Soft-link Hypertext for Information Retrieval

Mark d’Inverno a Paul Howells a Michael Hu b Michael Luck c

a School of Computer Science, University of Westminster, 115 New Cavendish Street, London, W1W 6UW, UK. Email: {dinverm,howellp}@westminster.ac.uk
b School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore 639798
c Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK. Email: mml@ecs.soton.ac.uk

Abstract

This paper provides a formal specification in Z of a new intelligent hypertext model called the soft-link hypertext model (SLHM). This model has been implemented and extensively tested, and provides a new methodology for constructing the future generation of information retrieval systems. SLHM has the following three major advantages. First, it is automatically formulated. Second, powerful neural learning mechanisms are applied, thereby improving its efficiency and applicability. Third, machine intelligence installed can be utilised for on-line assistance during navigating and information browsing. This specification has been developed by application of an existing formal framework for specifying hypertext systems.

Keywords Hypertext; Intelligent Information Retrieval Systems; Formal Specification

1 Introduction

Much of the information circulating in the modern world is available in machine-readable form and can be easily stored in computer systems. However, storage of large masses of information in itself is of no value unless systems are designed to make selected items available to interested users. In traditional boolean systems, large collections of documents are processed and indexed; associations from individual keywords (also known as index terms) to related documents provide a quick and straightforward approach to information retrieval. As a result, document references that match available boolean query statements can be found rapidly and can be displayed in real time, with users checking
these references and retrieving those that interest them. This search and retrieval process is repeated until users have found the information they need. Alternatively, a different type of information retrieval environment is based on the **hypertext** model, in which documents are linked with each other to form an inter-connected information web. Users obtain information by following these pre-installed links and browsing from one document to another in the information database.

The traditional boolean search model represents global and direct access to any documents in the database. Hypertext, however, complements this global approach with local browsing based on these links among different documents, or among different parts of one document and, therefore, it provides new potential for structuring and presenting information.

A hypertext system is one that attempts to superimpose an external structure on data, in order that it can be accessed efficiently according to different user needs. Essentially, generating a hypertext structure involves creating associations — typically manually — between various parts of different documents, resulting in a structure known as the information web. However, there are many difficulties in the generation, description, and presentation of this inter-connected hypertext structure. This leads to several well-documented problems [16], of which the salient ones are as follows.

- **Designing and constructing these information webs is expensive, laborious and tedious.** It involves defining each individual **node** (also called **card**, or **frame** in some systems), identifying proper keywords to be connected (also known as **anchors**), and building up **hypertext links** between a pair of suitable anchors. Furthermore, manual hypertext construction tends to represent only the author’s or editor’s interpretation of related documents, which may not always be well accepted.

- **These manually built-up webs often confuse or mislead users because there is no transparency nor appropriate guidance facilities.** Without systematic rules to govern the creation of the hypertext structure, plus the lack of adequate and efficient on-line navigation tools to help, users often find it difficult to find information by browsing in a hypertext system. This is the well-known “**lost in hypertext**” symptom [14,25]. Many efforts have been made to address the issue, but this is still one of the problems that may constrain the further applications of hypertext, especially as target databases grow further.

- **Once these inter-connected information webs are built, they usually exist as structures that are difficult and expensive to alter.** Consequently it is very difficult, if not impossible, to amend and improve these webs. So far, most hypertext systems constitute a static and unalterable management structure. Some efforts have been made to relate them to indexes used in boolean search in order to impose a more dynamic and flexible option into
the hypertext environment, but this has proven inadequate for sophisticated applications; more dedicated data structures are still needed.

In response to these problems, a new design of information web, known as the conceptual index has been proposed [19]. Based on the conceptual index, a new information retrieval model called the soft-link hypertext model (SLHM) has been developed and implemented. The conceptual index extends the functionality of conventional boolean search to more sophisticated applications including hypertext. Compared with other similar projects in building hypertext structures upon boolean systems, SLHM provides the following three advantages.

(1) The conceptual index is automatically formulated.
(2) Powerful neural learning mechanisms are applied to the conceptual index. As the hypertext is used, these mechanisms are invoked to train the hypertext, thus improving its efficiency and applicability for user groups.
(3) Statistical user data are collected in the model to formulate intelligent indicators, which are utilised for on-line assistance during information search and browsing.

This paper provides a complete formal description of the soft-link hypertext model. In Section 2 we give a specification of the structure of the conceptual index and define the operations of indexing and hyperization on the conceptual index. This represents the high level description of SLHM. Section 3 presents the specification of the intelligent conceptual index, Section 4 shows how the SLHM can be applied in information retrieval and Section 5 shows how SLHM can be updated whilst maintaining the validity of machine-learnt information. Lastly, Section 6 provides a summary of the work presented in this paper.

