ V\ F = \{ d : DNAME ;\ x, t : V.T \mid d \mapsto t \in F \wedge x \in (limvexp\ V\ t) \bullet d \mapsto x \} \quad \wedge \\
\quad limiexp\ V\ I = \{ f : FNAME \mid f \in dom\ I \bullet f \mapsto limfexp\ V\, (I\ f) \}
\end{array}
$$

## 5.10 Index Expansion Weights and Result Scoring

The schema below defines a field expansion weight function *fexpw* that obtains the maximum expansion weight for a particular document name – concept name pair with respect to an unexpanded field.

$$\begin{array}{|l}
fexpw : FIELD \rightarrow (DNAME \times CNAME) \rightarrow \mathbb{R} \\
\hline
\forall F : FIELD ;\ \forall d : DNAME ;\ \forall t : CNAME \bullet \\
\quad fexpw\ F(d, t) = max(\{x : CNAME \mid d \mapsto x \in F \bullet expw(x, t)\})
\end{array}$$

The field expansion weights can be factored into the scoring of results of queries against a limited cost index expansion, as given by the schema below.

$$\begin{array}{|l}
iexpidfscore : INDEX \rightarrow (QUERY \times DNAME) \rightarrow \mathbb{R} \\
\hline
\forall I, I^{\exp} : INDEX ;\ \forall q : QUERY ;\ \forall d : DNAME \bullet \\
\quad I^{\exp} = limiexp\ I \Rightarrow iexpidfscore\ I^{\exp}(q, d) = \\
\sum \{ f : FNAME ; t : CNAME \mid atom(f, t) \in matchingposatoms(q, d) \bullet \\
\qquad (fb\ f) * idf((I\ f), t) * (fexpw(I\ f)(d, t)) \}
\end{array}$$

Note that the *iexpidfscore* sums the product of the field boost weight, the inverse document frequency weight and the field expansion weight over all matching positive query atoms. Note also that both the inverse document frequency weight and the field expansion weight are calculated with respect to the corresponding field in the **unexpanded** index.

## 5.11 Limited Cost Query Expansions

A limited cost vocabulary expansion function can be used to derive an expanded query in a manner directly analogous to the use of the narrowing expansion function as given in chapter 4. A limited cost query expansion function is given by the schema below.

$$\begin{array}{|l}
limqexp : StructuredVocabulary \rightarrow QUERY \rightarrow QUERY \\
\hline
\forall V : StructuredVocabulary ;\ \forall R, O, P : \mathbb{P}\, QUERY \bullet \\
\quad limqexp\ V(rop(R, O, P)) = rop(limqexp\ V \llparenthesis R \rrparenthesis, limqexp\ V \llparenthesis O \rrparenthesis, limqexp\ V \llparenthesis P \rrparenthesis) \\
\forall V : StructuredVocabulary ;\ \forall E : \mathbb{P}\, QUERY \bullet \\
\quad limqexp\ V(and\ E) = and(limqexp\ V \llparenthesis E \rrparenthesis) \ \wedge \\
\quad limqexp\ V(or\ E) = or(limqexp\ V \llparenthesis E \rrparenthesis) \\
\forall V : StructuredVocabulary ;\ \forall e : QUERY \bullet \\
\quad limqexp\ V(not\ e) = not(limqexp\ V\ e) \\
\forall V : StructuredVocabulary ;\ \forall f : FNAME ;\ \forall t : CNAME \bullet \\
\quad limqexp\ V(atom(f, t)) = or(\{x : T \mid x \in (limvexp\ V\ t) \bullet atom(f, x)\})
\end{array}$$

Note that the only fundamental different between this and the naïve expansion of a query lies in the final line of the schema where the limited cost vocabulary expansion function has been applied.

Note also that the parametrisation of the underlying cost function used will necessarily be different for the expansion of an index and the expansion of a query. This is because each strategy is expanding in the opposite direction from the other. I.e. To expand along the narrowing relation for a query is the lowest cost route whereas to expand along the narrowing relation for an index is the highest cost route. I.e. Typically for the expansion of an index the arc weight parameters will be set such that $aw_n > aw_a > aw_b$, and for the expansion of a query the order is reversed.

## 5.12 Query Expansion Weights and Result Scoring

I first define a function *qexpw* to derive the expansion weight of a given field name – concept name pair with respect to an unexpanded query, given in the schema below.

$$qexpw : QUERY \rightarrow (FNAME \times CNAME) \rightarrow \mathbb{R}$$

$$\begin{array}{l} \forall\, q : QUERY\, ;\ \forall\, f : FNAME\, ;\ \forall\, t : CNAME \bullet \\ \quad qexpw\, q\, (f\, , t) = max\, (\{ x : CNAME \mid atom(f\, , x) \in posatoms\, q \bullet expw\, (x\, , t)\}) \end{array}$$

The query expansion weights can then be factored into the scoring of results of expanded queries as defined in the schema below.

$$qexpidfscore : INDEX \rightarrow (QUERY \times DNAME) \rightarrow \mathbb{R}$$

$$\begin{array}{l} \forall\, I : INDEX\, ;\ \forall\, q\, , q^{\exp} : QUERY\, ;\ \forall\, d : DNAME \bullet \\ \quad q^{\exp} = limqexp\, q \Rightarrow qexpidfscore\, I\, (q^{\exp}, d\,) = \\ \sum \{\, f : FNAME\, ; t : CNAME \mid atom\, (f\, , t) \in matchingposatoms\, (q^{\exp}, d\,) \bullet \\ \qquad (fb\, f\,) * idf\, ((I\, f\,), t) * (qexpw\, q\, (f\, , t))\, \} \end{array}$$

Note that the *qexpidfscore* sums the product of the field boost weight, the inverse document frequency weight and the query expansion weight over all matching positive atoms in the **expanded** query.

## 5.13 Summary

This chapter has stated a set of quantitative assumptions that provide a basis for numerical inferences about relative probabilities of relevance to be drawn from the structure of a controlled vocabulary. This in turn supports finer-grained mechanisms for the expansion of either queries or indexes to improve recall and more detailed ranking of results to preserve perceived precision. This chapter has provided a mathematical basis for the implementation of index and query expansion strategies based on these assumptions, which I have called “limited cost expansions”. The final part of this chapter has been concerned with arriving at a numerical estimate for the probability of relevance of members of an expanded result set so that results may be ranked accordingly.

## 5.14 Use Case D - “FACET” System

### *5.14.1 Informal Description*

I refer to the description of the “FACET” system as given in [TUD02].

### *5.14.2 Classification*

**Vocabulary structure**: monohierarchical, associative, fundamental facets.

**Index structure**: either (multiple-field, relational fields, no coordination) or (single-field, relational fields, coordination).

**Query capability**: composite queries (*rop* expressions only).

**Query evaluation strategy**: direct evaluation using limited cost expansion of either index or query with ranking due to expansion weights.

# 6 A Theory of Retrieval Using Structured Vocabularies: Coordination

This chapter extends the theory developed in previous chapters to consider the use of structured vocabularies where the concept names provided by a vocabulary are used in combination to create composite names that have a more specific meaning. I call the act of combining concept names to create new composite names "coordination". Coordination allows a limited number of concept names to be used in more versatile ways and to create a greater range of meaning and degree of specificity. Below I give formal definitions for the construction of coordinations and for their use in indexes and queries. I also give formal definitions for the expansion of indexes and queries involving coordinations, which depends on assumptions regarding the implications of coordination for relevance.

As a hypothetical example of the utility of coordination, consider a document whose subject of discourse is the side-effects of aspirin and the pharmacological action of paracetamol. If we had a controlled vocabulary that captured the four concepts of aspirin, paracetamol, side-effects and pharmacological action we might index this document with all four concept names. Then, however, a composite *and* query for paracetamol and side-effects would spuriously match the given document. If, on the other hand, both the indexer and the searcher were able to indicate that specific combinations were intended via coordinating the appropriate concept names, the occurrence of spurious matches would no longer occur. The ability to provide greater specificity and to avoid this type of spurious match is the primary utility of coordination.

## 6.1 Construction of Coordinations

I define two alternative types of coordination. An ordered coordination is constructed from a **sequence** of concept names, where the sequence of the coordination **is** considered to be important to the meaning of the coordination. An unordered coordination is constructed from a **set** of concept names, where the order of the coordination **is not** considered to be important to the meaning. The *ocoord* function is a constructor function for ordered coordinations and the *ucoord* function is a constructor function for unordered coordinations as specified by the schema below.

$$\begin{array}{l} ocoord : seq\,CNAME \rightarrow CNAME \\ ucoord : \mathbb{P}\,CNAME \rightarrow CNAME \\ \hline \forall\, t : CNAME \bullet \\ \quad ocoord\,\langle t \rangle = t \quad \wedge \\ \quad ucoord\,\{ t \} = t \\ \forall\, S, T : seq\,CNAME \bullet \\ \quad ocoord\; S = ocoord\; T \Leftrightarrow S = T \\ \forall\, S, T : \mathbb{P}\,CNAME \bullet \\ \quad ucoord\; S = ucoord\; T \Leftrightarrow S = T \end{array}$$

Note that I have chosen to treat coordinations as of the same basic type as uncoordinated concept names. This allows coordinations to be used in indexes and in queries without any modifications to the definitions of indexes and queries and query evaluation as given in previous chapters.

## 6.2 Sub-Coordination

In order to establish further definitions below I first define the notions of sub-coordination and of direct sub-coordination. Sub-coordination is a relationship between two coordinations based on their composition as defined by the schema below. Note that I have chosen to use an infix notation – the expression *x subcoordof y* indicates that *x* is a sub-coordination of *y* and the expression *x dirsubcoordof y* indicates that *x* is a direct sub-coordination of *y*. Note also that the definitions below assume that a notion of a subsequence has been defined.

$$\begin{array}{l} \_\,subcoordof\,\_ : CNAME \times CNAME \rightarrow \{ true, false \} \\ \_\,dirsubcoordof\,\_ : CNAME \times CNAME \rightarrow \{ true, false \} \\ \hline \forall\, t : CNAME \bullet \\ \quad t\; subcoordof\; t = true \\ \forall\, S, T : seq\,CNAME \bullet \\ \quad (ocoord\; S)\, subcoordof\, (ocoord\; T) \Leftrightarrow S \text{ is a subsequence of } T \quad \wedge \\ \quad (ocoord\; S)\, dirsubcoordof\, (ocoord\; T) \Leftrightarrow S \text{ is a subsequence of } T \wedge \# S = \# T - 1 \\ \forall\, S, T : \mathbb{P}\,CNAME \bullet \\ \quad (ucoord\; S)\, subcoordof\, (ucoord\; T) \Leftrightarrow S \subset T \\ \quad (ucoord\; S)\, dirsubcoordof\, (ucoord\; T) \Leftrightarrow S \subset T \wedge \# S = \# T - 1 \end{array}$$

## 6.3 Decomposing and Coordinating Expansions and Relevance Assumptions

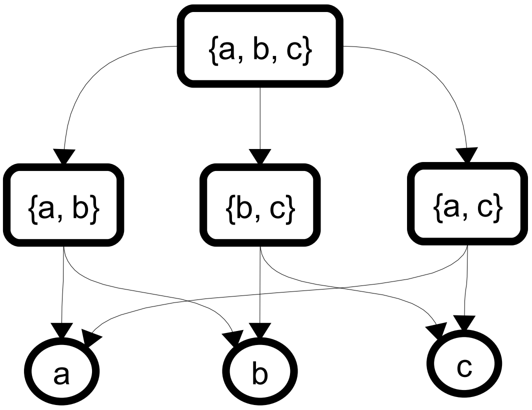
The function *dexp* defined by the schema below expands a coordination to return the set of all sub-coordinations – I refer to this as the decomposing expansion function. The function *cexp* defined below expands a coordination to return the set of all coordinations from a given set for which the first coordination is a sub-coordination – I refer to this as the coordinating expansion function.

$$
\begin{array}{|l}
dexp : CNAME \rightarrow \mathbb{P}\, CNAME \\
cexp : \mathbb{P}\, CNAME \rightarrow CNAME \rightarrow \mathbb{P}\, CNAME \\
dexpgraph : \mathbb{P}\, CNAME \rightarrow (CNAME \leftrightarrow CNAME) \\
\hline
\forall\, t : CNAME \bullet \\
\quad dexp\, t = \{ x : CNAME \mid x\ subcoordof\ t \} \\
\forall\, T : \mathbb{P}\, CNAME ;\ \forall\, t : CNAME \bullet \\
\quad cexp\, T\, t = \{ x : CNAME \mid t\ subcoordof\ x \wedge x \in T \} \\
\forall\, T : \mathbb{P}\, CNAME \bullet \\
\quad dexpgraph\, T = \\
\qquad \{ x, y : CNAME \mid x \in \bigcup dexp(T) \wedge y \in \bigcup dexp(T) \wedge y\ dirsubcoordof\ x \bullet x \mapsto y \}
\end{array}
$$

I state the “naive assumption of coordinating relevance” informally as, “if *x* is a sub-coordination of *y* then **all** documents indexed with y in any given field are relevant to a query for *x* in the same field and **some** documents indexed with *x* in any given field are relevant to a query for *y* in the same field”. The naïve assumption of coordinating relevance has a fundamental asymmetry which is very similar to the naïve assumption of broadening relevance. To explain this assumption with an example, consider that all documents about the side-effects of aspirin could be considered relevant to a query for aspirin, or that all documents about the pharmacological action of paracetamol could be considered relevant to a query for paracetamol.

