In ZUM'97: Proceedings of the Tenth International Conference on Z for Users
Bowen, Hinchey and Till (eds.), Lecture Notes in CS, 1212, 225-240, Springer-Verlag, 1997.

A Z Specification of the Soft-Link Hypertext Model

Mark d'Inverno[1] and Michael Hu[2]

[1] School of Computer Science, University of Westminster, 115 New Cavendish Street, London, W1M 8JS, UK. Email: dinverm@westminster.ac.uk
[2] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore 639798, Email: emjhu@ntuvax.ntu.ac.sg

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. Its core is the adoption of a data structure called the conceptual index, which allows hypertext structure to be built automatically upon conventional Boolean systems. The functionality of resulting systems using this approach is then extended from Boolean search to more sophisticated information retrieval applications, including associative searches and information browsing. Compared with other similar projects, SLHM has the following three major advantages. First, the conceptual index is automatically formulated. Second, powerful neural learning mechanisms are applied to the conceptual index, thereby improving its efficiency and applicability. Third, machine intelligence installed on the conceptual index can be utilised for online assistance during navigating and information browsing. This specification has been developed by application of an existing formal framework for specifying hypertext systems. The specification presented here provides: a formal account of the state and operations of this new model; a sound basis for instantiations of the model to be built; a case study in the application of an existing formal framework; and an environment in which further refinements of new learning and hypertext strategies can be presented and evaluated.

1 Introduction

A hypertext system is a system which attempts to superimpose an external structure on data, in order that it can be accessed efficiently according to different users' 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 [14], three of which are as follows.

1. Designing and constructing these information webs is expensive, laborious and tedious.

2. These manually built-up webs often confuse or mislead the users because there is no transparency nor any appropriate guidance facilities.
3. Once these inter-connected information webs are built, they usually exist as an unalterable structure. Consequently it is very difficult, if not impossible, to amend and improve these webs.

In response to these problems, a new design of the information web has been proposed [17]. It is called the conceptual index. 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 our model to formulate intelligent indicators. These indicators are utilised for online assistance during information search and browsing.

1.1 Background

Some solutions to the problems given above have been proposed [5,7,14]. In particular, different types of semantic networks have been considered and applied as a supplement for the conventional Boolean search [6,21,23]. However, these projects are only successful in providing alternative tools for refining the user's queries for information search and retrieval, but not tailored for information browsing from one document to another [23]. In our model, the associations between a pair of keywords are accommodated, so that the functionality of our model can be fully extended from the Boolean search to some 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 [17].

The algorithms used in SLHM are based on connectionist networks [1,3,13,16]. 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 [22]. Earlier attempts to apply the connectionist network methodology, include Mozer's first application to information retrieval [26], Cohen and Kjeldsen's constrained spreading activation in their semantic network for matching research funding agencies and research topics [4], Lelu's application of spreading activation in image databases [19] 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 [16]. Compared with these projects, SLHM accommodates significantly more complicated information databases, and is potentially extendible to multimedia environments. Further, 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 online assistance during information browsing [17].

1.2 An Overview of SLHM

It is the case that any information retrieval system can be treated as a self-inclusive Information Space. SLHM 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 [15]; 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 one another to formulate a new data structure called the conceptual index. Consequentially, 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.

In most current systems the links are defined and assembled manually, and so liable to human prejudice and ignorance. In addition, two concepts are related by a link without an explicit motivation as to it was created. Lastly, 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. Further, the motivation for the creation of each link can be understood since these rules are made explicit and available for the users once the conceptual index is formulated. Furthermore, powerful 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 online 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 the users' mobility in an Information Space, and make the application of the Index Space more flexible and powerful.

1.3 The Specification

In this paper we provide a formal specification of SLHM which represents a formal, concise and readable definition of the *SLHM* and its applications. The specification can then be used as the basis for implementation, as well as a framework which can be further refined to develop and evaluate new hypertext and learning strategies in information retrieval. We choose the language Z [25] to formalise our model because of experience of the using it in a number of fields including interactive conferencing systems [8], multi-agent systems systems [11] and design methodologies [9]. In addition, we have previously considered the requirements for the structures or *formal frameworks* that are necessary to provide a rigorous approach to any discipline [20]. In particular we have presented a formal framework for hypertext systems [12], which provides an explicit formal environment for the presentation, evaluation and comparison of hypertext systems. The specification given in this paper has been derived from applying this framework to SLHM. This approach of deriving formal specifications from formal frameworks has also been considered in multi-agent systems [10,11].