1.1 Background

The problems described above have been described elsewhere [16] and some potential solutions proposed [7,9]. In particular, different types of semantic networks have been considered and applied as a supplement to conventional boolean search [8,26,28]. However, these projects have only been successful in providing alternative tools for refining the user queries for information search and retrieval, but are not tailored for information browsing from one document to another [28]. In our model, the associations between a pair of keywords are accommodated so that the model's functionality can be fully extended from the boolean search to more sophisticated information retrieval applications including hypertext. This is possible by using the conceptual index data structure and makes information search and browsing much more flexible. For a full review of this work see [19].
The algorithms used in SLHM are based on connectionist networks [1,5,15,18]. Once the conceptual index is defined and represented as a connectionist network, various neural network learning algorithms can be applied to support its operation and applications [27]. Earlier attempts to apply the connectionist network methodology include Mozer's first application to information retrieval [32], Cohen and Kjeldsen's constrained spreading activation in their semantic network for matching research funding agencies and research topics [6], Lelu's application of spreading activation in image databases [22] and, finally and most thoroughly, Belew's AIR project on spreading activation on query-index-abstract networks [2]. Nonetheless, these projects are only limited to small-scale bibliographic environments [18]. Compared with them, SLHM accommodates significantly more complicated information databases, and can potentially be extended to multimedia environments. Furthermore, it differs from these earlier connectionist network projects in its collection of statistical data from previous applications and the use of this data as intelligent indicators for on-line assistance during information browsing [19].

1.2 An Overview of SLHM

Any information retrieval system can be treated as a self-inclusive information space. SLHM, shown in Figure 1, divides any such space into two layers: the document space and the index space. The document space is analogous to the Storage Layer of the Dexter Hypertext Model [17]; it contains all relevant documents in their original forms. The index space is analogous to the Link Layer of the Dexter Hypertext Model; it includes an abstract description of the document space, together with all the auxiliary mechanisms necessary to support the operations on, and applications of, the information space.

Traditionally, an index provides a collection of keywords (also called index terms) as an external description of the document space. In SLHM, these different index terms are further associated with each other to formulate a new data structure called the conceptual index. Consequently, the conceptual index comprises two main elements: keywords and the associations between keywords, represented by concepts and links respectively. The mapping from concepts to related documents is used to support the boolean search. Furthermore, the links between different keywords are used to support browsing in the information space from one concept to another, and from one document to another. In such a way, the conceptual index extends the functionality of the traditional index from the boolean search to more sophisticated information retrieval applications including hypertext.

Most current systems are limited in the following ways.
• The links are defined and assembled manually, and are thus liable to human prejudice and ignorance.
• Two concepts are related by a link without necessarily recording an explicit description or motivation as to why it was created.
• Once a link is made it remains unless it is manually removed. We refer to this group of links as hard links.

In SLHM, however, general rules are used to identify the links automatically, ensuring that all links are systematically generated. Since these rules are made explicit and available for users once the conceptual index is formulated, the motivation for the creation of each link can be understood. Furthermore, learning mechanisms are installed to record intelligent statistical information and to train the conceptual index as it is used in information retrieval. This information can be utilised for on-line assistance during an information retrieval session. We call such links soft links and argue that they do not suffer from the limitations of hard links. They enhance user mobility in an information space, and make the application of the index space more flexible and powerful. Note that we are not implying that a system that combines both hard and soft links is necessarily the most effective hypertext system, but only that soft links are not limited in the same way as hard links.
1.3 The Specification

In this paper we provide a formal specification of SLHM that represents a formal, concise and readable definition of SLHM and its applications. The specification can then be used as the basis for implementation, as well as a framework that can be further refined to develop and evaluate new hypertext and learning strategies in information retrieval. We choose the z specification language [30] to formalise our model partly because of previous work in specifying multi-agent systems [12,23], programming languages [11] and design and agent methodologies [10,24]. In particular, we have previously considered the requirements for the structures or formal frameworks that are necessary to provide a rigorous approach to any discipline [23], and have also defined a formal framework for hypertext systems [13] that provides an explicit formal environment for their presentation, evaluation and comparison. The specification given in this paper has been derived from applying this framework to SLHM.

1.4 Introduction to Z

The formal specification language, Z, is based on set theory and first order predicate calculus. It extends the use of these languages by allowing an additional mathematical type known as the schema type. Z schemas have two parts: the upper declarative part, which declares variables and their types, and the lower predicate part, which relates and constrains those variables. The type of any schema can be considered as the Cartesian product of the types of each of its variables, without any notion of order, but constrained by the schema's predicates. Modularity is facilitated in Z by allowing schemas to be included within other schemas.

To introduce a type in Z, where we wish to abstract away from the actual content of elements of the type, we use the notion of a given set. For example, we write $[\text{NODE}]$ to represent the set of all nodes. If we wish to state that a variable takes on some set of values or an ordered pair of values we write $x : \mathbb{P}\text{NODE}$ and $z : \text{NODE} \times \text{NODE}$, respectively. A relation type expresses some relationship between two existing types, known as the source and target types. The type of a relation with source $X$ and target $Y$ is $\mathbb{P}(X \times Y)$. A relation is therefore a set of ordered pairs. When no element from the source type can be related to two or more elements from the target type, the relation is a function. A total function ($\rightarrow$) is one for which every element in the source set is related, while a partial function ($\rightarrow$) is one for which not every element in the source is related. A sequence (seq) is a special type of function where the domain is the contiguous set of numbers from 1 up to the number
of elements in the sequence. For example, the first relation below defines a function between nodes, while the second defines a sequence of nodes.

\[
\text{Rel1} = \{(n1, n2), (n2, n3), (n3, n2)\}
\]
\[
\text{Rel2} = \{(2, n3), (3, n2), (1, n4)\}
\]