Note the function *dexpgraph* defined above derives the graph of all direct sub-coordinations for a given set of concept names including coordinations – I call this the decomposition graph for the given set of coordinations. By combining this graph with the structure graph of a controlled vocabulary we may arrive at a quantified assumption of relative relevance that includes coordination – decomposition relationships. This idea is elaborated further in the sections below. The figure directly below illustrates the decomposition graph for an unordered coordination of the concept names *a*, *b* and *c*.

*Figure 11. Example decomposition graph.*



## 6.4 Broadening and Narrowing Expansion of Coordinations

In order to support the naïve expansion of coordinated indexes and queries I extend the definition of the broadening expansion function *bexp* and the narrowing expansion function *nexp* as given below.

$$bexp\,,\,nexp : StructuredVocabulary \rightarrow CNAME \rightarrow \mathbb{P}\,CNAME$$

$$\begin{array}{l}
\forall\, V : Structured\ Vocabulary \bullet \forall\, t_1, \ldots, t_n : V.T \bullet \\
\quad bexp\ V\,(ocoord\,\langle t_1, \ldots, t_n \rangle) = \\
\qquad \{x_1, \ldots, x_n : V.T \mid x_1 \in bexp\ V\ t_1 \wedge \ldots \wedge x_n \in bexp\ V\ t_n \bullet\ ocoord\,\langle x_1, \ldots, x_n \rangle\,\} \ \wedge \\
\quad bexp\ V\,(ucoord\,\{t_1, \ldots, t_n\}) = \\
\qquad \{x_1, \ldots, x_n : V.T \mid x_1 \in bexp\ V\ t_1 \wedge \ldots \wedge x_n \in bexp\ V\ t_n \bullet\ ucoord\,\{x_1, \ldots, x_n\}\,\} \ \wedge \\
\quad nexp\ V\,(ocoord\,\langle t_1, \ldots, t_n \rangle) = \\
\qquad \{x_1, \ldots, x_n : V.T \mid x_1 \in nexp\ V\ t_1 \wedge \ldots \wedge x_n \in nexp\ V\ t_n \bullet\ ocoord\,\langle x_1, \ldots, x_n \rangle\,\} \ \wedge \\
\quad nexp\ V\,(ucoord\,\{t_1, \ldots, t_n\}) = \\
\qquad \{x_1, \ldots, x_n : V.T \mid x_1 \in nexp\ V\ t_1 \wedge \ldots \wedge x_n \in nexp\ V\ t_n \bullet\ ucoord\,\{x_1, \ldots, x_n\}\,\} \ \wedge
\end{array}$$

## 6.5 Broadening/Narrowing/Associating Relations for Coordinations

Relationships between coordinations may be derived from relationships between their constituents. Below I define the *drel* function which evaluates as *true* if and only if two coordinations may be directly related via one of their constituents with respect to some structure relation and are otherwise identical.

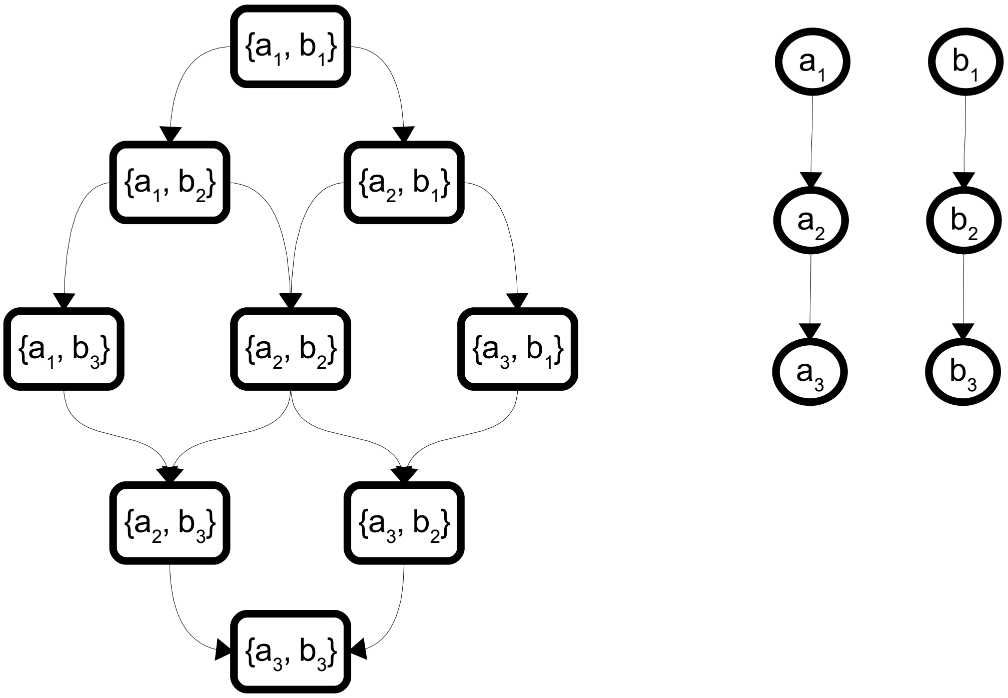
$$drel : (CNAME \leftrightarrow CNAME) \rightarrow (CNAME \times CNAME) \rightarrow \{true\,,\, false\}$$

$$\begin{array}{l}
\forall\, R : CNAME \leftrightarrow CNAME\,;\ \forall\, x\,,\, y : CNAME \bullet \\
\quad drel\,R\,(x\,,\, y) \Leftrightarrow x \mapsto y \in R \\
\forall\, R : CNAME \leftrightarrow CNAME\,;\ \forall\, S\,,\, T : \mathbb{P}\,CNAME \bullet \\
\quad drel\,R\,(ucoord\,S\,,\, ucoord\,T\,) \Leftrightarrow \exists\, s : S\,; \exists\, t : T \bullet s \mapsto t \in R \wedge S - s = T - t \\
\forall\, R : CNAME \leftrightarrow CNAME\,;\ \forall\, S\,,\, T : seq\,CNAME \bullet \\
\quad drel\,R\,(ocoord\,S\,,\, ocoord\,T\,) \Leftrightarrow \\
\qquad \exists\, i\,,\, j : \mathbb{Z} \bullet S\,i \mapsto T\,j \in R \wedge squash(S \setminus \{i \mapsto S\,i\}) = squash(T \setminus \{j \mapsto T\,j\})
\end{array}$$

Note that the definition for ordered coordinations depends on a *squash* function which "squashes" a sequence with missing values into a continuous sequence.

By applying the *drel* function a graph may obtained for any coordination with respect to any structure relation of a controlled vocabulary. The figure below depicts the broadening graph for an unordered coordination of the concept names $a_1$ and $b_1$ with respect to the broadening relation $\{a_1 \mapsto a_2, a_2 \mapsto a_3, b_1 \mapsto b_2, b_2 \mapsto b_3\}$ .

*Figure 12. Depiction of the broadening graph for an unordered coordination.*



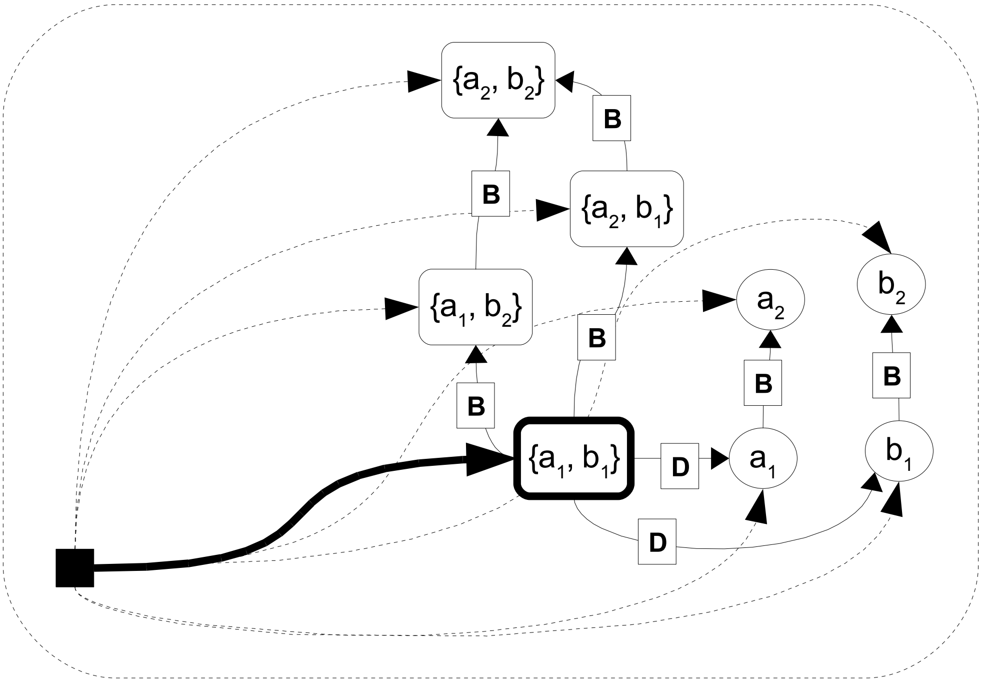
## 6.6 Naïve Expansion of a Coordinated Field

The naïve assumption of coordinating relevance justifies the expansion of the fields of an index as given by the following definitions, where *cfexp* is the coordinated field expansion function and *ciexp* is the coordinated index expansion function.

$$
\begin{array}{|l}
cfexp : StructuredVocabulary \longrightarrow FIELD \longrightarrow FIELD \\
ciexp : StructuredVocabulary \longrightarrow INDEX \longrightarrow INDEX \\
\hline
\forall\, V : StructuredVocabulary;\ \forall\, I : INDEX;\ \forall\, F : FIELD \bullet \\
\quad cfexp\ V\ F = \{ d : DNAME; x, t : CNAME \mid d \mapsto t \in F \land x \in \bigcup (bexp\ V\ (\!| dexp\ t |\!)) \bullet d \mapsto x \} \land \\
\quad ciexp\ V\ I = \{ f : FNAME \mid f \in dom\ I \bullet f \mapsto cfexp\ V\ (I\ f) \}
\end{array}
$$

Note that the essence of the expansion is that all concept names in the index are first decomposed and then broadened. The figure below illustrates the naïve expansion of a coordinated field $\{d \mapsto ucoord\{a_1, b_1\}\}$ given a broadening relation $\{a_1 \mapsto a_2, b_1 \mapsto b_2\}$, where broadening relationships are labeled “B”, decomposing relationships are labeled “D” and links derived from the expansion of the field are shown as dashed arcs.

*Figure 13. Depiction of naïve expansion of coordinated field.*



## 6.7 Naïve Expansion of a Coordinated Query

The naïve assumption of coordinating relevance justifies the expansion of queries as defined by the following schema, where *cqexp* is the naïve coordinated query expansion function.

$$cqexp : (StructuredVocabulary \times INDEX) \rightarrow QUERY \rightarrow QUERY$$

$$\begin{array}{l}
\forall V : StructuredVocabulary;\ \forall I : INDEX;\ \forall R, O, P : \mathbb{P}\,QUERY \bullet \\
\quad cqexp(V, I)(rop(R, O, P)) = \\
\qquad rop(cqexp(V, I)\llparenthesis R \rrparenthesis, cqexp(V, I)\llparenthesis O \rrparenthesis, cqexp(V, I)\llparenthesis P \rrparenthesis) \\
\forall V : StructuredVocabulary;\ \forall I : INDEX;\ \forall E : \mathbb{P}\,QUERY \bullet \\
\quad cqexp(V, I)(and\ E) = and(cqexp(V, I)\llparenthesis E \rrparenthesis) \ \wedge \\
\quad cqexp(V, I)(or\ E) = or(cqexp(V, I)\llparenthesis E \rrparenthesis) \\
\forall V : StructuredVocabulary;\ \forall I : INDEX;\ \forall e : QUERY \bullet \\
\quad cqexp(V, I)(not\ e) = not(cqexp(V, I) e) \\
\forall V : StructuredVocabulary;\ \forall I : INDEX;\ \forall f : FNAME;\ \forall n : CNAME \bullet \\
\quad cqexp(V, I)(atom(f, n)) = \\
\qquad or(\{x : CNAME \mid x \in \bigcup(nexp\ V \llparenthesis cexp(ran(I\ f))\ n \rrparenthesis) \bullet atom(f, x)\})
\end{array}$$

Note that, whereas the naïve expansion of a coordinated field involves first decomposing then broadening all names, the naïve expansion of a coordinated query involves first composing then narrowing all names. Note also that the necessity to find the coordinating expansion with respect to the range of the relevant field (and not with respect to the universe of possible coordinations – this would lead to unnecessary combinatorial explosion) requires that the coordinated query expansion function take a different form from the previously defined naïve query expansion function *qexp* (which can be applied independently from any particular index).

## 6.8 Limited Cost Expansions Including Coordinations

The previous chapter derived formulas for obtaining a limited cost vocabulary expansion for a primitive (uncoordinated) concept name with respect to the structure graph of a controlled vocabulary. In this section I consider how to extend this notion to include a set of coordinations. I.e. The definitions in this section allow the limited cost expansion of any coordination to be computed with respect to a structured vocabulary and allow the limited cost expansion of any primitive concept name to include a set of coordinations drawn from a coordinated field.

Given a structured vocabulary $V$ and a coordinated field $F$ indexed with coordinations of $V$, we can extend the structure graph of $V$ to include those coordinations found in $F$. This can be done by including the decomposition graphs for all coordinations and then deriving broadening graphs for all coordinations by applying the direct relation test as defined earlier in this chapter. The extended structure graph can then be used to derive a limited cost vocabulary expansion, which can in turn be used to derive and to weight expanded queries and indexes in the same way as described in the previous chapter.