1.4 Overview of the Paper

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.

2 The Conceptual Index

The conceptual index is a data structure which presents an abstract view of a Document Space. It defines all the concepts and the links between these concepts. We envisage the Universe consisting of a set of concepts.

$$[\mathit{CONCEPT}]$$

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.

$$\mathit{LINK} == \mathit{CONCEPT} \times \mathit{CONCEPT}$$

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.

Concept

$$\mathit{Concepts} : \mathbb{P}\,\mathit{CONCEPT}$$

Link

$$\mathit{Links} : \mathbb{P}\,\mathit{LINK}$$

ConceptualIndex

$$\begin{array}{l} \mathit{Concept} \\ \mathit{Link} \\ \hline (\mathrm{dom}\ \mathit{Links} \cup \mathrm{ran}\ \mathit{Links}) \subseteq \mathit{Concepts} \end{array}$$

InitConceptualIndex

$$\begin{array}{l} \mathit{ConceptualIndex} \\ \hline \mathit{Concepts} = \{\ \} \end{array}$$

We next 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.

ReadabilityFunctions

$$\begin{array}{l} \mathit{ConceptualIndex} \\ \mathit{LeavingLinks}, \mathit{ArrivingLinks} : \mathit{CONCEPT} \nrightarrow \mathbb{P}\,\mathit{LINK} \\ \mathit{Children} : \mathit{CONCEPT} \nrightarrow \mathbb{P}\,\mathit{CONCEPT} \\ \mathit{Child}, \mathit{Parent} : \mathit{LINK} \nrightarrow \mathit{CONCEPT} \\ \hline \forall\, c : \mathit{Concepts};\ l : \mathit{Links} \bullet \\ \qquad \mathit{LeavingLinks}\ c = \{c\} \lhd \mathit{Links} \land \\ \qquad \mathit{ArrivingLinks}\ c = \mathit{Links} \rhd \{c\} \land \\ \qquad \mathit{Children}\ c = \mathit{second}\ (\!|\ \{c\} \lhd \mathit{Links}\ |\!) \land \\ \qquad \mathit{Parent}\ l = \mathit{first}\ l \land \\ \qquad \mathit{Child}\ l = \mathit{second}\ l \end{array}$$

The above schema introduces five functions which are used later in presenting a readable specification. Whenever we wish to use these functions the above schema is included.

2.1 Updating the Conceptual Index

The construction of the conceptual index includes two groups of operation known as indexing 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.

AddConcept

$\Delta ConceptualIndex$
$\Xi Link$
$c? : CONCEPT$

$c? \notin Concepts$
$Concepts' = Concepts \cup \{c?\}$

AddLink

$\Delta ConceptualIndex$
$\Xi Concept$
$c_1?, c_2? : CONCEPT$

$(c_1?, c_2?) \notin Links$
$\{c_1?, c_2?\} \subseteq Concepts$
$Links' = Links \cup \{(c_1?, c_2?)\}$

Associated with each adding operation is a corresponding removing operation. A link can be removed from the state space without affecting other links and concepts. However, no concept can be removed unless all leaving and arriving links to the concept have been removed first.

RemoveLink

$\Delta ConceptualIndex$
$\Xi Concept$
$c_1?, c_2? : CONCEPT$

$(c_1?, c_2?) \in Links$
$Links' = Links \setminus \{(c_1?, c_2?)\}$

RemoveConcept

$\Delta ConceptualIndex$
$\Xi Link$
$c? : CONCEPT$

$c? \in Concepts$
$c? \notin (\mathrm{dom}\ Links \cup \mathrm{ran}\ Links)$
$Concepts' = Concepts \setminus \{c?\}$

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 [12]. We now refine this specification in the next three sections 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.

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

ReadConceptsVisited

$ConceptsVisited : \mathrm{bag}\ CONCEPT$

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. The activation is defined as a rational in the interval $[0, 1]$. This indicator is non-accumulative which means that is 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 $[RAT_0^1]$ as the rationals between 0 and 1, assuming basic arithmetic operations are applicable [27].

ReadConceptActivation

$ConceptActivation : CONCEPT \nrightarrow RAT_0^1$

We define *ReadConcepts* as the schema which includes all the concepts of the systems along with this other information.