In Z, a sequence is more usually written as \(\langle n4, n3, n2 \rangle\). The domain (\(\text{dom}\)) of a relation or function comprises those elements in the source set that are related, and the range (\(\text{ran}\)) comprises those elements in the target set that are related. In the examples above, \(\text{dom \ Rel1} = \{n1, n2, n3\}\), \(\text{ran \ Rel1} = \{n2, n3\}\), \(\text{dom \ Rel2} = \{1, 2, 3\}\) and \(\text{ran \ Rel2} = \{n2, n3, n4\}\). A relation can be restricted to a particular subset of its domain or range using range restriction (\(\supset\)) or domain restriction (\(\prec\)). Similarly a relation can be domain anti-restricted by a set in such a way that the resulting relation does not contain any ordered pairs whose first element is in the restricting set. This is known as anti-domain restriction (\(\preceq\)). Anti-range restriction is written \(\supseteq\). Finally, one relation can be updated by another relation using relational overriding. The second relation can be thought of as ‘new’ information about its domain elements, overwriting any pairs in the relation whose first element is in the domain of the second relation.

For a more complete treatment of the Z language, the interested reader is referred to one of the numerous texts, such as [31]. Details of the formal semantics of Z are given in [29]. We will not consider such issues further in this paper.

2 The Conceptual Index

Before any hypertext structure can be considered, we need a set of documents that comprise the document space.

\[
[\text{DOCUMENT}]
\]

\[
\begin{array}{c}
\text{DocumentSpace} \\
\text{Documents} : \mathcal{P} \text{ DOCUMENT}
\end{array}
\]

The conceptual index is then a data structure that presents an abstract view of a document space. It defines all the concepts and the links between these concepts. This defines the index space.

We envisage the Universe consisting of a set of concepts.
Any *information space* of concern is some subset of this Universe. Further, in our Universe, a pair of concepts can be related to each other by an association called a link. A link is therefore characterised by the two concepts it connects.

\[
\text{LINK} = \text{CONCEPT} \times \text{CONCEPT}
\]

The structure, or the *state space*, of the conceptual index is represented as a collection of concepts and links where links can only exist between a pair of concepts in the space. Concepts may be linked to themselves and may also be totally isolated. Initially, the state space of any *information space* contains no concepts.

\[
\begin{align*}
\text{Concept} & \quad \text{Concepts} : \mathbb{P} \text{ CONCEPT} \\
\text{Link} & \quad \text{Links} : \mathbb{P} \text{ LINK}
\end{align*}
\]

In the schema defining the conceptual index we also define the following terms, which enable us to make our specification more readable.

(1) The *LeavingLinks* of a concept are those links which connect from that concept.
(2) The *ArrivingLinks* of a concept are those links which connect to that concept.
(3) The *Children* of a concept are all those concepts which are possible destinations reachable using the leaving links of that concept.
(4) The *Parent* of a link is the concept from which a link leaves.
(5) The *Child* of a link is the concept to which a link arrives.

\[
\begin{align*}
\text{ConceptualIndex} & \\
\text{Concept} & \\
\text{Link} & \\
\text{LeavingLinks, ArrivingLinks} : \text{CONCEPT} \rightarrow \mathbb{P} \text{ LINK} \\
\text{Children} : \text{CONCEPT} \rightarrow \mathbb{P} \text{ CONCEPT} \\
\text{Child, Parent} : \text{LINK} \rightarrow \text{CONCEPT}
\end{align*}
\]

\[
(\text{dom Links} \cup \text{ran Links}) \subseteq \text{Concepts}
\]

\[
\forall c : \text{Concepts}; l : \text{Links} \bullet
\begin{align*}
\text{LeavingLinks} c &= \{c\} \triangleleft \text{Links} \wedge \\
\text{ArrivingLinks} c &= \text{Links} \triangleright \{c\} \wedge \\
\text{Children} c &= \text{second} (\lfloor \{c\} \triangleleft \text{Links} \rfloor) \wedge \\
\text{Parent} l &= \text{first} l \wedge \\
\text{Child} l &= \text{second} l
\end{align*}
\]
Next, we define the relationship between the document space and the index space, where a concept may relate to many documents, and a document may relate to many concepts, as shown in Figure 1.

![Diagram of InitConceptualIndex](image)

![Diagram of DocumentIndex](image)

### 2.1 Updating the Conceptual Index

The construction of the conceptual index includes two groups of operations known as indexing (which we omit) and hyperization. Indexing concerns isolating, or extracting, all concepts in the information space and is typified by the conventional indexing process used in most current boolean systems. Hyperization concerns defining the relationship between a particular pair of concepts typified by the construction of traditional hard hypertext links.

In our model, concepts and links may be introduced, as long as they are not already part of the state space. However, it should be noted that no link can be added unless both of the concepts it connects currently exist in the state space.