## 6.9 Summary

This chapter has considered the coordination of concept names to produce new composite names for use in indexes and queries. Two types of coordination were defined, being unordered and ordered coordinations. These two types are considered because unordered coordinations may provide a compromise between the high specificity (and therefore difficulty of creating a matching query) of ordered coordinations and the lack of specificity and possibility for spurious results found when no coordinations are used. Functions were defined for the decomposition and for the broadening (or narrowing) of coordinations, in addition to the decomposing, coordinating, broadening and narrowing expansion of coordinations. Finally the naïve and limited cost expansions of coordinated indexes and queries were considered.

## 6.10 Use Case E - "When Lion Could Fly"

### *6.10.1 Informal Description*

A library uses a subject heading system to organise the books it holds. Each book is classified under a single subject heading.

The subject heading system consists of a hierarchy of main headings, and a list of subdividing headings. The hierarchy of main is as follows.

```
animals
  lions
```

This list of subdividing headings is as follows.

```
mythology
behaviour
```

To create a subject heading for a book, one of the main headings is combined with one of the subdividing headings.

The table below shows the books held by the library, and the subject heading for each.

| Book Title | Subject Heading |
|---|---|
| The Serengeti Lion: Study of Predator-Prey Relations | lions - behaviour |
| When Lion Could Fly: And Other Tales from Africa | lions - mythology |

The search system allows users to search using just the main subject headings. The system also allows users to construct a subject heading on the fly by combining main headings with subdividing headings, and request matching items.

Jake first searches using the "animals" heading, and is returned two items, being the two books listed above.

Jake then combines "animals" with "mythology", and is returned a single matching item, being the book entitled "When Lion Could Fly: And Other Tales from Africa". Jake then combines "animals" with "behaviour", and is returned a single item, being the book entitled "The Serengeti Lion: Study of Predator-Prey Relations".

### *6.10.2 Classification*

**Vocabulary structure**: monohierarchical.

**Index structure**: single-field, functional fields, ordered coordination.

**Query capability**: atomic queries with coordination.

**Query evaluation strategy**: direct evaluation using naïve expansion of either index or query.

# 7 A Theory of Retrieval Using Structured Vocabularies: Translation

This chapter develops a theory for the general situation where an index expressed in terms of one controlled vocabulary must be queried in terms of another controlled vocabulary. In these circumstances a mapping is required between the respective vocabularies, in order that either queries or indexes may be translated appropriately. The primary consideration is achieving a translation that does not degrade the performance of retrieval applications, i.e. that preserves both recall and precision as far as possible.

Two alternative methods are developed for expressing a mapping between two controlled vocabularies, which I call “structural mapping” and “query expression mapping” respectively. A theoretical framework is developed for the different ways in which either queries or indexes may be translated according to each mapping method, with particular attention paid to the consequences for recall and precision of different translation strategies.

Note that a theory is provided for translation of both queries and indexes, because as with expansion strategies, although translation of an index is mathematically equivalent to translation of a query, these are different computational strategies and hence worth exploring.

## 7.1 Structural Mapping

I define a structural mapping from a source vocabulary to a target vocabulary as fundamentally consisting of a set of binary relations between the concept names of the source vocabulary and the concept names of the target vocabulary – I call these “mapping relations”. Specifically I define four types of mapping relations, being the (exact) equivalence mapping, the broadening mapping, the narrowing mapping, and the associating mapping. As with the structure relations of similar names I make no attempt to provide an independent definition of the intended meaning of these relations, rather I define their semantics in terms of the translation operations that they may reasonably be used to achieve and the underlying assumptions about relevance that justify these translations and their purported consequences.

StructuralMapping

$V_{source}, V_{target} : StructuredVocabulary$

$equivalent, broader, narrower, associated, G : V_{source}.T \leftrightarrow V_{target}.T$

$etrans, btrans, ntrans, atrans : V_{source}.T \rightarrow \mathbb{P}\, V_{target}.T$

$G = equivalent \cup broader \cup narrower \cup associated$

$\forall\, t : CNAME \bullet$

$\quad etrans\ t = equivalent \llparenthesis \{t\} \rrparenthesis \;\wedge$

$\quad btrans\ t = broader \llparenthesis \{t\} \rrparenthesis \;\wedge$

$\quad ntrans\ t = narrower \llparenthesis \{t\} \rrparenthesis \;\wedge$

$\quad atrans\ t = associated\ t \llparenthesis \{t\} \rrparenthesis \;\wedge$

Note that I have included for convenience a set of vocabulary translation functions *etrans, btrans, ntrans* and *atrans* defined on the mapping relations *equivalent, broader, narrower* and *associated* respectively.

## 7.2 Naïve Query Translation Using a Structural Mapping

The basic philosophy of the naïve query translation is that the query atoms are translated by an equivalence translation if available, otherwise they are translated by either narrowing or broadening translations according to whether or not they make a positive contribution to the query in order that recall may be preserved (or by inverting the formulas in order that precision may be preserved). To preserve recall, positive atoms are translated by their broadening translations and negative atoms are translated by their narrowing translations. This is the basis for the *posqtrans* and *negqtrans* query translation functions defined by the schema below. To preserve precision the reverse may be applied by modifying the formulas accordingly. Note that the justification for this translation method is derived directly from the naïve assumptions of broadening and narrowing relevance as applied to the broadening and narrowing mapping relations.

The schema is rather long winded because it must first define how the translation of queries is propagated through arbitrarily nested composite query expressions. The propositions of most interest are those concerning the translation of query atoms, which are given last.

$$
\begin{array}{l}
posqtrans\,,\, negqtrans : StructuralMapping \rightarrow QUERY \rightarrow QUERY \\
\hline
\forall\, M : StructuralMapping;\ \forall\, R, O, P : \mathbb{P}\, QUERY\ \bullet \\
\quad posqtrans\, M\, (rop(R, O, P)) = \\
\qquad rop(\, posqtrans\, M\, \llparenthesis\, R\, \rrparenthesis,\, posqtrans\, M\, \llparenthesis\, O\, \rrparenthesis,\, posqtrans\, M\, \llparenthesis\, P\, \rrparenthesis) \ \wedge \\
\quad negqtrans\, M\, (rop(R, O, P)) = \\
\qquad rop(negqtrans\, M\, \llparenthesis\, R\, \rrparenthesis,\, negqtrans\, M\, \llparenthesis\, O\, \rrparenthesis,\, negqtrans\, M\, \llparenthesis\, P\, \rrparenthesis) \ \wedge \\
\forall\, M : StructuralMapping;\ \forall\, E : \mathbb{P}\, QUERY\ \bullet \\
\quad posqtrans\, M\, (and\ E) = and\, (\, posqtrans\, M\, \llparenthesis\, E\, \rrparenthesis) \ \wedge \\
\quad negqtrans\, M\, (and\ E) = and\, (\, negqtrans\, M\, \llparenthesis\, E\, \rrparenthesis) \ \wedge \\
\quad posqtrans\, M\, (or\ E) = or\, (\, posqtrans\, M\, \llparenthesis\, E\, \rrparenthesis) \ \wedge \\
\quad negqtrans\, M\, (or\ E) = or\, (negqtrans\, M\, \llparenthesis\, E\, \rrparenthesis) \\
\forall\, M : StructuralMapping;\ \forall\, e : QUERY\ \bullet \\
\quad posqtrans\, M\, (not\ e) = not\, (negqtrans\, M\ e) \ \wedge \\
\quad negqtrans\, M\, (not\ e) = not\, (\, posqtrans\, M\ e) \\
\forall\, M : StructuralMapping;\ \forall\, f : FNAME;\ \forall\, t : CNAME\ \bullet \\
\quad M.etrans\, t \neq \varnothing \Rightarrow \\
\qquad posqtrans\, M\, (atom(\, f\,, t)) = negqtrans\, M\, (atom(\, f\,, t)) = \\
\qquad\quad or(\{x : CNAME \mid x \in (M.etrans\, t) \bullet atom(\, f\,, x)\}) \\
\forall\, M : StructuralMapping;\ \forall\, f : FNAME;\ \forall\, t : CNAME\ \bullet \\
\quad M.etrans\, t = \varnothing \Rightarrow \\
\qquad posqtrans\, M\, (atom(\, f\,, t)) = or(\{x : CNAME \mid x \in (M.btrans\, t) \bullet atom(\, f\,, x)\}) \ \wedge \\
\qquad negqtrans\, M\, (atom(\, f\,, t)) = or(\{x : CNAME \mid x \in (M.ntrans\, t) \bullet atom\, (\, f\,, x)\})
\end{array}
$$

## 7.3 Naïve Index Translation Using a Structural Mapping

In the case of index translation using a structural mapping we are not able to make an absolute choice as to whether to consistently preserve recall or precision because there is no way of knowing *a priori* whether matches will be positive or negative. Assuming that the majority of query atoms will be positive atoms, a choice may then be made as to whether to preserve precision or recall in this majority case. To preserve recall, if an equivalence translation is not available, the narrowing translation must be used; to preserve precision the broadening translation is used (this is the reverse of the rule for query translation as would be expected). To be consistent with the above schema for query translation to preserve recall, the schema below defines a translation function for the naïve translation of fields and indexes in order to preserve recall in the anticipated majority of situations being queries with positive atoms. Therefore in the case where an equivalence translation is not available the narrowing translation is applied. To instead preserve precision instead of recall in the majority of cases the formulas below can be modified to use the broadening translation accordingly.

$$
\begin{array}{|l}
ftrans : StructuralMapping \rightarrow FIELD \rightarrow FIELD \\
itrans : StructuralMapping \rightarrow INDEX \rightarrow INDEX \\
\hline
\forall\, M : StructuralMapping ;\ \forall\, F : FIELD \bullet \\
\quad ftrans\, M\ F = \\
\{ d : DNAME ;\ x , t : CNAME | d \mapsto t \in F \land x \in ( M.etrans\, t ) \bullet d \mapsto x \} \cup \\
\{ d : DNAME ;\ x , t : CNAME | d \mapsto t \in F \land M.etrans\, t = \emptyset \land x \in ( M.ntrans\ t ) \bullet d \mapsto x \} \\
\forall\, M : StructuralMapping ;\ \forall\, I : INDEX \bullet \\
\quad itrans\, M\ I = \{ f : FNAME \,|\, f \in ( dom\, I ) \bullet f \mapsto ftrans\, M\ ( I\ f ) \}
\end{array}
$$

## 7.4 Limited Cost Translations Using a Structural Mapping

By combining the broadening, narrowing, and associating mapping relations of a structural mapping with the broadening, narrowing and associating relations of the target vocabulary and by additionally considering the equivalence mappings with an appropriate arc weight approaching unity, the limited cost vocabulary expansion function defined in chapter 5 can be adapted to provide a limited cost vocabulary translation function. Additionally the normalised expansion weight function can be adapted to provide instead a “normalised translation weight” that models the relative probability of relevance for different translations. In this way, all links available via the mapping relations of a structural mapping can be exploited to achieve a translation of either index or query with maximum preservation of recall, whilst applying the translation weights to achieve effective ranking and hence also maintain perceived precision. A full elaboration of this idea is beyond the scope of the current report - this is an interesting area for further theoretical and empirical investigation.

## 7.5 Query Expression Mapping

I define a query expression mapping from a source vocabulary to a target vocabulary as a set of functions mapping the concept names of the source vocabulary directly onto query templates expressed in terms of the target vocabulary.

Query templates are the same as proper queries, except that none of the query atoms include a field name – the field name is introduced later by means of a field name substitution function. The schema below defines a new atomic query expression constructor function *temp* for template atoms. The schema below also defines the field name substitution function *fsub* for obtaining a proper query from a query template.

$$\begin{array}{|l}
temp : CNAME \rightarrow QUERY \\
fsub : FIELD \rightarrow QUERY \rightarrow QUERY \\
\hline
\forall\, f : FNAME;\ \forall\, R, O, P : \mathbb{P}\, QUERY \bullet \\
\quad fsub\ f\,(rop(R, O, P)) = rop(fsub\ f\,(\!| R |\!),\ fsub\ f\,(\!| O |\!),\ fsub\ f\,(\!| P |\!)) \\
\forall\, f : FNAME;\ \forall\, E : \mathbb{P}\, QUERY \bullet \\
\quad fsub\ f\,(and\ E) = and\,(fsub\ f\,(\!| E |\!))\ \wedge \\
\quad fsub\ f\,(or\ E) = or\,(fsub\ f\,(\!| E |\!)) \\
\forall\, f : FNAME;\ \forall\, e : QUERY \bullet \\
\quad fsub\ f\,(not\ e) = not\,(fsub\ f\ e) \\
\forall\, f, g : FNAME;\ \forall\, t : CNAME \bullet \\
\quad fsub\ f\,(atom(g, t)) = atom(g, t)\ \wedge \\
\quad fsub\ f\,(temp\ t) = atom(f, t)
\end{array}$$

I now define a query expression mapping as given in the schema below.

QueryExpressionMapping

$$\begin{array}{|l}
V_{source}, V_{target} : StructuredVocabulary \\
etrans, btrans, ntrans : V_{source}.T \nrightarrow QUERY \\
\hline
\end{array}$$

The functions *eqtrans*, *bqtrans* and *nqtrans* I refer to as the equivalence, broadening and narrowing query mapping translation functions respectively.

## 7.6 Naïve Query Translation Using a Query Expression Mapping

The schema below defines the functions *posqqtrans* and *negqqtrans* as the naïve query translation functions for query expression mappings. The assumptions for their application are the same as for the naïve translation of queries using a structural mapping.