ReadConcepts

Concept
ReadConceptsVisited
ReadConceptActivation

$\mathrm{dom}\ ConceptsVisited \subseteq Concepts$
$\mathrm{dom}\ ConceptActivation = Concepts$

Links Whenever a link is used during an information retrieval session we record that link in the accumulative indicator *LinksVisited*. For each link, we also record a measure of its popularity known as the weight. The accumulative indicator *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 it may never be used.

ReadLinksVisited

$LinksVisited : \mathrm{bag}\ LINK$

ReadLinkWeight

$LinkWeight : LINK \nrightarrow RAT_0^1$

ReadLinks

Link
ReadLinksVisited
ReadLinkWeight

$\mathrm{dom}\ LinksVisited \subseteq Links$
$\mathrm{dom}\ LinkWeight = Links$

Note, that the variables *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.

FrequencyFunctions

ReadConceptsVisited
ReadLinksVisited
ConceptualIndex

$$\mathit{ConceptFrequency} : \mathit{CONCEPT} \rightarrow\!\!\!\!\!+ \mathbb{N}$$
$$\mathit{LinkFrequency} : \mathit{LINK} \rightarrow\!\!\!\!\!+ \mathbb{N}$$

$$\mathit{ConceptFrequency} = (\lambda\, x : \mathit{Concepts} \bullet 0) \oplus \mathit{ConceptsVisited}$$
$$\mathit{LinkFrequency} = (\lambda\, x : \mathit{Links} \bullet 0) \oplus \mathit{LinksVisited}$$

The functions *ConceptFrequency* and *LinkFrequency* in the schema above can be defined by overriding the function which 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 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.

ReadSystemState

ReadConcepts
ReadLinks
ConceptualIndex

ReadabilityFunctions
FrequencyFunctions

$$\forall\, c_1, c_2 : \mathit{CONCEPTS} \mid (c_1, c_2) \in \mathit{Links} \bullet$$
$$\qquad \mathit{LinkWeight}\ (c_1, c_2) = \frac{1+(\mathit{LinkFrequency}\ (c_1,c_2))}{\#(\mathit{LeavingLinks}\ c_1)+(\mathit{ConceptFrequency}\ c_1)}$$
$$\forall\, c : \mathit{Concepts} \bullet \mathit{ConceptFrequency}\ c =$$
$$\qquad \mathsf{sumseq}\ (\ \mathsf{mapsettoseq}\ \ \mathit{LinkFrequency}\ (\mathit{LeavingLinks}\ \ c))$$
$$\forall\, c : \mathsf{dom}\ \mathit{Links} \bullet$$
$$\qquad \mathsf{sumseq}\ (\ \mathsf{mapsettoseq}\ \ \mathit{LinkWeight}\ (\mathit{LeavingLinks}\ \ c)) = 1$$

The generic function mapsettoseq takes a function, and applies it to every member of a set creating a sequence. This is to protect against the partial function not being injective. The function sumseq simply sums the elements of a sequence of rationals.

$$
\begin{array}{l}
[X, Y] \quad \mathsf{mapsettoseq} : (X \mathrel{+\!\!\!\rightarrow} Y) \rightarrow \mathbb{P}\, X \mathrel{+\!\!\!\rightarrow} \mathsf{seq}\ Y \\
\hline
\forall f : X \mathrel{+\!\!\!\rightarrow} Y;\ xs : \mathbb{P}\, X;\ x : X \mid (xs \cup \{x\}) \subseteq \mathsf{dom}\, f \bullet \\
\qquad \mathsf{mapsettoseq}\ f\ (\{x\} \cup xs) = \langle f(x) \rangle \frown \mathsf{mapsettoseq}\ f\ xs \land \\
\qquad \mathsf{mapsettoseq}\ f\ \{\} = \langle\rangle
\end{array}
$$

$$
\begin{array}{l}
\mathsf{sumseq} : \mathsf{seq}\ RAT_0^1 \mathrel{+\!\!\!\rightarrow} RAT_0^1 \\
\hline
\forall n : RAT_0^1;\ ms, ns : \mathsf{seq}\ RAT_0^1 \bullet \\
\qquad \mathsf{sumseq}\ \langle n \rangle = n \land \\
\qquad \mathsf{sumseq}\ (ms \frown ns) = \mathsf{sumseq}\ ms + \mathsf{sumseq}\ ns
\end{array}
$$

The initial state of the read state is where 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.