![Diagram of AddConcept](image)
AddLink
\[ \Delta \text{ConceptualIndex} \]
\[ \exists \text{Concept} \]
\[ c_1?, c_2? : \text{CONCEPT} \]
\[ (c_1?, c_2?) \notin \text{Links} \]
\[ \{c_1?, c_2?\} \subseteq \text{Concepts} \]
\[ \text{Links}' = \text{Links} \cup \{(c_1?, c_2?)\} \]

RemoveLink
\[ \Delta \text{ConceptualIndex} \]
\[ \exists \text{Concept} \]
\[ c_1?, c_2? : \text{CONCEPT} \]
\[ (c_1?, c_2?) \in \text{Links} \]
\[ \text{Links}' = \text{Links} \setminus \{(c_1?, c_2?)\} \]

RemoveConcept
\[ \Delta \text{ConceptualIndex} \]
\[ \exists \text{Link} \]
\[ c? : \text{CONCEPT} \]
\[ c? \in \text{Concepts} \]
\[ c? \notin (\text{dom Links} \cup \text{ran Links}) \]
\[ \text{Concepts}' = \text{Concepts} \setminus \{c?\} \]

The specification presented in this section represents a high-level description of SLHM and is equivalent to the higher-level specification provided by our formal framework [13]. We now refine this specification in the next section to provide a lower-level description of the intelligent mechanisms of SLHM.

3 The Intelligent Conceptual Index

In order to introduce some intelligent mechanisms into our specification and to apply the conceptual index more efficiently in information retrieval, two read states are defined to record statistical information. These are the system read state, which records the statistical information regarding the applications of the conceptual index, and the user read state, which records general user information such as the user browsing history.
3.1 System Read State

The system read state consists of the statistical information recorded as a collection of indicators. An indicator is either accumulative, in which case it is a measure of the combined use of the system, or non-accumulative, sometimes called current, in which case it is a measure dependent on only the current user. We first consider and discuss the indicators associated with concepts.

3.1.1 Concepts

Whenever a concept is visited as a hypertext source anchor, we record that concept. This indicator is accumulative.

\[ \text{ReadConceptsVisited} \equiv \text{ConceptsVisited : bag CONCEPT} \]

For each concept, we also record a measure of the likelihood that it is relevant to the current user's information needs, known as the activation, which is defined as a rational in the interval \([0, 1]\). This indicator is non-accumulative, which means that it is only valid and meaningful for the current information retrieval session. If the activation of a concept is equal to 0, it suggests that the concept is totally irrelevant to the user's needs, and if equal to 1, it suggests that the concept is exactly appropriate. We define \([RAT^1_0]\) as the rationals between 0 and 1, assuming basic arithmetic operations are applicable [33].

\[ \text{ReadConceptActivation} \equiv \text{ConceptActivation : CONCEPT} \rightarrow RAT^1_0 \]

We define \textit{ReadConcepts} as the schema that includes all the concepts of the systems along with this other information.

\[ \text{ReadConcepts} \equiv \begin{align*}
\text{Concept} \\
\text{ReadConceptsVisited} \\
\text{ReadConceptActivation}
\end{align*} \]

\[ \text{dom ConceptsVisited} \subseteq \text{Concepts} \]

\[ \text{dom ConceptActivation} = \text{Concepts} \]

3.1.2 Links

Whenever a link is used during an information retrieval session we record that link in the accumulative indicator \textit{Links Visited}. For each link, we also record a measure of its popularity, known as the weight. The accumulative indicator \textit{LinkWeight} is defined as a rational number in the interval \((0, 1]\) so that the
weight of a link cannot take the value 0. This indicator is totally dependent on other indicators. If the weight of a link is 1, it suggests that the link is the only link leaving some concept whereas as the weight of a link approaches 0, it suggests that the link is so unpopular that it may never be used.

\[
\begin{align*}
\text{ReadLinksVisited} \\
\text{LinksVisited} : \text{bag LINK}
\end{align*}
\]

\[
\begin{align*}
\text{ReadLinkWeight} \\
\text{LinkWeight} : \text{LINK} \rightarrow \mathbb{RAT}_0^1 \\
0 \notin \text{ran LinkWeight}
\end{align*}
\]

\[
\begin{align*}
\text{ReadLinks} \\
\text{Link} \\
\text{ReadLinksVisited} \\
\text{ReadLinkWeight} \\
\text{dom LinksVisited} \subseteq \text{Links} \\
\text{dom LinkWeight} = \text{Links}
\end{align*}
\]

Note that the variable ConceptsVisited records a concept every time it is used as a hypertext source anchor in a session and consequently its domain is a subset of all system concepts. This is in contrast to ConceptActivation, which maps every system concept to a rational between 0 and 1. Analogously, the domain of LinksVisited is a subset of all system links whilst LinkWeight maps every system link to a rational. In response, we define the frequency of a system link or concept to be the number of times it has been visited.

\[
\begin{align*}
\text{FrequencyFunctions} \\
\text{ReadConceptsVisited} \\
\text{ReadLinksVisited} \\
\text{ConceptualIndex} \\
\text{CFrequency} : \text{CONCEPT} \rightarrow \mathbb{N} \\
\text{LFrequency} : \text{LINK} \rightarrow \mathbb{N} \\
\text{CFrequency} = (\lambda x : \text{Concepts} \cdot 0) \oplus \text{ConceptsVisited} \\
\text{LFrequency} = (\lambda x : \text{Links} \cdot 0) \oplus \text{LinksVisited}
\end{align*}
\]

The functions CFrequency and LFrequency in the schema above can be defined by overriding the function that maps all system links and concepts to zero with the respective bag representations. In this way, these functions are defined for all system links and concepts, and return their frequency. In future, we
may treat these data structures as either bags or functions (by including this schema), depending on our purpose.

The reader should be clear that CFrequency and LFrequency are simply functional versions of the bags Concepts Visited and Links Visited and are used so that we can apply this data structure to all system concepts and links.