As before, most of the schema is concerned with defining how the query translation propagates through arbitrarily nested composite query expressions. The last part of the schema defines the translation of query atoms according to the query expression mapping.

$$
\begin{array}{l}
posqqtrans, negqqtrans : QueryExpressionMapping \rightarrow QUERY \rightarrow QUERY \\
\hline
\forall M : QueryExpressionMapping;\ \forall R, O, P : \mathbb{P}\, QUERY \bullet \\
\quad posqqtrans\ M\ (rop(R, O, P)) = \\
\qquad rop(\ posqqtrans\ M\ \llparenthesis\ R\ \rrparenthesis, posqqtrans\ M\ \llparenthesis\ O\ \rrparenthesis, posqqtrans\ M\ \llparenthesis\ P\ \rrparenthesis)\ \wedge \\
\quad negqqtrans\ M\ (rop(R, O, P)) = \\
\qquad rop(negqqtrans\ M\ \llparenthesis\ R\ \rrparenthesis, negqqtrans\ M\ \llparenthesis\ O\ \rrparenthesis, negqqtrans\ M\ \llparenthesis\ P\ \rrparenthesis)\ \wedge \\
\forall M : QueryExpressionMapping;\ \forall E : \mathbb{P}\, QUERY \bullet \\
\quad posqqtrans\ M\ (and\ E) = and\ (\ posqqtrans\ M\ \llparenthesis\ E\ \rrparenthesis)\ \wedge \\
\quad negqqtrans\ M\ (and\ E) = and\ (negqqtrans\ M\ \llparenthesis\ E\ \rrparenthesis)\ \wedge \\
\quad posqqtrans\ M\ (or\ E) = or\ (posqqtrans\ M\ \llparenthesis\ E\ \rrparenthesis)\ \wedge \\
\quad negqqtrans\ M\ (or\ E) = or\ (negqqtrans\ M\ \llparenthesis\ E\ \rrparenthesis) \\
\forall M : QueryExpressionMapping;\ \forall e : QUERY \bullet \\
\quad posqqtrans\ M\ (not\ e) = not\ (negqqtrans\ M\ e)\ \wedge \\
\quad negqqtrans\ M\ (not\ e) = not\ (posqqtrans\ M\ e) \\
\forall M : QueryExpressionMapping;\ \forall f : FNAME;\ \forall t : CNAME \bullet \\
\quad t \in dom\ M.etrans \Rightarrow \\
\qquad posqqtrans\ M\ (atom(f, t)) = negqqtrans\ M\ (atom(f, t)) = fsub\ f\ (M.etrans\ t) \\
\forall M : QueryExpressionMapping;\ \forall f : FNAME;\ \forall t : CNAME \bullet \\
\quad t \notin dom\ M.etrans \Rightarrow \\
\qquad posqqtrans\ M\ (atom(f, t)) = fsub\ f\ (M.btrans\ t)\ \wedge \\
\qquad negqtrans\ M\ (atom(f, t)) = fsub\ f\ (M.ntrans\ t)
\end{array}
$$

## 7.7 Naïve Index Translation Using a Query Expression Mapping

A query expression mapping can be used to translate an index, but in a slightly unintuitive way. By matching the query templates in the range of the mapping against the field to be translated, matching documents may be "mapped backwards" to the domain of the mapping. I.e. given a query expression mapping for source and target vocabularies it is possible to translate an index expressed in terms of the *target* vocabulary into an index expressed in terms of the *source* vocabulary. The index and field translation function defined by the schema below illustrate this principle.

$$
\begin{array}{l}
fqtrans : QueryExpressionMapping \rightarrow FIELD \rightarrow FIELD \\
iqtrans : QueryExpressionMapping \rightarrow INDEX \rightarrow INDEX \\
\hline
\forall M : QueryExpressionMapping;\ \forall I : INDEX;\ \forall f : FNAME \bullet \\
\quad fqtrans\ M\ (I\ f) = \\
\{d : DNAME;\ t : CNAME \mid d \in results\ I\ (fsub\ f\ (M.etrans\ t)) \bullet d \mapsto t\} \cup \\
\{d : DNAME;\ t : CNAME \mid t \notin dom\ M.etrans \wedge d \in results\ I\ (fsub\ f\ (M.btrans\ t)) \bullet d \mapsto x\} \\
\\
\forall M : QueryExpressionMapping;\ \forall I : INDEX \bullet \\
\quad iqtrans\ M\ I = \{f : FNAME \mid f \in (dom\ I) \bullet f \mapsto fqtrans\ M\ (I\ f)\}
\end{array}
$$

As with the previous analysis of naïve index translation the theory above favours the preservation of recall at the expense of precision by applying the broadening translation (but remember operating in reverse so effectively the narrowing translation) whenever an equivalent translation is not available.

## 7.8 Summary

This chapter has applied the theory developed in previous chapters to the problem of evaluating queries expressed in terms of one controlled vocabulary against an index expressed in terms of another controlled vocabulary. Two types of mapping were defined, being the structural mapping and the query expression mapping methods. Naïve translations of both queries and indexes were defined for both types of mapping. A method for using the limited cost expansion functions to achieve limited cost translations and translation weights was suggested, although the theoretical foundation of this strategy has been left to future work.

## 7.9 Use Case F - "AQUARELLE" Project

### *7.9.1 Informal Description*

I refer to the examples of mapping and translation given in [DOE01].

### *7.9.2 Classification*

**Vocabulary Structure**: hierarchical and associative.

**Mapping Types**: Both structural mapping and query expression mapping.

**Translation Strategies**: Naïve query translation.

# 8 RDF Representations

This chapter describes alternative design patterns for the mapping of the data structures defined in chapters 3-7 of this report (hereafter "the theory") to RDF graphs. I.e. This chapter suggests ways in which "structured vocabularies", "indexes", "queries" and "vocabulary mappings" may be represented using RDF. In all cases the assumptions of the theory with respect to the retrieval operations that may reasonably be performed on these data structures  have a direct bearing on the application-level semantics of the RDF vocabularies chosen as the building blocks of an RDF representation. The implications of these assumptions are discussed, particularly in relation to the challenge of providing precise operational definitions for the classes and properties of the SKOS vocabularies [MIL05A] [MIL05B].

This chapter also describes strategies for deriving the same data structures from existing RDF graphs. In these cases the operational semantics of the RDF vocabularies deployed in these graphs must be consistent with the assumptions of the theory and this chapter discusses situations where this is likely to be the case.

In this chapter URI references are given in the text as qualified names using the following prefix abbreviations – x: <http://www.example.com/retrieval-schema#>, eg: <http://www.example.com/examples#>, skos: <http://www.w3.org/2004/02/skos/core#>, rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, owl: <http://www.w3.org/2002/07/owl#> – i.e. "x:foo" is equivalent to the URI reference <http://www.example.com/retrieval-schema#foo>.

## 8.1 Representing Binary Relations

The building blocks of a "structured vocabulary" as defined in chapter 3 are a set of "concept names" and one or more binary relations on this set. The building blocks of an "index" as defined in chapter 3 are a set of "field names", and a set of binary relations (which I have called "fields") between a set of "document names" and a set of "concept names". The building blocks of a "structural mapping" as defined in Chapter 7 are two sets of "concept names" (the "source" and the "target") and one or more binary relations between the "source" and the "target" sets.

Before considering appropriate RDF vocabularies for the representation of specific data structures, I first consider two basic alternatives for the representation of a generic binary relation in RDF. Let *R* be a binary relation between the sets *X* and *Y*. Let $V_x$ be a set of URI references denoting the objects in *X*, let $V_y$ be a set of URI references denoting the objects in *Y*, let *I* be the simple interpretation of these vocabularies and let *IEXT* be the extension mapping of *I* [HAY04].

The simplest way of representing a binary relation as an RDF graph is as the extension of a single property. I.e. Given a URI reference *p*, construct a set of RDF triples $\{s:V_x;\, o:V_y \mid (I(s), I(o)) \in R \bullet (s, p, o)\}$ . *R* is then equal to $IEXT(I(p))$ , i.e. *R* is given by the property extension of *p*.

As an example of this pattern, consider a set of objects $\{a, b, c\}$ , a set of URI references $\{x, y, z\}$ and an interpretation *I* such that $I(x)=a, I(y)=b, I(z)=c$ . The binary relation $\{a \mapsto b, a \mapsto c\}$ may then be represented by the RDF graph $\{(x, p, y), (x, p, z)\}$ where *p* is a URI reference chosen to denote the relation.

An alternative pattern for representing a binary relation as an RDF graph is using the n-ary relations pattern described in [NOY06]. In this case we require a URI reference *C* and two further URI references *p1* and *p2*. A representation of the binary relation *R* is then given by constructing a set of RDF triples

$\cup\{x:V_x;\ y:V_y \mid (I(x), I(y)) \in R \bullet \{(blank, p1, x), (blank, p2, y), (blank, type, C)\}\}$

where *blank* is a new blank node identifier created for each binding of the variables *x* and *y* and *type* is the rdf:type URI reference. *R* is then represented by the class extension of *C*, with the property extensions of *p1* and *p2* giving the *first* and *second* projection functions for each member of *R*.

As an example of this pattern, the relation $\{a \mapsto b, a \mapsto c\}$ as used in the previous example may be represented by the RDF graph

$\{(b1, p1, x), (b1, p2, y), (b1, type, C), (b2, p1, x), (b2, p2, z), (b2, type, C)\}$ where *b1* and *b2* are blank node identifiers and *C*, *p1* and *p2* are URI references chosen to denote the relation.

Binary relations may of course be represented by more complicated RDF graph structures. However, below I consider only the representation of a binary relation as the extension of a single property of an RDF graph, being the simplest sufficient representation pattern.

## 8.2 Representing a Structured Vocabulary

As mentioned above, according to the theory a “structured vocabulary” is characterised by a set of “concept names” and one or more binary relations on this set. Specifically, the theory considers only those types of structured vocabulary for which three relations are defined, which I have called *broader*, *narrower* and *associated*. No attempt was made to provide an independent definition of the meaning of these relations, rather their meaning was defined entirely in terms of assumptions regarding relevance and recall that could reasonably be used to justify certain retrieval operations.

The notion of a “concept name” (*CNAME*)was introduced as a given set and no restrictions were placed on its membership. In this section I consider options for constructing an RDF representation of a structured vocabulary by first assuming that all “concept names” are URI references. All that is then required for the RDF representation of a structured vocabulary is a URI reference for each of the three relations, which I give here using a temporary namespace as x:broader, x:narrower and x:associated corresponding to the structure relations with similar names.

Given then a structured vocabulary *V* defined by set of concept names *T* and binary relations *broader*, *narrower* and *associated*, an RDF representation of *V* may be given by constructing an RDF graph *G* where, under the simple Herbrand interpretation of *G* (where the interpretation of each URI reference is the URI reference itself [HAY04]), the property extension of eg:broader gives the *broader* relation, the property extension of eg:narrower gives the *narrower* relation and the property extension of eg:associated gives the *associated* relation.

For example, let *V* be a structured vocabulary defined by the set of URI references $T=\{a,b,c\}$ (where *a* is the URI reference eg:a, *b* is eg:b and *c* is eg:c) and the binary relations $broader=\{a\mapsto b\}$ , $narrower=\{b\mapsto a\}$ and $associated=\{a\mapsto c, c\mapsto a\}$ . The RDF representation of *V* constructed by the method described above is then given by the RDF graph shown below in the Turtle syntax.

```
@prefix x: <http://www.example.com/retrieval-schema#> .
@prefix : <http://www.example.com/examples#> .

:a x:broader :b.
:b x:narrower :a.
:a x:associated :c.
:c x:associated :a.
```

What semantics would be appropriate for the x:broader, x:narrower and x:associated properties?

At the logical level, the theory assumed that the *broader* and *narrower* relations are inverse relations. Therefore it would be appropriate to declare that the x:broader and x:narrower properties were inverse properties. This could be done either by declaring an appropriate pair of rules, or by providing an appropriate OWL description.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix owl: <http://www.w3.org/@@TODO>.

x:broader a owl:ObjectProperty.
x:narrower a owl:ObjectProperty; owl:inverseOf x:broader.
```

The theory assumed that the *associated* relation was symmetric and therefore it would be appropriate to declare that the x:associated property is a symmetric property. Again this could either be done with a rule, or by providing an appropriate OWL description.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix owl: <http://www.w3.org/@@TODO>.

x:associated a owl:SymmetricProperty.
```

The theory also assumed that the *broader* relation defined a graph with no cycles, which is equivalent to the statement that the transitive closure of the *broader* relation is irreflexive. Because there is no notion of inconsistency at the level of the RDF semantics it is not possible to declare this characteristic for the x:broader property using RDF alone. The OWL semantics do have a notion of inconsistency, however there is no support for declaring irreflexive properties. Such an extension would be relatively trivial and the OWL declaration given below uses a hypothetical extension to declare this semantics for the x:broader property. The x:narrower property also acquires similar semantics by virtue of the fact that it has been declared as the inverse of x:broader above.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix owl: <http://www.w3.org/@@TODO>.
@prefix owl-x: <http://www.example.org/owl-extensions#>.

x:broader a owl:TransitiveProperty, owl-x:IrreflexiveProperty.
```

Finally, regarding the logical characteristics assumed for the *broader*, *narrower* and *associated* relations, these relations were assumed to be all pairwise disjoint. Again, there is no support for declaring the disjointness of properties in OWL, although such an extension would be relatively trivial. The declaration below assumes a hypothetical OWL extension for declaring the disjointness of properties.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix owl: <http://www.w3.org/@@TODO>.
@prefix owl-x: <http://www.example.org/owl-extensions#>.

x:broader owl-x:disjointWithProperty x:narrower, x:associated.
x:narrower owl-x:disjointWithProperty x:associated.
```

What about the application-level semantics for these properties? I.e. What operational semantics would be consistent with the assumptions of broadening and associative relevance, which the *broader*, *narrower* and *associated* relations are assumed to justify and hence which the x:broader, x:narrower and x:associated properties are intended to imply?