$$
\begin{array}{l}
\textit{InitReadSystemState} \\
\hline
\textit{ReadSystemState} \\
\hline
\textit{ConceptsVisited} = [\![]\!] \\
\textit{LinksVisited} = [\![]\!] \\
\textit{ConceptActivation} = \{c : \textit{Concepts} \bullet (c, 0)\} \\
\textit{LinkWeight} = \{c_1, c_2 : \textit{CONCEPT} \mid \\
\qquad (c_1, c_2) \in \textit{links} \bullet ((c_1, c_2), \frac{1}{\#(\textit{LeavingLinks}\ c_1)})\}
\end{array}
$$

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 so, reason about their next move. This feature of the model is not specified in this paper. For more details, the interested reader is asked to consult [17,18].

AccumulateHistory

$AccHistory : \mathbb{F}(\mathbb{F}\ CONCEPT)$

CurrentHistory

$CurrHistory : \mathbb{F}\ CONCEPT$

History

$AccumulateHistory$
$CurrentHistory$

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 which 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).

Position

$CurrentConcept : \mathsf{optional}\ [CONCEPT]$
$NextConcept : \mathsf{optional}\ [CONCEPT]$

$\mathsf{undefined}\ CurrentConcept \Rightarrow \mathsf{undefined}\ NextConcept$

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.

$$\mathsf{optional}\ [X] == \{xs : \mathbb{P}\, X \mid \#\, xs \ \leq\ 1\}$$

[[]

$X]\ \mathsf{defined}\ \_ : \mathbb{P}(\ \mathsf{optional}\ [X])$
$\mathsf{undefined}\ \_ : \mathbb{P}(\ \mathsf{optional}\ [X])$
$\mathsf{the} : \mathsf{optional}\ [X] \rightarrow\!\!\!\!\!\!+ X$

$\forall\, xs : \mathsf{optional}\ [X] \bullet \mathsf{defined}\ \ xs \Leftrightarrow \#\, xs \ = \ 1 \ \wedge$
$\qquad \mathsf{undefined}\ \ xs \Leftrightarrow \#\, xs \ = \ 0 \ \wedge$
$\qquad (\mathsf{defined}\ xs) \Rightarrow \mathsf{the}\ \ xs = (\mu\, x : X \mid x \in xs)$

The user read state comprises the history and the position.

ReadUserState

History
Position

Initially, there is no statistical information.

InitReadUserState

ReadUserState

$AccHistory = \{\}$
$CurrHistory = \{\}$

3.3 The Soft-Link Hypertext Model

The combination of system read state and user read state defines the read state of SLHM. We refer to this as a session. The read state constitutes the intelligence in the model; it is applied as an auxiliary navigation tool for the 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, the current indicators are concept activations, the current history, the current (if known) and next (if known) position.

The combined history of the past and present users is the set of all concepts which have been visited. Furthermore, although any concept which 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.

SoftLinkHypertext

ReadSystemState
ReadUserState

$(\bigcup AccHistory) \cup CurrHistory = \mathrm{dom}\ ConceptsVisited$
$CurrHistory \subseteq \{c : Concepts \mid ConceptActivation\ c = 1 \bullet c\}$
$CurrentConcept \subseteq Concepts$
$NextConcept \subseteq Concepts$

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.

InitSoftLinkHypertext

SoftLinkHypertext
InitReadSystemState
InitReadUserState

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. Any such read application does not affect the structure of the conceptual index.

```
── ΔSoftLinkHypertext ──────────────────────
│ SoftLinkHypertext
│ SoftLinkHypertext'
├──────────
│ Ξ 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.

```
── Start ───────────────────────────────────
│ ΔSoftLinkHypertext
│ Ξ ReadLinks
│ Ξ ReadConceptsVisited
│ Ξ AccumulateHistory
├──────────
│ ConceptActivation' = {c : Concepts • (c, 0)}
│ CurrHistory' = {}
│ 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, it is only the position which changes. We call any such move starting a new trail.

```
── StartNewTrail ───────────────────────────
│ ΔSoftLinkHypertext
│ Ξ ReadSystemState
│ Ξ History
│ NewStartingConcept? : CONCEPT
├──────────
│ NewStartingConcept? ∈ Concepts
│ the CurrentConcept' = NewStartingConcept?
│ undefined NextConcept'
└──────────────────────────────────────────
```

From this stage 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 want to 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.

$$Report ::= No\_Such\_Leaving\_Link \mid Must\_Start\_Trail\_Before\_Browsing$$

$UserBrowseMoveError1$

$link? : LINK$
$\Xi SoftLinkHypertext$
$report! : Report$

$\mathsf{undefined}\ CurrentConcept$
$report! = Must\_Start\_Trail\_Before\_Browsing$

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.