3.1.3 The System Read State

Combining the records about concepts and links defines the system read state of the conceptual index. The indicators in the system read state are related to each other as follows.

- The weight of any link is given by the frequency of the link plus 1 divided by the sum of the number of leaving links from the parent concept and the frequency of the parent concept.
- The number of times a concept has been visited as the hypertext source anchor is equal to the sum of the number of times each of its leaving links have been visited.
- As long as there are links leaving a concept, the sum of the weights of all the links leaving a concept is equal to 1.

\[
\begin{align*}
\text{SystemReadState} & \\
\text{ReadConcepts} & \\
\text{ReadLinks} & \\
\text{ConceptualIndex} & \\
\text{FrequencyFunctions} & \\
\forall c_1, c_2 : \text{CONCEPTS} \mid (c_1, c_2) \in \text{Links} \bullet \\
\text{LinkWeight} (c_1, c_2) & = \frac{1 + (\text{LFrequency} (c_1, c_2))}{\#(\text{LeavingLinks} c_1) + (\text{CFrequency} c_1)} \\
\forall c : \text{Concepts} \bullet \text{CFrequency} c = \\
\text{subset} (\text{mapset} \ \text{LFrequency} (\text{LeavingLinks} c)) & \\
\forall c : \text{dom} \text{Links} \bullet \text{subset} (\text{mapset} \ \text{LinkWeight} (\text{LeavingLinks} c)) = 1
\end{align*}
\]

The generic function mapset takes a function, and applies it to every member of a set.

\[
\begin{align*}
\text{mapset} & : (X \to Y) \to \mathcal{P} X \to \mathcal{P} Y \\
\forall f : X \to Y; \ zs : \mathcal{P} X; \ z : X \mid z \not\in zs \land (zs \cup \{z\}) \subseteq \text{dom} f \bullet \\
\text{mapset} f \ \{\} & = \{\} \land \\
\text{mapset} f \ (\{z\} \cup zs) & = \{f(z)\} \cup \text{mapset} f \ zs
\end{align*}
\]
The function sumset simply sums the elements of a set of rationals.

```
sumset : \( \mathbb{F} R A T_{0}^{1} \rightarrow R A T_{0}^{1} \)
\[ \forall q : R A T_{0}^{1}; \quad sq : \mathbb{F} R A T_{0}^{1}, q \notin sq \cdot \]
\[ \text{sumset} \{\} = 0 \land \]
\[ \text{sumset} (\{q\} \cup sq) = q + \text{sumset} \, sq \]
```

Initially, in the read state, there is no statistical information. No links or concepts have been visited, the activation of all system concepts is 0, and the weight of any link is the reciprocal of the number of leaving links of the parent concept of that link and so, initially, all the leaving links of any concept have equal weights.

```
InitSystemReadState

SystemReadState

ConceptsVisited = []
LinksVisited = []
ConceptActivation = (\( \lambda \; c : \text{Concepts} \cdot 0 \))
LinkWeight = \{\( c_{1}, c_{2} : \text{CONCEPT} \mid (c_{1}, c_{2}) \in \text{Links} \cdot (c_{1})\cdot 1, \#(\text{LeavingLinks} \, c_{1})\})
```

### 3.2 User Read State

In SLHM, the *accumulative history* records all past users' visited concepts, and the *current history* records the current user's set of visited concepts (known as their browsing history). The use of the current history is standard and enables, for example, the user to re-visit any concepts within the current session. The *history* of a session refers to both the accumulative and current histories, and is used by the system to infer the current user's information needs. For example, by comparing the current history with the accumulative history, the system may obtain a more accurate user model of the current information user, and thus reason about his or her next move. This feature of the model is not specified in this paper. For more details, the interested reader is asked to consult [19,20].

```
AccumulatedHistory

AccHistory : \( \mathbb{F}(\mathbb{F} \text{CONCEPT}) \)
```

```
CurrentHistory

CurrHistory : \( \mathbb{F} \text{CONCEPT} \)
```
In addition, the user may have a current position in the conceptual index known as the current concept and a destination concept known as the next concept. These indicators do not contribute to the intelligence of SLHM but are included in the user read state for the dynamic description of the user's movement. Note too, that the next concept the user visits can only be defined if the current concept has been defined. (In other words, the user needs a position in order to choose a link).

We have found it useful in this and other specifications — as in the previous schema — to be able to assert that an element is optional. The following definitions provide for a new type optional T for any existing type T. In addition, we define predicates defined and undefined to test whether an element of optional T is defined or not, and an operation the to extract the T element from a defined member of optional T, which are used later in this paper.

\[
\text{optional}[X] \equiv \{zs : \mathbb{P} X \mid \# zs \leq 1\}
\]

\[
[X]
\]

\[
\begin{align*}
\text{defined } & : \mathbb{P}(\text{optional } [X]) \\
\text{undefined } & : \mathbb{P}(\text{optional } [X]) \\
\text{the } & : \text{optional } [X] \rightarrow X \\
\forall zs : \text{optional } [X] & \bullet \text{defined } zs \leftrightarrow \# zs = 1 \\
& \quad \text{undefined } zs \leftrightarrow \# zs = 0 \\
& \quad (\text{defined } zs) \Rightarrow \text{the } zs = (\mu z : X \mid z \in zs)
\end{align*}
\]