Generally speaking, when using an RDF vocabulary for the representation of a particular data structure, the vocabulary should carry no more semantics than are implied by the original data structures. Therefore, the application-level semantics of the x:broader property are simply that the use of this property implies that broadening relevance may be assumed naïvely to a first approximation and quantitatively with an appropriate set of parameter values – and nothing more. I.e. The naive expansion of an index may reasonably be derived from the extension of the x:broader property and similarly the naive expansion of a query may reasonably be derived from the extension of the x:narrower property. Similarly, the limited cost expansion of either indexes or queries may be derived from the graph of the x:broader, x:narrower and x:associated properties, with a weighting of these properties appropriate to the quantitative assumptions of relative relevance.

The Simple Knowledge Organisation System (SKOS) Core Vocabulary [MIL05A] [MIL05B] defines three properties skos:broader, skos:narrower and skos:related. The logical characteristics of these properties have been defined to be similar to those declared above for the x:broader, x:narrower and x:associated properties – skos:broader and skos:narrower are declared to be each other's inverse, and skos:related is declared to be symmetric. Note also that skos:broader and skos:narrower have been declared to be transitive.

An attempt has been made to provide an independent definition of the meaning of these properties, however the definitions are extremely vague. For example, the definition of skos:broader is given as: "A concept that is more general in meaning" and the definition of skos:related is given as: "A concept with which there is an associative semantic relationship" [MIL05B]. In practice, the meaning of these properties has been defined to be consistent the common usage of the words "broader", "narrower" and "related" in the context of thesauri broadly conforming to the ISO 2788:1986 standard [ISO86]. Both ISO 2788 and the more recent BS 8723-2 standard [BSI05B] restrict the use of hierarchical relationships to being either a generic (subclass/superclass) relationship, an instantial (class/instance) relationship, or some types of partitive (part/whole) relationship on the underlying “concepts”. However, many deployed thesauri do not strictly conform to these restrictions. In addition, other types of structured vocabulary employ a hierarchical arrangement of vocabulary units that does not entirely correspond to the stricter definition of BS 8723-2, although the arrangement may nevertheless justify the assumption of broadening relevance. These issues are discussed at further length in chapter 8.

I suggest that the meaning of the skos:broader, skos:narrower and skos:related properties be defined exactly as the hypothetical x:narrower, x:broader and x:associated properties were defined above. I.e. The use of skos:broader implies that broadening relevance may be assumed, the use of skos:related implies that associating relevance may be assumed and that these statements wholly define the semantics of these properties. I make this suggestion because it would provide a practical, heuristic way of determining the semantics of these traditionally very vague relationships, entirely in terms of the retrieval operations that are licensed by their use. This is arguably consistent with the intended use of hierarchical structures in thesauri, classification schemes and taxonomies.

## 8.3 Representing an Index

According to the theory, an "index" is characterised by a set of "fields", which are relations between a set of "document names" and a set of "concept names". An "index" is also characterised by a set of "field names", which allow the different "fields" of an "index" to be referenced from within a composite query, and a function mapping "field names" to "fields".

The notions of a "field name", a "document name" and a "concept name" were introduced as given sets. I consider options for creating an RDF representation of an index by first assuming that all "field names", all "document names" and all "concept names" are URI references. A simple RDF representation of an index may then be constructed by representing each "field name" as a property of an RDF graph and each "field" as the extension of the corresponding "field name" property, assuming the Herbrand interpretation of the graph [HAY04].

For example, consider a single-field index $\{f\mapsto\{d_1\mapsto t_1, d_2\mapsto t_2\}\}$ , where $\{f\}$ is the set of field names, $\{d_1, d_2\}$ is the set of document names and $\{t_1, t_2\}$ is the set of concept names. This index is then represented according to the above method by the RDF graph given below in the turtle syntax.

```
@prefix : <http://www.example.com/examples#>.

:d1 :f :t1.
:d2 :f :t2.
```

For example, consider the multiple-field index $\{f_1\mapsto\{d_1\mapsto t_1, d_2\mapsto t_2\}, f_2\mapsto\{d_1\mapsto t_3, d_2\mapsto t_4\}\}$ . This index is represented by the RDF graph given below in the turtle syntax.

```
@prefix : <http://www.example.com/examples#>.

:d1 :f1 :t1; :f2 :t3.
:d2 :f1 :t2; :f2 :t4.
```

Using the Herbrand interpretation, the set of “field names” is then given by the set of properties of the graph, the set of “fields” is given by the set of property extensions of the graph and the function mapping “field names” to “fields” is equivalent to the property extension function.

## 8.4 Representing Queries

Composite queries as defined in chapter 4 could also be given an RDF representation. For example, given the URIs x:AND, x:OR, x:NOT, x:ROP, x:ATOM denoting classes and x:field, x:concept, x:child, x:children, x:required, x:optional and x:prohibited denoting properties I suggest a possible representation for composite queries via the following examples, as a prelude to suggesting RDF representations for query expression mappings.

$$\begin{aligned}
&q_1 = atom(f, x)\\
&q_2 = and(\{atom(f, x), atom(f, y)\})\\
&q_3 = or(\{atom(f, x), atom(f, y)\})\\
&q_4 = and(\{atom(f, x), not(atom(f, y))\})\\
&q_5 = rop(\{atom(f, x)\}, \{atom(f, y)\}, \{atom(f, z)\})
\end{aligned}$$

RDF representations given below in the Turtle syntax, where variable names in the above are mapped to URIs in the eg: namespace with the same local name.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix : <http://www.example.com/examples#>.

:q1 a x:ATOM; x:field :f; x:concept :x.

:q2 a x:AND; x:children (
  [a x:ATOM; x:field :f; x:concept :x],
  [a x:ATOM; x:field :f; x:concept :y],
).

:q3 a x:OR; x:children (
  [a x:ATOM; x:field :f; x:concept :x],
  [a x:ATOM; x:field :f; x:concept :y],
).

:q4 a x:AND; x:children (
  [a x:ATOM; x:field :f; x:concept :x],
  [a x:NOT; x:child [a x:ATOM; x:field :f; x:concept :y]]
).

:q5 a x:ROP;
  x:required ([a x:ATOM; x:field :f; x:concept :x]);
  x:optional ([a x:ATOM; x:field :f; x:concept :y]);
  x:prohibited ([a x:ATOM; x:field :f; x:concept :z]).
```

Note that I have used RDF list constructs to ensure that the children of a query expression can be closed – i.e. cannot be modified through merging of data from multiple sources.

## 8.5 Representing Coordinations

Coordinations can be added to the RDF representation of either an index or a query with some additional language constructs, for example x:ucoord and x:ocoord as illustrated by the examples below.

$$\begin{aligned}
&I_1 = \{f \mapsto \{d \mapsto ocoord \langle x, y \rangle\}\}\\
&I_2 = \{f \mapsto \{d \mapsto ucoord \{x, y\}\}\}\\
&q_1 = atom(f, ocoord \langle x, y \rangle)\\
&q_2 = atom(f, ocoord \{x, y\})
\end{aligned}$$

RDF representations given below using the TRIG named graph syntax with variable names mapped to URIs in the eg: namespace.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix : <http://www.example.com/examples#>.

<I1> {
  :d :f [x:ocoord (:x, :y)].
}

<I2> {
  :d :f [x:ucoord (:x, :y)].
}

<queries> {
  :q1 a x:ATOM; x:field :f; x:concept [x:ocoord (:x, :y)].
  :q2 a x:ATOM; x:field :f; x:concept [x:ucoord (:x, :y)].
}
```

## 8.6 Representing a Structural Mapping

The same properties x:broader, x:narrower and x:associated can be reused to represent a structural mapping as they carry essentially the same semantics with respect to the relevance assumptions that they imply. Also required is a x:equivalent property. The representation of a structural mapping is then achieved for the four mapping relations in exactly the same way as suggested above for the three structural relations of a structured vocabulary.

## 8.7 Representing a Query Expression Mapping

To represent a query expression mapping we can reuse the same language constructs as for queries and omit the field name from atoms to indicate a query template. I also coin x:btrans, x:btrans and x:etrans to represent the functions of a query expression mapping. The query expression mapping *M* where $M.etrans=\{a \mapsto temp(b)\}$ is then represented by the graph below given using the TRIG syntax.

```
@prefix x: <http://www.example.com/retrieval-schema#>.
@prefix : <http://www.example.com/examples#>.

:M {
  :a x:etrans [a x:ATOM; x:concept :b].
}
```

## 8.8 Deriving an Index from an RDF Graph

As defined in Chapter 3, an "index" is fundamentally characterised by a set of "fields", where a "field" is a relation between a set of "document names" and a set of "concept names". A "field" can be derived from an arbitrary RDF graph via the set of bindings of a two-variable SPARQL query. For example, consider the query below given in the SPARQL query syntax.

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?d ?t
WHERE
{
  ?d dc:subject ?t.
}
```

Each binding of the variables ?d and ?t can be used to construct an ordered pair (?d, ?t), the set of which can be used to construct a “field”.

A set of “fields” can be derived from a graph via a set of two variable queries, and a “field name” can be associated with each “field” after-the-fact to construct an “index”. Alternatively, the “field name” can be taken directly from the graph via queries with 3 variables. For example, consider the query below intended to derive an “index” with 2 “fields”.

```
SELECT ?d ?f ?t
WHERE
{
  ?d ?f ?t.
  FILTER (
    ?f = <http://purl.org/dc/elements/1.1/subject> OR
    ?f = <http://purl.org/dc/elements/1.1/type> )
}
```

The set of distinct bindings of the ?f variable is taken as the set of "field names", and for each distinct value of the ?f variable the set of bindings of the ?d and ?t variables is used to derive a set of ordered pairs (?d, ?t) constituting the "field" for that "field name".

Note that, although when constructing an RDF graph to represent an "index" according to the method described above a "field" is represented as the extension of a property, when extracting a "field" from an existing RDF graph there is no need to be restricted to only this pattern. For example, the queries below are intended to derive a "field" from the set of bindings of the ?d and ?t variables.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT ?d ?t
WHERE
{
  ?t skos:isSubjectOf ?d.
}
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT ?d ?t
WHERE
{
  ?d a foaf:Person;
    foaf:homepage [skos:subject ?t].
}
```

## 8.9 Implementing Retrieval Functions Using SPARQL

Given the RDF representation of an index as suggested above it would be possible to implement much of the query evaluation functionality described in the theory directly as SPARQL queries over the RDF graph. For example, *and* expressions can be achieved through a simple graph pattern, *or* expressions can be achieved through the “UNION” operator, *not* expressions can be achieved in a slightly convoluted way by filtering on optional variables that are not bound and *rop* expressions can be achieved via the “OPTIONAL” operator. Further discussion of the implementation of retrieval systems directly and/or indirectly on top of generic RDF systems is beyond the scope of this report and remains an important area for future work.

# 9 Literature Review and Discussion

In this chapter I discuss similarities and analogies between the theory and practise presented in this report and previous work on text retrieval, retrieval using controlled structured vocabularies and Semantic Web languages for retrieval applications.

## 9.1 Approaches to Text Retrieval

The canonical reference for both theoretical approaches to text retrieval and questions of practical implementation is van Rijsbergen [RIJ79]. The chapters of his book "Information Retrieval" that are most relevant to this report are chapters 2, 3, 5, and 6. Because this reference is considered to be the canonical reference for this topic I consider each chapter in turn at length.

In chapter 2 of [RIJ79] ("Automatic Text Analysis") van Rijsbergen establishes the principle of a "document representative" as being the representation of the content of a document in an information retrieval system. A document is represented by a set of "class names", where each "class name" denotes a class of words that share a common stem – the set of "class names" for a document being thus the representation of the set of classes of word deemed to be significant for that document. An "index" is then derived from this set of class names, such that the class names constitute the effective indexing language. In the theoretical part of this dissertation I chose to use the term "concept names" to denote the components of the indexing language provided by a controlled vocabulary to draw out the analogy with the terminology employed by van Rijsbergen. Chapter 2 of [RIJ79] also establishes the principles of index term weighting based on the distribution of index terms in the collection, to reflect the power of an index term to discriminate between documents in the collection. Partly because he emphasizes the experimental evidence supporting the notion that index term weighting provides significant benefit and partly because most modern implementations of text retrieval systems employ weightings based on a distribution calculation, I have chosen to consider a similar weighting for controlled vocabulary index terms, although because the nature of controlled vocabulary indexing differs from text retrieval this may not ultimately prove worthwhile – this is an area for further empirical investigation.

In chapter 3 of [RIJ79] ("Automatic Classification") van Rijsbergen establishes theoretical principles for the classification of documents according to some measure of their similarity. This chapter is not immediately relevant to the subject of this report because there is little evidence that clustering techniques as applied to a controlled vocabulary index would offer any particular benefit with respect to retrieval. The importance of this chapter is perhaps in the interaction between document classification and clustering techniques based on text content and the semi-automatic construction of a controlled vocabulary index. The cost of manually creating and curating a controlled vocabulary index is of course high in terms of the initial and ongoing investment in skilled manual labour that is required. There may be a reciprocal relationship between document clustering and manual indexing and this remains a fruitful area for future research.