$UserBrowseMoveError2$

$link? : LINK$
$\Xi SoftLinkHypertext$
$report! : Report$
$\mathsf{ReadabilityFunctions}$

$\mathsf{defined}\ CurrentConcept$
$link? \notin LeavingLinks\ (\mathsf{the}\ CurrentConcept)$
$report! = No\_Such\_Leaving\_Link$

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.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{UserBrowseMove}} \\
\Delta \textit{SoftLinkHypertext} \\
\textit{link}? : \textit{LINK} \\
\mathsf{ReadabilityFunctions} \\
\hline
\mathsf{defined}\ \textit{CurrentConcept} \\
\textit{link}? \in \textit{LeavingLinks}\ (\mathsf{the}\ \ \textit{CurrentConcept}) \\
\mathsf{the}\ \ \textit{NextConcept}' = \textit{Child link}? \\
\textit{CurrentConcept}' = \textit{CurrentConcept} \\
\hline
\end{array}$$

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 automatically updated so maintaining the state invariant definition of weight.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{BackwardLearning}} \\
\Delta \textit{SoftLinkHypertext} \\
\Xi \textit{ReadUserState} \\
\mathsf{ReadabilityFunctions} \\
\hline
\textit{ConceptsVisited}' = \textit{ConceptsVisited} \uplus [\![\mathsf{the}\ \ \textit{CurrentConcept}]\!] \\
\textit{LinksVisited}' = \textit{LinksVisited} \uplus \\
\qquad\qquad\qquad [\![(\mathsf{the}\ \ \textit{CurrentConcept}, \mathsf{the}\ \ \textit{NextConcept})]\!] \\
\textit{ConceptActivation}' = \textit{ConceptActivation} \oplus \\
\qquad\qquad\qquad \{(\mathsf{the}\ \ \textit{CurrentConcept}, 1)\} \\
\hline
\end{array}$$

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.

```
ForwardActivationSpreading
  $\Delta SoftLinkHypertext$
  $\Xi ReadLinks$
  $\Xi ReadConceptsVisited$
  $\Xi ReadUserState$
  ReadabilityFunctions
  $InputActivation : LINK \rightarrow\!\!\!\!\!+ RAT_0^1$
  ----------------
  $ConceptActivation' = ConceptActivation \oplus$
    $\{c : Concepts \mid c \in Children\ (\mathsf{the}\ \ CurrentConcept) \bullet$
      $(c, \mathsf{min}\ \ (1, (ConceptActivation\ \ c)+$
        $\mathsf{sumseq}\ \ (\ \mathsf{mapsettoseq}\ \ InputActivation\ \ (ArrivingLinks\ \ c))))\}$
  $\forall c_1, c_2 : CONCEPT \mid (c_1, c_2) \in Links \bullet$
      $InputActivation\ \ (c_1, c_2) =$
          $LinkWeight\ \ (c_1, c_2) * ConceptActivation\ \ c_1$
```

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 algorithm spreading more readable.

At the end of each browsing move, the system outputs a sequence of concepts, listed in the order of their activations. This is used as the online navigation guide for the current user to search and retrieve information more efficiently.

```
OutputConceptOrder
  $\Xi SoftLinkHypertext$
  $reply! : \mathsf{seq}\ \ CONCEPT$
  ----------------
  $reply! = \mathsf{sort}\ \ Concepts\ \ ConceptActivation$
```

The function sort simply takes a set of elements and sorts them in decreasing order according to an ordering function.

```
[[]
  $X]\ \mathsf{sort} : (\mathbb{P}\, X) \rightarrow (X \rightarrow\!\!\!\!\!+ RAT_0^1) \rightarrow\!\!\!\!\!+ \mathsf{seq}\ \ X$
  ----------------
  $\forall f : (X \rightarrow\!\!\!\!\!+ RAT_0^1);\ x_1, x_2 : X;\ xs : \mathbb{P}\, X \mid xs \subseteq \mathsf{dom}\ f \bullet$
              $\mathsf{ran}\ (\mathsf{sort}\ \ xs\ f) = xs\ \wedge$
              $\#(\mathsf{sort}\ \ xs\ f) = \#xs\ \wedge$
              $\langle x_1, x_2 \rangle\ \mathsf{in}\ \mathsf{sort}\ \ xs\ f \Rightarrow f\ \ x_1 \geq f\ \ x_2$
```