The user read state comprises the history and the position.

\[
\text{UserReadState}
\]

\[
\begin{align*}
\text{History} \\
\text{Position}
\end{align*}
\]

Initially, there is no statistical information.
3.3 The Soft-Link Hypertext Model

The combination of system read state and user read state defines the read state, referred to as a session. The read state constitutes the intelligence in the model; it is applied as an auxiliary navigation tool for more efficient information retrieval operations. In the read state, the accumulative indicators are the frequencies of the links and concepts, the weight of the links and the accumulative history, while the current indicators are concept activations, the current history, and the current and next positions (if they are known). The combined history of past and present users is the set of all concepts that have been visited. Furthermore, although any concept that has been visited during the current session should have an activation value of 1, the converse is not necessarily true: there may exist some circumstances when the activation of a concept may reach 1 without ever being visited. Finally, the position of a user must be within the information space.

\[
\begin{array}{l}
\text{InitUserReadState} \\
\text{UserReadState} \\
\text{AccHistory} = \{} \\
\text{CurrHistory} = \{}
\end{array}
\]

\[
\begin{array}{l}
\text{InitSoftLinkHypertext} \\
\text{SoftLinkHypertext} \\
\text{InitSystemReadState} \\
\text{InitUserReadState}
\end{array}
\]

Before the first user starts, all session indicators are set at their initial value. From this moment, the accumulative indicators start collecting statistical information about the operations of the conceptual index. The life-span of these indicators is the same as that of the system, whereas the life-span of non-accumulative indicators is that of one information retrieval session.
4 Applications of the Soft-Link Hypertext Model

In this section, we detail the application of SLHM in information retrieval. This is concerned with the user’s movement in the information space, and how the statistical indicators of the conceptual index are updated. Such read applications do not affect the structure of the conceptual index.

\[
\Delta \text{SoftLinkHypertext} \\
\text{SoftLinkHypertext} \\
\text{SoftLinkHypertext}' \\
\Xi \text{ConceptualIndex}
\]

4.1 Starting an Information Retrieval Session

When an information retrieval session starts, all non-accumulative indicators are reset, whilst accumulative indicators remain unchanged. The user does not have a position in the information space.

\[
\text{UserStartsNewSession} \\
\Delta \text{SoftLinkHypertext} \\
\Xi \text{ReadLinks} \\
\Xi \text{ReadConceptsVisited} \\
\Xi \text{AccumulatedHistory} \\
\text{ConceptActivation}' = (\lambda c : \text{Concepts} \cdot 0) \\
\text{CurrHistory}' = \{\} \\
\text{undefined CurrentConcept}'
\]

4.2 Starting a New Trail

Once a session is started, a concept in the index space may be chosen from which to start the information retrieval session. This corresponds to a conventional boolean search operation: the user makes an information request explicitly and the system responds by highlighting the requested concept in the index space; and the user retrieves a document in the document space in which the information requested is included. It should be noted that this operation does not affect any statistical information of the read state, and it is only the position that changes. We refer to such moves as starting a new trail.
UserStartsNewTrail

\[ \Delta \text{SoftLinkHypertext} \]
\[ \exists \text{SystemReadState} \]
\[ \exists \text{History} \]

\text{NewStartingConcept? : CONCEPT}

\begin{align*}
\text{NewStartingConcept?} &\in \text{Concepts} \\
\text{the CurrentConcept'} &\equiv \text{NewStartingConcept?} \\
\text{undefined NextConcept'}
\end{align*}

¿From this point on, the user has essentially two types of movement available, each representing a different information retrieval methodology.

(1) The user can repeat the above operation and start another new trail. In this case, another boolean search can be initialised, or the user may just re-visit a concept of the current history.

(2) The user can treat the current concept as a hypertext source anchor and move to a hypertext destination concept by following one of the soft links leaving the current concept. Here, the user utilises the soft links provided by the conceptual index, and moves around in the information space by following these links. In this case, some indicators of the read state are updated. This process is known as browsing.

4.3 Browsing

Browsing is defined as a process of moving to a specific concept in the information space by following a soft link from a concept. In this situation, the user treats the current concept as a hypertext source anchor, specifies a soft link leaving the current concept and moves to the new concept to which the link connects. To move in such a manner, the user must choose a link. There are three possible scenarios.

(1) The user has not started a trail, and so does not have a position in the information space. In other words, no concept can be currently used as a hypertext source anchor. In this case, there is no change to any of the session indicators, and a report is given.

Report \( ::= \text{No\_Such\_Leaving\_Link} \mid \text{Must\_Start\_Trail\_Before\_Browsing} \)
(2) The user has a position at a concept, but the link specified is not one of the leaving links of that concept. There is no change to any of the session indicators, and a report is given.

(3) The user has a current position at a concept and the specified link is a leaving link of the current concept. In this case, the user successfully makes a legitimate browsing move. The next concept becomes defined; it is the child of the user-specified link.