Chapter 5 (“Search Strategies”) of [RIJ79] is perhaps the most directly relevant to the subject of this report. In this chapter van Rijsbergen describes the principles of “Boolean search” in the evaluation of queries with Boolean connectives, which are directly analogous to the theory of query expressions and direct query evaluation developed in chapters 3 and 4 of this report. He also defines the notion of “coordination level” being the number of query atoms matched by a particular document and the notion of partially ranking documents based on coordination level, which he calls “simple matching”. These concepts are formalised and extended for arbitrarily nested composite query expressions in this report by the *mathchingposatoms* and *unweightedscore* functions.

Chapter 6 of [RIJ79] (“Probabilistic Retrieval”) establishes certain basic principles of the probabilistic approach to the concept of relevance. Although the treatment goes well beyond what is directly relevant to this report, the fundamental assumptions are important with respect to the interpretation of the “broader” “narrower” and “associated” relations as encapsulated in this report by the assumptions of broadening, associating and coordinating relevance (and of course the assumptions of ideal indexing). It is these assumptions that are absolutely key to the operational interpretation of the structure of a controlled vocabulary with respect to retrieval operations. This is one of the main thrusts of my argument for developing operational definitions framed in terms of relevance and retrieval for Semantic Web languages such as SKOS that intend to support controlled vocabulary indexing and search. Chapter 7 of [RIJ79] makes a discussion of the widely applied concepts of relevance, recall and precision and these discussions bear directly on the findings of this report.  The concepts of relevance, recall and precision have been used to build the foundations of the arguments presented here with respect to the application of structured vocabularies to retrieval and especially to the more involved challenges of extending this theory to account for coordination and mapping between vocabularies addressed in the later theoretical chapters.

Robertson and Spark Jones [ROB97] distill some of the principles common to [RIJ79] into a short paper describing simple strategies for text retrieval that have strong empirical justification. It is recommended that documents be indexed with stems (analogous to "class names" in [RIJ79]). The only type of "request" (query) modeled is a simple unstructured list of terms. Results may be ranked by the number of matching terms (the "coordination level" of [RIJ79]), however it is strongly recommended to use a formula for term weighting based on empirical evidence demonstrating the improvements in performance this brings. A simple formula is given for the "collection frequency weight" (a.k.a. "inverse document frequency weight") which is the source of the *idf* function presented in chapter 4 of this report. Robertson and Spark Jones go on to describe a term weighting function that takes into account within-document term frequency and document length, however neither of these are relevant to retrieval with controlled vocabularies and hence do not factor into the theory of this report. Also addressed are iterative strategies for improving search results, however I have chosen not to address iterative techniques in this report, because of course iterative techniques depends first on a solid theoretical description of basic non-iterative techniques. There is also a suggestion in [ROB97] that some documents may sensibly be indexed on "complex" or "compound terms" e.g. specialised proper names or fixed multi-word strings. There are parallels between this type of indexing and the theory of coordination presented in chapter 6 of this report.

Salton et al. [SAL75] define a vector space model for automatic indexing of documents based on analysis of text content. The paper is mostly concerned with calculating the similarity between two documents based on a vector representation of the content and thereby calculating the configuration and density of a document space, therefore the work is not of direct immediate relevance to the theory of controlled vocabulary retrieval developed here. The paper does however provide a theoretical foundation for the "term discrimination" model and empirical support for its usefulness in automatic indexing and hence I chose to explore the applicability of the inverse document frequency weights to controlled vocabulary indexing in this report. The notion of identifying "good" indexing terms by virtue of the extent to which their assignment "spreads out" documents in the document space could be relevant to strategies for automatic or semi-automated indexing with a controlled vocabulary, by preferentially providing suggestions to indexers that match this criterion of "goodness". Strategies to support the work of manual indexing remain an important topic of future research and it is hoped that some of the basic theories established in this report in conjunction with theories and empirical results from text retrieval such as those established by Salton et al. may lead to some positive findings.

Modern implementations of text retrieval systems include the Lucene indexing system [LUC04], the Xapian Project [XAP06] and of course Google's search engine [GOO06] [BRI98]. Many of the elements of the theory presented in this report are directly analogous to the underlying structures of the Lucene query system. The basic theory I have presented in chapters 3 and 4 whereby pseudo-Boolean query expressions ("and", "or", "not") are implemented in conjunction with weights based on inverse document frequency and other metrics is directly analogous to that which is implemented by the Lucene query engine. The notion of a "required-optional-prohibited" query as another form of pseudo-Boolean expression is implemented in the Lucene system and also within Google's query engine. The notion of a document being separately indexed in multiple fields, which are then available for independent matching within the same query, is drawn directly from the Lucene architecture, as is the notion of a “field boost” number which is factored into the scoring of results for queries that reference multiple fields. The basic metrics of term weighting are also implemented in Xapian, which also goes beyond the basic indexing and query support available in Lucene or Google to provide support for probabilistic retrieval based on an iterative approach to search.

## 9.2 Using Structured Vocabularies for Retrieval

ISO 2788-1986 [ISO86] establishes principles for the development of monolingual thesauri, which are controlled vocabularies intended for use in manual indexing of documents. The basic features of a monolingual thesaurus as defined by ISO 2788 are modeled formally by the notion of a "structured vocabulary" as described in chapter 3 of this report. The set of "preferred terms" (a.k.a. "descriptors") established by a monolingual thesaurus is intended to be used as an "indexing language" and hence there is a direct correspondence between the set of preferred terms provided by a monolingual thesaurus and the set of "concept names" constituting a "structured vocabulary" as defined here. Note that no restriction was placed on the nature of a "concept name" in this report – a concept name could be a string of Unicode characters representing a thesaurus descriptor or it could be a Uniform Resource Identifier (URI) [RFC05] for example.

ISO 2788 also defines two basic types of "*a priori*" or "thesaural" relationships between the preferred terms of a thesaurus, being the "hierarchical" ("broader"/"narrower") relationship and the "associative" relationship. These relationships are referred to as "*a priori*" because they are inherent in the sense of the terms and are not simply linked because they may appear together in the context of any given document. The relationships may be regarded as directly analogous to the *broader*, *narrower* and *associated* binary relations of a structured vocabulary as modeled in the theory of this report. While ISO 2788 provides a detailed explanation for the meaning of these relationships from a philosophical point of view, it has been a primary goal of this report to complement these definitions by providing a precise mathematical account of the operational meaning of these relationships with respect to their exploitation within retrieval systems. Hence this report has not attempted to provide an independent or parallel definition from a philosophical perspective. The operational definitions provided here are intended to be entirely consistent with the philosophical definitions provided by ISO 2788.

It should be noted that ISO 2788 is focused on providing guidelines for the developers of thesauri and only hints at strategies for designing and implementing the retrieval solutions that are assumed as the ultimate application of the thesaurus. Therefore it is a major goal of this report to articulate those assumptions about the operation of controlled vocabulary retrieval systems that might be consistent with and be seen to underly the ISO 2788 developmental guidelines. This report, in considering the issue of limited cost expansions and coordination in some detail, probably goes beyond the assumptions of ISO 2788, but that isn't to say that they are necessarily inconsistent. ISO 2788 does provide hints as to how specific combinations of terms may be indicated syntactically by a pre-coordinating indexer, which is essentially a suggestion with respect to the design and implementation of retrieval solutions that provide support for coordination. However the understanding of "pre-coordinate" and "post-coordinate" indexing as inherent in ISO 2788 and elsewhere can be quite misleading. Accordingly, "pre-coordinate" indexing is regarded as the construction of combinations of terms indicated syntactically at the time of indexing and "post-coordinate" indexing is regarded as the construction of combinations of terms after indexing has taken place (i.e. at the time of the user request) and this latter has been taken to be synonymous with the notion of a composite query expression. As I hope I have managed to convey by the structure of the theory presented in chapters 3 to 7 of this report, there is a fundamental distinction between the notion of a composite query and the notion of coordination – a composite query may or may not involve coordinations. The decision to formulate the theory thus was intended to provide a mathematical basis for the a solution to the requirement for avoiding the spurious results that may be obtained by the false interpretation of an "and" query expression as a "coordination".

ISO 5964-1985 [ISO85] is intended to provide practical and philosophical guidance with regards to the construction of multilingual thesauri. ISO 5964 is an extension of ISO 2788, depending on the definitions established therein and as such all the principles that apply to monolingual thesauri are also applied to multilingual thesauri. The ultimate purpose of a multilingual thesaurus is to allow indexers and searchers to operate in their native languages whilst retrieving documents from a common set. Thus ISO 5964 is intended to deal with the underlying situation where an index or a query has to be translated into another form with minimal loss of retrieval performance. The situation covered by ISO 5964 is in fact subtly different from the theoretical situation addressed in chapter 7 of this report, because ISO 5964 is addressing the problem of constructing a multilingual thesaurus "*ab initio*" which may then be used as a set of equivalent indexing languages, whereas chapter 7 addresses the more general problem of expressing a mapping between two pre-existing structured vocabularies such that queries and/or indexes may be appropriately translated. The difference lies in the fact that in this latter scenario it as assumed that feedback to the vocabularies being mapped is not possible and therefore that the nature of the mapping may not be either total or perfectly equivalent. By then expressing the nature of the mapping as either "broader", "narrower", "associated" or "equivalent", or by using a query expression mapping, approximations may be made with respect to anticipated losses of precision or recall under different circumstances, which in turn enables the translation of queries or indexes to be optimised with respect to preferred criteria.

Although the notions of "broader"/ "narrower", "associated" and "equivalent" mappings as modeled here are directly analogous to the notions of "partial", "inexact" and "exact" equivalences as introduced in ISO 5964, whereas ISO 5964 provides recommendations for modifying source and/or target language components towards an exact (for the purposes of retrieval) equivalence this report attempts to support the expression of the nature of a mapping relationship as it is, which does not require any modification to either source or target vocabularies. Note also another fundamental difference between this report and ISO 5964 in that chapter 7 considers the expression of mappings between "source" and "target" controlled vocabularies, ISO 5964 considers the construction of what is essentially a single controlled vocabulary which has interchangeable language components and therefore the notions of "source" and "target" are only applied temporarily when certain difficulties in establishing exact equivalences are encountered and roles may even be reversed.

The BS 8723 document series is intended to be a revision of the earlier thesaurus standards

ISO 2788 and ISO 5964. The scope of the BS 8723 is broader than the previous ISO standards, in that it considers thesauri in addition to other types of "structured vocabularies" intended for use as information retrieval tools. Additionally it considers a number of factors relevant to the implementation of computer systems supporting the development and use of structured vocabularies for information retrieval systems.

BS 8723-1:2005 [BSI05A] establishes the definitions used throughout the document series and BS 8723-2:2005 [BSI05B] focuses on the development and application of thesauri. Much of the guidance is identical to that provided by ISO 2788 however BS 8723-2 does make much more explicit the reasoning behind the guidelines provided in order to obtain the desired retrieval behaviours in information systems. I.e. many of the assumptions regarding the application of a thesaurus for information retrieval are elaborated. For example in the description of the application of the hierarchical relationship ("broader"/"narrower") it is stated that a strict adherence to the recommendations with respect to the types of philosophical relationships considered should be maintained in order to ensure that relevant results are consistently obtained when a search system performs an "exploded search" by including all narrower terms in a given query. This is an informal description of the strategy modeled in chapters 3 and 4 by the naive expansion of a query or equivalently an index. BS 8723-2 comes much closer to making explicit the underlying assumptions regarding the inferences that can be drawn regarding relevance of indexed documents from the structure of a controlled vocabulary. It is precisely these assumptions that I have attempted to state explicitly and capture formally in the theory part of this report and it is these assumptions that I am proposing become an integral part of the operational definitions of the SKOS vocabularies [MIL05A] [MIL05B]. Similarly in its discussion of "complex concepts" BS 8723-2 justifies heuristic rules for when and when not to "split" a concept on the grounds of the impact that this would have for retrieval systems that involve either pre- or post- coordinate indexing. BS 8723-2 in the example regarding the splitting of the "road safety" term informally describes the assumption underlying the inferences regarding relevance that might reasonably be drawn from the act coordination that I have attempted to capture formally in chapter 6 of this report.

BS 8723-2 and working drafts of BS 8723-3 [BSI06A] and BS 8723-4 [BSI06B] make repeated reference to the use of keywords "AND" and "OR" without providing any formal definition of what exactly is intended or how this relates to particular indexing and retrieval strategies. By providing a formal definition of composite queries in chapter 4 of the theory section of this report and by developing a theory for the interactions between composite queries and other technical issues such as limited cost expansions, coordination and mapping/translation I have hoped to provide a formal underpinning for some of the notions informally described in the BS 8723 document series.

Another area of direct relevance to this report in respect of BS 8723-2 is its treatment of the problem of change management in structured vocabularies intended for information retrieval. The challenge arises because an index may become inaccurate or inappropriate in response to changes in the indexing language. A mechanism is required to describe the nature of the change and to represent the change formally such that adaptation may be achieved in either indexes or queries. This requirement is seen as central to maximising the utility of a structured vocabulary over a potentially long working lifetime, a notion explored also by [MIL06A] in relation to the planned standardisation of the SKOS vocabularies [MIL05A] [MIL05B]. It is intended that the framework of the theory presented here should provide the necessary tools required to achieve one or more reasonable solutions to this requirement, because in effect the representation of change can be seen as equivalent to the mapping of one vocabulary to another and therefore all of the theory in support of mapping and translation can be brought to bear on the problem of change representation and management.