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

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_ \ \mathit{UpdateUserHistory}\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\Delta \mathit{SoftLinkHypertext} \\
\Xi \mathit{ReadSystemState} \\
\Xi \mathit{AccumulateHistory} \\
\hline
\mathit{CurrHistory}' = \mathit{CurrHistory} \cup \mathit{CurrentConcept} \\
\mathit{CurrentConcept}' = \mathit{NextConcept} \\
\text{undefined } \mathit{NextConcept}' \\
\hline
\end{array}$$

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

$$\begin{array}{l}
\mathit{Browse} == (\mathit{UserBrowseMove} \land \\
\qquad\qquad (\mathit{BackwardLearning} \fatsemi \mathit{ForwardActivationSpreading} \fatsemi \\
\qquad\qquad\qquad \mathit{OutputConceptOrder} \fatsemi \mathit{UpdateUserHistory})) \\
\qquad\qquad \lor \\
\qquad \mathit{UserBrowseMoveError}1 \\
\qquad\qquad \lor \\
\qquad \mathit{UserBrowseMoveError}2
\end{array}$$

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.

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_ \ \mathit{Quit}\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\Delta \mathit{SoftLinkHypertext} \\
\Xi \mathit{ReadSystemState} \\
\Xi \mathit{CurrentHistory} \\
\hline
\mathit{AccHistory}' = \mathit{AccHistory} \cup \{\mathit{CurrHistory}\} \\
\hline
\end{array}$$

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.

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_ \ \mathit{UpdateAddConcept}\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\Delta \mathit{ReadSystemState} \\
\Xi \mathit{ReadLinks} \\
\Xi \mathit{ReadConceptsVisited} \\
\mathit{AddConcept} \\
\hline
\mathit{ConceptActivation}' = \mathit{ConceptActivation} \cup \{(c?, 0)\} \\
\hline
\end{array}$$

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

$$\begin{array}{|l}
\textit{UpdateAddLink} \\
\hline
\Delta \textit{ReadSystemState} \\
\Xi \textit{ReadConcepts} \\
\Xi \textit{ReadLinksVisited} \\
\textit{AddLink} \\
\textsf{ReadabilityFunctions} \\
\hline
\end{array}$$

When a link is removed the frequency of that link is subtracted from the frequency of its parent concept.

$$\begin{array}{|l}
\textit{UpdateRemoveLink} \\
\hline
\Delta \textit{ReadSystemState} \\
\Xi \textit{ReadConceptActivation} \\
\textit{RemoveLink} \\
\textsf{ReadabilityFunctions} \\
\textsf{FrequencyFunctions} \\
\hline
\textit{LinksVisited}' = \{(c_1?, c_2?)\} \lhd \textit{LinksVisited} \\
\textit{ConceptsVisited}' = \textit{ConceptsVisited} \\
\qquad\qquad\qquad\qquad \{(c_1?, \textit{LinkFrequency}\ (c_1?, c_2?))\} \\
\hline
\end{array}$$

4. When we remove a concept, we must delete any occurrence of it from the accumulative history in the system read state.

$$\begin{array}{|l}
\textit{UpdateRemoveConcept} \\
\hline
\Delta \textit{ReadSystemState} \\
\Xi \textit{ReadLinks} \\
\textit{RemoveConcept} \\
\Delta \textit{AccumulateHistory} \\
\hline
\textit{ConceptActivation}' = \{c?\} \lhd \textit{ConceptActivation} \\
\textit{ConceptsVisited}' = \{c?\} \lhd \textit{ConceptsVisited} \\
\textit{AccHistory}' = \{cs : \textit{AccHistory} \bullet cs \setminus \{c?\}\} \\
\hline
\end{array}$$

6 Conclusions

This paper has presented a Z specification of a new information retrieval model called the soft-link hypertext model (SLHM). 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 [12]. We claim the benefits of doing this are twofold:

- We have gained confidence in the validity and utility of the framework.
- 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

Particular thanks to Colin Myers and also to Claire Cohen, Jennifer Goodwin, Mike Hinchey, Paul Howells, Michael Luck and Mark Priestley for comments on earlier versions of this paper. Thanks too to the anonymous referees who made valuable suggestions and isolated several errors in the original specification. The specification in this paper was checked using the *fuzz* package [24].