We now describe how the statistical information is updated as a consequence of a browsing mode. The frequency of the hypertext source anchor (current concept), is incremented along with the frequency of the specified leaving link. The activation of the current concept is set to 1. If there is more than one leaving link from the current concept, the weights of all these links are updated to maintaining the state invariant definition of weight as defined in the SystemReadState. In addition, the current concept is added to the current history, which maintains the first invariant from the SoftLinkHypertext schema.
System Backward Learning

\[ \begin{align*}
\text{Concepts Visited}' &= \text{Concepts Visited} \cup \{ \text{the Current Concept} \} \\
\text{Links Visited}' &= \text{Links Visited} \cup \{ (\text{the Current Concept}, \text{the Next Concept}) \} \\
\text{Concept Activation}' &= \text{Concept Activation} \oplus \{ (\text{the Current Concept}, 1) \} \\
\forall (c_1, c_2) : \text{Links} \bullet \text{Link Weight}'(c_1, c_2) &= \frac{1 + (L \text{Frequency}'(c_1, c_2))}{\text{#(Leaves Links c_1) + (C \text{Frequency}'c_1)}} \\
\text{Curr History}' &= \text{Curr History} \cup \{ \text{the Current Concept} \}
\end{align*} \]

Then the new activation of the current node spreads to the children of the current concept as described by the following algorithm.

For the child of each leaving link of the activated concept, add the activation of that child to the sum of the input activation of each of the arriving links to that child concept, where the input activation of any link is defined as the product of its weight and the concept activation of its parent concept. If this value is greater than 1 then set the activation to 1.

\[ \begin{align*}
\text{Forward Activation Spreading} \\
\Delta \text{Soft Link Hypertext} \\
\exists \text{Read Links} \\
\exists \text{Read Concepts Visited} \\
\exists \text{User Read State}
\end{align*} \]

let \( \text{Input Activation} \) == \( \lambda c_1, c_2 : \text{Concepts} \mid (c_1, c_2) \in \text{Links} \bullet \text{Link Weight}'(c_1, c_2) \bullet \text{Concept Activation}' c_1 \) +
\( \{ c : \text{Concepts} \mid c \in \text{Children (the Current Concept)} \bullet \text{Concept Activation}' c \} + \)
\sum \text{set ( map set Input Activation (Arriving Links c )))} \)

The schema above makes use of the intermediate concept of the input activation of a link, as defined above, in order to make the definition of the forward activation spreading algorithm more readable. This is a local variable used simply to improve the readability of the schema and the reader should attach no particular significance to the name.

At the end of each browsing move, the system provides a sequence of concepts as output, listed in the order of their activations. This is used as the on-line navigation guide for the current user to search and retrieve information more efficiently.
The function `sortedperms` simply takes a set of elements and an ordering function and returns the set of all possible sorted permutations. (It is defined to return a set of orderings rather than one ordering because the ordering function is not necessarily injective. In this specification, many concepts will have an activation of 1.)

\[
[X] \quad \text{sortedperms} : (\mathcal{P} X) \to (X \to \mathcal{R} \mathcal{T}_1) \to (\mathcal{P}(\text{seq } X))
\]
\[
\forall f : (X \to \mathcal{R} \mathcal{T}_1); \; zs : \mathcal{P} X \mid zs \subseteq \text{dom } f \bullet \\
\text{sortedperms} \; zs \; f = \{ s : \text{seq } X; \; x_1, x_2 : X \mid \text{ran } s = zs \land \\
\langle x_1, x_2 \rangle \in s \Rightarrow f \; x_1 \geq f \; x_2 \bullet s \}
\]

Finally, the user is transferred to the child of the specified link, which becomes the current concept, while the next concept is now undefined.

\[ UserMakesMove \]
\[ \Delta \; \text{SoftLinkHypertext} \]
\[ \exists \; \text{SystemReadState} \]
\[ \exists \; \text{History} \]
\[ \text{CurrentConcept}' = \text{NextConcept} \]
undefined \text{NextConcept}'

The total move operation is then given by either of the above three scenarios.

\[ Browse \triangleq ( \text{UserSelectsNextConcept} \land \\
( \text{SystemBackwardLearning} \circ \text{ForwardActivationSpreading} \circ \\
\text{OutputConceptOrder} \circ \text{UserMakesMove} ) ) \lor \\
\text{UserSelectsNextConceptError1} \lor \text{UserSelectsNextConceptError2} \]

### 4.4 Quitting

A user may finish an information retrieval session at any time, in which case the local history is added to the accumulative history.
5 Authoring the Soft-Link Hypertext Model

Any changes made to the state space of the conceptual index through indexing or hyperization should not invalidate the statistical information collected by any accumulative indicators. We consider each of the four authoring cases defined in Section 2.1 in turn, refining the previously defined schemas. First, a new concept is given an activation of 0.

\[
\text{Update AddConcept} \\
\Delta \text{SystemReadState} \\
\exists \text{ReadLinks} \\
\exists \text{ReadConceptsVisited} \\
\text{AddConcept} \\
\text{ConceptActivation}' = \text{ConceptActivation} \cup \{(c?, 0)\}
\]

When a new link is introduced, its frequency is set to 0 and its weight is given the appropriate value.

\[
\text{Update AddLink} \\
\Delta \text{SystemReadState} \\
\exists \text{ReadConcepts} \\
\exists \text{ReadLinksVisited} \\
\text{AddLink} \\
\text{LFrequency}' = \text{LFrequency} \cup \{((c_1?, c_2?), 0)\} \\
\text{LinkWeight}' = \text{LinkWeight} \cup \\
\{(c_1?, c_2?), \left\{ \frac{1}{\text{#LeavingLinks } c_1? + 1 + (\text{CFrequency } c_1?)} \right\} \} \oplus \\
\{l : \text{LeavingLinks } c_1? \cdot (l, \left\{ \frac{1}{\text{#LeavingLinks } c_1? + 1 + (\text{CFrequency } c_1?)} \right\})\}
\]

When a link is removed, the frequency of that link is subtracted from the frequency of its parent concept.
When we remove a concept, we must delete any occurrence of it from the accumulative history in the system read state.