It should be noted that the logical constraints placed on the *broader*, *narrower* and *associated* binary relations of a structured vocabulary as defined in chapter 3 are in complete agreement with the recommendations of BS 8723-2 with respect to the requirements of thesaurus management software systems.

BS8723-3 (draft 2006-05-17) [BSI06A] extends the scope of structured vocabularies to consider vocabularies other than thesauri whose intended application is for information retrieval. Specifically, classification schemes, subject heading systems, taxonomies and ontologies are described in enough detail as to make a clear comparison with thesauri and to draw out those features of establishment, development and presentation that bear directly on the implementation and effectiveness of retrieval systems. In deriving a formal theory for the structure and application of structured vocabularies in this report I have intended to lay a foundation within which the interrelationships between these vocabulary types may be precisely analysed and potential interoperability may be maximised. For example by providing operational definitions of the *broader, narrower* and *associative* relationships in terms of relevance assumptions it is hoped that it may be seed that hierarchical relationships in classification schemes and in taxonomies serve a fundamentally similar purpose and are hence amenable to similar representations and analysis, even though they may not completely fulfil the philosophical criteria demanded by hierarchical relationships in thesauri.

In particular, attention is drawn in BS 8723-3 to the fact that in classification schemes and in taxonomies the hierarchical relationships may not conform to the strict constraints as recommended for the application of hierarchical relationships in thesauri. The question posed by this report is then, do these hierarchical relationships satisfy the assumption of broadening relevance or not? The answer to this question is fundamental to establishing the possibility for a common representation framework for both thesauri and classification schemes or taxonomies.

BS 8723-3 also pays some attention to the construction and application of synthetic classification schemes, in addition to the application of subject heading schemes, where coordination is a fundamental principle of use. An open question is whether the theory of coordination developed in chapter 6 of this report and the assumption of coordinating relevance as stated there is consistent with the use of synthetic classification schemes such as the Bliss Classification Schedule and subject heading systems such as the Library of Congress Subject Headings. Developing a theory for the effective expansion of synthetic (i.e. coordinated) indexes is a vital element for the use of these tools that have enormous potential for specificity and adaptability but are difficult to use and have a low concomitant probability of perfectly similar query.

The example of "parametric search" as an application of taxonomies as given in BS 8723-3 is modelled by an index with multiple fields in the theory chapters of this report and this type of application is a primary motivation for the theory of multiple-field indexes developed here.

BS 8723-4 (draft 2006-06-29) [BSI06B] considers the problem of expressing mappings between vocabularies in order that queries may be translated so that one vocabulary may be used to query an collection indexed with another vocabulary. The issue of multilingualism is considered a special case of this more general situation. The standard provides recommendations on the way relationships may be expressed across vocabularies. The types of mapping considered in BS 8723-4 suggest both the structural mapping and the query expression mappings defined in chapter 7 of this report. BS 8723-4 also makes suggestions for how to translate an index based on the value of various types of mappings and I have attempted to extend and to formalise some of the suggestions made there in this report.

Doerr [DOE01] provides much of the inspiration for the theoretical basis of this report, especially the theory of translation presented in chapter 7. His paper addresses the problem of expressing mappings between different vocabularies so that queries may be transformed such that either recall or precision may be preserved. His interpretation of the nature of the “broader” and “narrower” relationships of a thesaurus as implying a set subsumption relationship between sets of correspondingly indexed objects is directly analogous to the statement of the assumption of broadening relevance presented in chapter 3 of this report. Doerr also establishes the principles of "concept-based mapping", "complete mapping" and "optimal mapping" which are the basis for the notion of a structural mapping as defined in chapter 7 of this report. [DOE01] considers projects that have used Boolean operators to construct mapping expressions and also assumes a particular interpretation of these expressions in the underlying retrieval systems. The theory of composite queries presented in chapter 4 of this report is intended to provide a formal basis for these less formal statements and for the expansion and translation of queries (and indexes). Although Doerr recommends the structural approach over the use of query expressions to achieve a mapping I have sought to provide a theoretical basis for both so that they may be directly compared and so that empirical studies may be executed.

A number of authors have developed models for the use of structured vocabularies for information retrieval and in particular for various metrics for computing "semantic distance" (generalised in this report as "relevance cost") in order to expand and to rank results. Tudhope et al. [TUD02] develop a matching function for query expansion as applied in the FACET project, also described at [TUD06]. Similar ideas are presented for ontology-driven search by Corby et al. [COR05] and Gandon et al. [GAN05].

## 9.3 RDF Representation of Thesauri & Similar Vocabulary Types

The basic technological framework of the Semantic Web is documented at [RFC05] [BRI04] [MAN04] [KLY04] [HAY04] [PRU06] [CLA06] [PAT04] [SMI04]. The Dublin Core Abstract model [POW05] is also directly relevant to the RDF representation of structured metadata to support information retrieval.

The Semantic Web Advanced Development for Europe (SWAD-Europe) project published a number of reports relating to the RDF representation of thesauri, classification schemes and similar vocabulary types. The review of RDF thesaurus work [MAT04] provides an overview of a number of design patterns for representing thesauri using RDF. [MIL04A] Describes the SKOS Core RDF language for the representation of thesauri (since deprecated and replaced by [MIL05A] and [MIL05B]). RDF representations for multilingual thesauri and inter-thesaurus mappings are provided in [MIL04B] and [MIL04C]. RDF representations of classification schemes and classification data are considered in [MIL04D]. Strategies for migrating existing thesauri to Semantic Web representations are considered in [MIL04E] (since deprecated and replaced by [MIL05C]). These reports form much of the foundation for this report. In particular, the theory of this report is intended to support the continued development of the SKOS Core vocabulary and additional vocabularies to support representation of mappings between vocabularies.

The Simple Knowledge Organisation System (SKOS) [MIL05A] [MIL05B] [MIL05C] is an RDF language for representing thesauri and similar vocabulary types. Examples of the application of SKOS are given in [MIL05D] and [MIL05E]. The W3C Semantic Web Deployment Working Group is chartered with developing SKOS to W3C Recommendation status and [MIL06A] describes challenges facing this activity and suggests initial options for progress. In particular, [MIL06A] suggested that a set of use cases be defined and the theory of this report has been intended to directly support that activity by enabling a comparative analysis of use case and hence an extraction of common requirements. Assem et al. [ASS06] presents a method for converting thesauri to SKOS, which should be contrasted with [ASS04].

# 10 Summary and Conclusions

This report has developed a formal theory of retrieval using controlled vocabularies that have a simple and intuitive structure. A theory has been developed in chapters 3 – 7 of this report to define the structure of a controlled vocabulary, the structure of an index, the structure of atomic and composite queries, strategies for the evaluation of queries, for the expansion of queries or indexes to improve recall and for the ranking of results to improve perceived precision. The theory follows from key assumptions about the implications of the structure of a vocabulary for the relevance of indexed documents with respect to queries. This report has attempted to articulate these assumptions in a clear and concise manner, so that the foundations of the theory may be critically evaluated and subjected to empirical investigation.

This report has also suggested design patterns for the concrete representation of data structures required for the implementation of distributed retrieval applications using Semantic Web languages. Specifically the design of RDF languages and representation patterns for the representation of structured vocabularies and indexes has been discussed and suggestions for simple and effective patterns given. As Semantic Web languages such as the Simple Knowledge Organisation System (SKOS) are developed on top of RDF and OWL to support distributed retrieval applications, it is the intention of this report to provide a degree of theoretical and mathematical rigour that can be used to ground the definitions of these languages and to ensure the consistency in implementation that is required for the effective deployment of a Web standard.

The theory presented here is by no means complete and particular areas require further development. The theory of limited cost expansions requires further theoretical study to establish the correctness of the underlying assumptions and of the basic analysis. The theory of limited cost expansions also requires empirical investigation to establish effective cost functions and parameter values. The notion of limited cost expansions provides perhaps the greatest potential for maximising the value of a structured vocabulary by enabling recall to be increased without compromising perceived precision, however the theory needs to be fully developed in respect of how limited cost expansions interact with composite query expressions, with coordinated queries and indexes and with vocabulary mappings. This remains a fruitful area for investigation because one of the major challenges regarding the use of structured vocabularies for retrieval via manual or semi-automatic indexing is the curation of vocabularies and indexes over time to ensure continued applicability and utility. A proper theory of mapping and of translation is critical to the problem of change management because versions of a vocabulary can be modeled as discrete vocabularies and hence mappings may be expressed between them. Given a mapping between versions, translations of queries or indexes may be achieved automatically with predictable effects in terms of the consequences for retrieval behaviour. Supporting the adaptation of indexes in response to vocabulary change is critical to ensuring the continued utility and hence value of both indexes and structured vocabularies over time.

With improvements in text retrieval systems, augmented with an understanding of the enormous utility of hyperlink structures and/or social networks as demonstrated by Google and by socially driven systems such as del.icio.us or flickr.com, there is significant pressure on the designers of retrieval systems based on the application of vocabulary control and manual indexing to demonstrate the potential for “profitability” of these systems. Although in many specialised domains it is arguably not possible to achieve the required levels of precision by any other means than vocabulary control and intellectual input to indexing, there is nevertheless a need to maximise the utility of controlled vocabulary solutions whilst minimising their cost. This is the primary reason why models of vocabulary development are evolving, to obtain a balance between functionality and cost. Furthermore, controlled vocabulary solutions are unlikely to be applied in isolation but rather in concert with other solutions to provide an integrated suite of functionalities. By drawing on the wealth of experience and theory regarding text retrieval systems and applying relevant techniques to the problem of retrieval using structured vocabularies it is hoped that this report has paved the way for the development of highly effective retrieval systems that obtain maximum value from the application of vocabulary control and from the intellectual input of end-users.

# 11 Appendix: Dissertation Plan

## 11.1 Abstract

Controlled, structured, vocabularies remain a useful tool for managing collections of digital artefacts, providing the basis for applications that allow a user to find (i.e. 'retrieve') objects of interest from within large collections in an efficient manner. A variety of conventions exist for the design, construction and application of controlled, structured, vocabularies intended for this purpose, such as thesauri, classification schemes, subject heading systems, taxonomies, 'ontologies' and 'folksonomies'. This variety in part reflects different traditions and communities of use, and in part reflects differences in retrieval functionality required by motivating applications, which can range from online library catalogues to social bookmarking websites.

The ultimate goal of this dissertation is to provide a logical model for the implementation of retrieval systems that use metadata derived from controlled, structured, vocabularies. The range of functionalities required by these systems is explored in relation to concrete usage scenarios: (1) a 'blogging' application with support for categorisation; (2) a social bookmarking website with support for 'tagging' via a folksonomy; (3) an online directory of organisations in the environmental sector, classified in several 'facets' using taxonomies; (4) an online portal of reports with browse and search functionality that uses search terms derived from a thesaurus. An abstract syntax is developed to support the comparative analysis of various indexing and retrieval strategies, and a model-theoretic semantics for this syntax is given. The semantics provides a mathematical basis for describing at an abstract level the implementation of a number of generalised indexing and retrieval strategies, which can then be mapped to concrete implementations using a particular technology. Examples of concrete implementations are given for the RDF, OWL and SPARQL family of languages.

## 11.2 Preliminary Research

Thesauri are deployed in retrieval systems in a number of sectors, including government (e.g. the 'Government Category List', now the 'Integrated Public Sector Vocabulary'), heritage (e.g. the English Heritage thesauri), humanities (e.g. the 'Art and Architecture Thesaurus'), and scientific literature (e.g. 'INSPEC'). Classification schemes are deployed mainly in the library sector, and can be either 'enumerative' e.g. the 'Dewey Decimal Classification', the 'Universal Decimal Classification', or 'analytico-synthetic' e.g. the 'Bliss Classification' and the 'Colon Classification'. Classification schemes are also deployed by more specialised sectors, e.g. the 'Physics and Astronomy Classification Scheme'. Generally speaking, 'taxonomies' implement a subset of the features offered by classification schemes and thesauri, and this term is a buzzword most employed with enterprise information management applications. Subject heading systems are used by libraries (e.g. the 'Library of Congress Subject Headings') and other sectors (e.g. the 'Medical Subject Headings'), and can involve complex rules for the coordination of headings from other terms.

'Folksonomies' are a relatively new phenomenon, with two exemplars of their use being the photo sharing web site 'Flickr' (www.flickr.com) and the social bookmarking web site 'Delicious' (del.icio.us).

The modern practice of construction and management of thesauri has been strongly influenced by the development of the ISO 2788 and ISO 5964 standards, dating from 1985/6. The standardisation of computational systems and formal representations for thesauri is relatively sparse, with the most prominent efforts being the MARC21 library record system and the ZThes XML format (intended to support the Z39.19 protocol). The MARC21 format is also widely used to exchange classification data. Some effort has been made to relate thesauri with the semantic web languages RDF and OWL, the most fully developed of these being the 'SKOS' ('Simple Knowledge Organisation Systems') initiated within W3C. Initiatives like SKOS could greatly benefit from a formal, mathematical, description of the meaning and use of thesauri, classification schemes &c. because such a mathematical model can motivate fundamental design goals. Some applications have already used SKOS in combination with OWL to deliver rich retrieval functionality via the web, e.g. the 'Semantic Web Environmental Directory'.

The use of model-theory to define the semantics of a language in terms of set theory, and hence to provide a formal description of the logical properties of a language, is exemplified by the RDF Semantics, and the OWL Abstract Syntax and Semantics recommendations from W3C.