References

1. M. Arbib. *The Metaphorical Brain 2: Neural Networks and Beyond.* Wiley, 1989.
2. R. Belew. Adaptive Information Retrieval. In *Proceedings of the 12th Annual International ACM SIGIR Conference on Research Development in Information Retrieval*, 1989.
3. A. Carling. *Introduction to Neural Networks.* Sigma Press, 1992.
4. P. Cohen and R. Kjeldsen. Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing & Management*, 23(4):255 – 268, 1987.
5. J. Conklin. Hypertext: An Introduction and Survey. *Computer*, 1987.
6. W. B. Croft and H. Turtle. A Retrieval Model for Incorporating Hypertext Links. In *Hypertext'89*, November 1989.
7. S. J. DeRose. Expanding the Notion of Links. In *Hypertext'89*, November 1989.
8. M. d'Inverno and J. Crowcroft. Design, specification and implementation of an interactive conferencing system. In *Proceedings of IEEE Infocom, Miami, USA. Published IEEE*, 1991.
9. M. d'Inverno, G. R. Justo, and P. Howells. A formal framework for specifying design methodologies. In *29th Annual Hawaii International Conference on System Sciences*, pages 741–750. IEEE Computer Society Press, 1996.

10. M. d'Inverno and M. Luck. A formal view of social dependence networks. In *Distributed Artificial Intelligence: Architecture and Modelling, First Australian Workshop on DAI, Lecture Notes in Artificial Intelligence, 1087*, pages 115–129. Springer Verlag, 1996.
11. M. d'Inverno and M. Luck. Formalising the contract net as a goal directed system. In W. Van de Velde and J. W. Perram, editors, *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, LNAI 1038*, pages 72–85. Springer-Verlag, 1996.
12. M. d'Inverno and M. Priestley. Structuring a Z specification to provide a unifying framework for hypertext systems. In J. P. Bowen and M. G. Hinchey, editors, *ZUM'95: 9th International Conference of Z Users, LNCS 967*, pages 83–102, Heidelberg, 1995. Springer-Verlag.
13. S. Grossberg. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.* D.Reidel Publishing Company, 1992.
14. F. G. Halasz. Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the ACM*, 31(7), July 1988.
15. F. G. Halasz and M. Schwartz. The Dexter Hypertext. *Communications of the ACM*, 37(2):30–39, 1994.
16. G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1 – 3):185 – 234, 1989.
17. M. J. Hu. *An Intelligent Information System.* PhD thesis, Department of Computer Science, Gower Street, University College London, 1994.
18. M. J. Hu and P. Kirstein. An Intelligent Hypertext System. In *The First International Workshop on Intelligent Hypertext (CIKM-93)*, Washington, 1993.
19. A. Lelu. Browsing through image databases via data analysis and neural networks. In *User-oriented Content-based Text and Image Handling: Proceedings of RIAO'88,*, pages 1034 – 1043, 1993.
20. M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 254–260. AAAI Press / MIT Press, 1995.
21. R. M. Fung et. al. An Architecture for Probabilistic Concept-based Information Retrieval. In *Proceedings of the ACM SIGIR'90 conference, Brussels, Belgium*, pages 455 – 467, 1990.
22. D. E. Rose and R. K. Belew. A connectionist and symbolic hybrid for improving legal research. *International Journal of Man-Machine Studies*, 35(1):1–31, 1991.
23. J. Savoy. Bayesian Inference Networks and Spreading Activation in Hypertext Systems. *Information Processing and Management*, 28(3):389 – 406, 1992.
24. J. M. Spivey. *The ƒUZZ Manual.* Computing Science Consultancy, 2 Willow Close, Garsington, Oxford OX9 9AN, UK, 2nd edition, 1992.
25. J. M. Spivey. *The Z Notation.* Prentice Hall, 2nd edition, 1992.
26. J. Reggia T. E. Doszkocs and X. Lin. Connectionist models and information retrieval. *Annual Review of Information Science and Technology*, 25, 1990.
27. S. H. Valentine. Putting numbers into the mathematical toolkit. In J. P. Bowen and J. E. Nicholls, editors, *Z User Workshop, London 1992*, Workshops in Computing, pages 9–36. Springer-Verlag, 1993.