6 Evaluating the Soft-Link Hypertext Model

6.1 Experimentation

The soft-link hypertext mechanism described above has been implemented on SuperBook [21] to produce a system known as the Enhanced SuperBook. The SuperBook system makes use of the structure of a large document to display query term hits in context. The table of contents for a book or manual are shown in a hierarchy on the left-hand side of the display, and the full text of a page or section is shown on the right-hand side. The user can manipulate the table of contents to expand or contract the view of sections and subsections.

Five main metrics were used to evaluate the system as follows.

- Recall — defined as the percentage of relevant items retrieved compared with the total number of relevant items in the collection.
- Precision — defined as the percentage of relevant items compared with the total number of items retrieved in the collection.
• Response Time — defined as the time needed for a user-system interaction.
• Search Time — the total time the user needs to obtain a concrete piece of information with an Information Retrieval (IR) system.
• User Actions — the number of IR jobs the user needs to fulfill in order to obtain a concrete piece of information.

In a series of experiments using 20 chemists and a database of a year's publications in chemistry journals, the Enhanced SuperBook with all the features of the soft-link hypertext was tested against the SuperBook. Four experiments were designed to measure the cost, effectiveness and benefits of the system. While the Superbook database itself remained unchanged, the full-text index was further extended to a conceptual index for the enhanced SuperBook by applying the automatic index formulation mechanism to the full text SuperBook.

• Experiment 1 evaluated the automatic indexing and formulation mechanism used in the soft-link hypertext system.
• Experiment 2 evaluated the acceptability of, and the user-overload caused by the soft-link hypertext browser.
• Experiment 3 compared the soft-link hypertext system with a boolean search system, and evaluated the role of the intelligent mechanisms in IR.
• Experiment 4 evaluated different IR models in an integrated environment.

The model was extensively tested in these experiments and showed that automatic generation of the conceptual index structure was possible in an acceptable cost of time and money, that the soft-link hypertext browser was acceptable to the user (the increased cognitive overload caused by the introduction of additional features was fully justified by the overall performance of the system) and that the user searches and information retrievals were themselves much faster and easier. Indeed, the learning mechanism could gradually increase the speed of this process.

These experiments show that the soft-link hypertext, which provides automatic formulation of the information structure, its self-adjustment and evolution based on the statistical information, machine learning applications to accumulate information about past experiences and user-centred services for information, can be used to potentially enourmous benefit.

6.2 Related Work

The soft-link hypertext model also has something in common with some existing systems such as the notion of hyperindices, as proposed by Bruza [3], which offer a new mechanism for supporting search in hypermedia. Bruza developed a framework in which the effectiveness of such hypertext indexes
could be judged, and which could be used as the basis for the development of a prototype system. However, this work has not been tested as extensively as the soft-link hypertext model. In subsequent work Bruza [4], along with van Linder, attempted to integrate non-monotonic reasoning into information retrieval. Searching in this system is modelled as navigating through an information space that is referred to as a hyperindex. However, this hyperindex system has not been fully implemented and tested. Furthermore, while the work incorporates some interesting ideas, it is not as broad in scope as the soft-link hypertext model. SLHM provides a formal model for the creation and maintenance of an involving hypertext structure, and it has also been tested in a real information retrieval environment with good results. Finally, work by Wondergem et al. [34] presents a new formalisation of index expressions, which are used in IR to characterise document contents. Again, the work is formal in nature, and the authors do not suggest how the theoretical model can be used in the design of real IR systems.

7 Conclusions

This paper has presented a Z specification of a new information retrieval model called the soft-link hypertext model. SLHM differs from other current information retrieval models by its conceptual index data structure, which allows information organisation and retrieval to be achieved more efficiently and effectively. The conceptual index is constructed automatically according to well-defined rules. Subsequently, it can improve itself by collecting statistical data and also by applying automatic learning algorithms to this data in order to produce a more detailed map of the hypertext structure. These new features solve many of the problems of current information retrieval systems; consequently this new model can become the paradigm for a new generation of hypertext systems.

Our specification has provided a formal, precise and unambiguous account of SLHM, which can be used as the basis from which sound implementations can be designed and built. This specification was derived from applying the formal framework described in [13] and we have gained confidence in the validity and utility of the framework. In addition, if we use this framework to specify other existing large-scale and reconfigurable information systems then we can more readily incorporate our conceptual index and its related learning algorithms into them. Finally, we believe that the SLHM specification provides an environment for research on further hypertext and learning strategies. In particular, our own future work will concentrate on installation of intelligent mechanisms on other aspects of the model, for example to extend the document space to multimedia applications.
Acknowledgements

Special thanks to Mark Hewitt who has animated this specification in PiZA and was able to point out a couple of errors in the original specification which have now been rectified. In addition, the specification in this paper was checked using the fuzz package [30].

References