## 11.3 Objectives

(1) To describe a set of usage scenarios that exemplify a range of functionalities employed in general by retrieval systems that use metadata derived from controlled, structured, vocabularies.

(2) To develop an abstract syntax that can be used to express (i) the structure of a controlled vocabulary; (ii) an index over a collection of digital artefacts using terms derived from a controlled vocabulary; (iii) a query over an index.

(3) To develop a model-theoretic semantics for the abstract syntax.

(4) To provide a mathematical description of the retrieval functionality offered in each of the usage scenarios.

(5) To provide a mapping from the mathematical description of a functionality to a concrete implementation using the family of semantic web languages RDF, OWL and SPARQL.

## 11.4 Methods

The usage scenarios will be developed by exploring and documenting the functionality of currently deployed applications, e.g. Wordpress for scenario (1) and Delicious for scenario (2).

The abstract syntax and semantics will be developed as an application of model-theory, and checked by analogy with the RDF and OWL abstract syntaxes and semantics.

That the abstract, mathematical description of each example scenario does in fact satisfy the functionality described should be demonstrable and amenable to mathematical proof.

The mapping from the abstract description of retrieval functionality to a concrete implementation using RDF, OWL and SPARQL will be tested in relation to a simple test framework, which will be used to exercise samples of code given.

## 11.5 Resources

Expert advice will be required to check the semantics of the abstract syntax, especially in relation to RDF and OWL semantics.

The only hardware required is a single development platform, and required software includes a development environment that supports XML, HTML and Java application authoring.

## 11.6 Schedule

A timeline for the major tasks of the project is given by the chart on the following page.

The output from task 1 is a bibliography of relevant sources.

The output from task 2 is a description of the four usage scenarios.

The output from task 3 is a formal specification of an abstract syntax, and a model-theoretic semantics for the syntax.

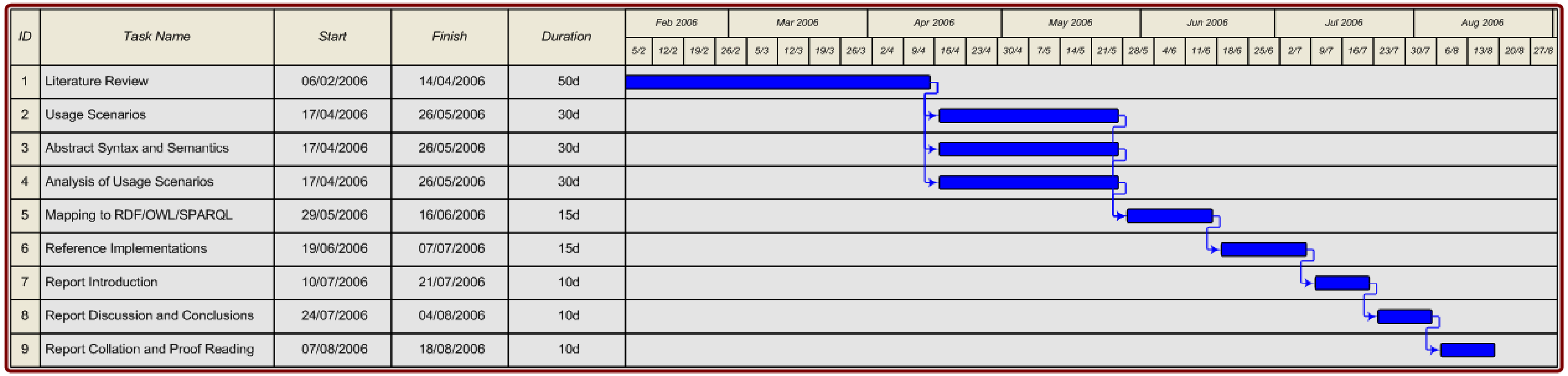
The output from task 4 is a formal description of each usage scenario in terms of the abstract syntax.

Tasks 2, 3 and 4 are carried out concurrently, because the syntax and semantics will be developed and refined as each usage scenario is elaborated and analysed.

The output from task 5 is a mapping from the abstract syntax to RDF triples, OWL, and SPARQL queries.

The output from task 6 is code snippets demonstrating the implementation of retrieval functionality using RDF, OWL and SPARQL technologies.

The output from tasks 7, 8 and 9 is a completed dissertation report.



| ID | Task Name | Start | Finish | Duration |
|---|---|---|---|---|
| 1 | Literature Review | 06/02/2006 | 14/04/2006 | 50d |
| 2 | Usage Scenarios | 17/04/2006 | 26/05/2006 | 30d |
| 3 | Abstract Syntax and Semantics | 17/04/2006 | 26/05/2006 | 30d |
| 4 | Analysis of Usage Scenarios | 17/04/2006 | 26/05/2006 | 30d |
| 5 | Mapping to RDF/OWL/SPARQL | 29/05/2006 | 16/06/2006 | 15d |
| 6 | Reference Implementations | 19/06/2006 | 07/07/2006 | 15d |
| 7 | Report Introduction | 10/07/2006 | 21/07/2006 | 10d |
| 8 | Report Discussion and Conclusions | 24/07/2006 | 04/08/2006 | 10d |
| 9 | Report Collation and Proof Reading | 07/08/2006 | 18/08/2006 | 10d |

## 11.7 Suggested Breakdown of Marks

(1) Evidence of research into the background to the topic: 25%

The theory presented will draw on Semantic Web standards, standards for the use of structured vocabularies for information retrieval and scholarly works on theoretical and empirical aspects of both text retrieval and retrieval using structured vocabularies.

(2) Analytical content: 50%

The main body of the dissertation will consist of the development of a theoretical framework for the analysis of information retrieval systems and strategies especially those involving the application of structured vocabularies. This theory is developed to be consistent with a set of use cases. The goal of the theory is to be able to formally characterise a particular family of problems relating to the use of structured vocabularies for information retrieval. The goal is also to provide a mathematical framework to support the analysis of the assumptions and the requirements for Semantic Web languages intended as support for retrieval applications. The theoretical framework and the accompanying use cases comprise the analytical content of the dissertation.

(3) Technical content: 15%

Once the theory has been developed, mappings and design patterns are suggested for the representation of relevant data structures using RDF and OWL. This comprises the technical content of the dissertation.

(4) Dissertation plan: 5%

(5) P0012 Research and Study Methods presentation: 5%

## 11.8 Retrospective Notes on Divergence from the Plan

I initially planned to develop a set of abstract syntaxes and a model-theoretic semantics for the syntaxes as the basis for a theoretical framework describing the use of structured vocabularies for retrieval. However, it became apparent that it was possible to develop a formal theory directly, using a formal mathematical notation (the Z notation), which removed the necessity for any abstract syntaxes and allowed expression of mathematical concept more directly. The formal theory expressed in this way could then be mapped directly to RDF graphs and to interpretations of RDF graphs, which proved a far simpler technique than would have been required if abstract syntaxes and semantics had had to be compared.  The ultimate goal was to provide a mathematical account of retrieval systems and this could be achieved more directly by the use of a formal notation system.

I initially planned to develop reference implementations of key parts of retrieval systems to demonstrate the implementation of certain features of the theoretical framework. However it was decided that this was beyond the scope of the current dissertation and that the dissertation should focus on developing the theoretical framework to cover a range of situations involving structured vocabularies and on suggestions for mapping these abstract data structures to RDF representations.

The use case involving the del.icio.us social bookmarking system was dropped in favour of more traditional structured vocabulary use cases – further work can easily extend the analysis presented to consider the more recent development of social bookmarking and the use of “folksonomies”.

# Bibliography

ASS04: Mark van Assem, Maarten R. Menken, Guus Schreiber, Jan Wielemaker and Bob W, A Method for Converting Thesauri to RDF/OWL, 2004, Proceedings of the Third International Semantic Web Conference (ISWC'04)
ASS06: Mark van Assem, Veronique Malaise, Alistair Miles, and Guus Schreiber, A Method to Convert Thesauri to SKOS, 2006, Proc. European Semantic Web Conference
BRA06: Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler and François Yergea, Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006, World Wide Web Consortium
BRI04: Dan Brickley and R. V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, 2004, World Wide Web Consortium
BRI98: Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, 1998
BSI05A: , Structured vocabularies for information retrieval - Guide - Part 1: Definitions, symbols and abbreviations, 2005, BSI
BSI05B: , Structured vocabularies for information retrieval - Guide - Part 2: Thesauri, 2005, BSI
BSI06A: , Structured Vocabularies for Information Retrieval - Guide. Part 3. Vocabularies other than thesauri, 2006, British Standards Institution
BSI06B: , Structured Vocabularies for Information Retrieval - Guide. Part 4. Interoperability between vocabularies, 2006, British Standards Institution
CLA06: Kendall Grant Clark, SPARQL Protocol for RDF, 2006, World Wide Web Consortium
COR05: Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker and Fabien Gandon, Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese, 2005, INRIA
DOE01: Martin Doerr, Semantic Problems of Thesaurus Mapping, 2001, Journal of Digital Information
GAN05: Fabien Gandon, Olivier Corby, Alain Giboin, Nicolas Gronnier and Cecile Gui, Graph-based Inferences in a Semantic Web Server for the Cartography of Competencies in a Telecom Valley, 2005
GOO06: , The Essentials of Google Search, 2006, http://www.google.com/help/basics.html
HAY04: Patrick Hayes, RDF Semantics, 2004, World Wide Web Consortium
ISO85: , Documentation - Guidelines for the establishment and development of multilingual thesauri, 1985, International Standards Organisation
ISO86: , Documentation - Guidelines for the establishment and development of monolingual thesauri, 1986, International Standards Organisation
KLY04: Graham Klyne and Jeremy J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax, 2004, World Wide Web Consortium
LUC04: Erik Hatcher, Otis Gospodnetic, Lucene in Action, 2004
MAN04: Frank Manola and Eric Miller, RDF Primer, 2004, World Wide Web Consortium
MAT04: Brian Matthews and Alistair Miles, Review of RDF Thesaurus Work, 2004, SWAD-Europe Project
MIL04A: Alistair Miles, Nikki Rogers and Dave Beckett, An RDF Schema for Thesauri (SKOS-Core 1.0 Guide), 2004, SWAD-Europe Project
MIL04B: Alistair Miles, Brian Matthews and Michael Wilson, RDF Encoding of Multilingual Thesauri, 2004, SWAD-Europe Project
MIL04C: Alistair Miles and Brian Matthews, Inter-Thesaurus Mapping, 2004, SWAD-Europe Project
MIL04D: Alistair Miles, RDF Encoding of Classification Schemes, 2004, SWAD-Europe

Project
MIL04E: Alistair Miles, Nikki Rogers and Dave Beckett, Migrating Thesauri to the Semantic Web, 2004, SWAD-Europe Project
MIL05A: Alistair Miles and Dan Brickley, SKOS Core Guide, 2005, World Wide Web Consortium
MIL05B: Alistair Miles and Dan Brickley, SKOS Core Vocabulary Specification, 2005, World Wide Web Consortium
MIL05C: Alistair Miles, Quick Guide to Publishing a Thesaurus on the Semantic Web, 2005, World Wide Web Consortium
MIL05D: A. Miles, B. Matthews, M. D. Wilson and D. Brickley, SKOS Core: Simple Knowledge Organisation for the Web, 2005, Proc. International Conference on Dublin Core and Metadata Applications
MIL05E: Alistair Miles, Brian Matthews, Dave Beckett, Dan Brickley, Michael Wilson, SKOS: A language to describe simple knowledge structures for the web, 2005, Proc. XTech 2005: XML, The Web and Beyond
MIL06A: Alistair Miles, SKOS: Requirements for Standardisation, 2006, Proc. Internation Conference on Dublin Core and Me
NOY06: Natasha Noy and Alan Rector, Defining N-ary Relations on the Semantic Web, 2006, World Wide Web Consortium
PAT04: Peter F. Patel-Schneider, Patrick Hayes and Ian Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax, 2004, World Wide Web Consortium
POW05: Andy Powell, Mikael Nilsson, Ambjörn Naeve and Pete Johnston, DCMI Abstract Model, 2005, Dublin Core Metadata Initiative
PRU06: Eric Prud'hommeaux and Andy Seaborne, SPARQL Query Language for RDF, 2006, World Wide Web Consortium
RFC05: T. Berners-Lee, R. Fielding and L. Masinter, Uniform Resource Identifier (URI): Generic Syntax, 2005, Internet Engineering Task Force
RIJ79: C. J. van Rijsbergen, Information Retrieval (2nd Edition), 1979
ROB97: S.E. Robertson and K. Spärck Jones, Simple, proven approaches to text retrieval, 1997, University of Cambridge Computer Laboratory
SAL75: G. Salton, A. Wong and C.S. Yang, A Vector Space Model for Automatic Indexing, 1975, Communications of the ACM
SMI04: Michael K. Smith, Chris Welty and Deborah L. McGuinness, OWL Web Ontology Language Guide, 2004, World Wide Web Consortium
SPI89: J.M. Spivey, The Z Notation: A Reference Manual, 1989
TUD02: Douglas Tudhope, Ceri Binding, Dorothee Blocks and Daniel Cunliffe, Compound Descriptors in Context: A Matching Function for Classifications and Thesauri, 2002, Proc. Joint Conference on Digital Libraries
TUD06: Douglas Tudhope, Ceri Binding, Dorothee Blocks and Daniel Cunliffe, Query Expansion via Conceptual Distance in Thesaurus Indexed Collections,
XAP06: , An introduction to information retrieval [Xapian Project], 2006, http://www.xapian.org/docs/intro_ir.html