Content Oriented Relations between Text Units – a Structural Model for Hypertexts[1]

Rainer Hammwöhner and Ulrich Thiel

University of Constance,

Dept. of Information Science

Project TWRM-TOPOGRAPHIC

Postfach 5560, D-7750 Konstanz, F.R.G.

I. Abstract

A common feature of various recently developed information systems is the decomposition of linear document structures which are enforced by conventional print media. Instead, a network organisation of information units of different forms (textual, graphical, pictorial and even auditory presentation modes may be combined) is provided. Documents organized this way are called 'hypertexts'. However, two questions arise immediately when an effort is made to build information systems on the basis of this conception:

- What are the 'units' constituting a hypertext?
- What sort of links between the units will be provided?

Most approaches to hypertext systems impose the task of deciding these questions on the authors of hypertexts, thus the systems are hypertext management devices (eg CHRISTODOULAKIS ET AL. 86, WOELK ET AL 86). The approach taken in this paper leaves a more active role to the software by applying knowledge based techniques. The starting point is the automatic content analysis of machine-readable full-text documents which may be downloaded from a full-text data base. The analysis process results in a partitioning of the document into thematically coherent text passages, which are one kind of node of the hypertextual version of this document. Other nodes contain graphics, tables and summarizations. The content analysis is accomplished by a semantic parser, which has access to an explicit model of the discourse domain. The TOPIC-system (HAHN/REIMER 86) comprises prototypical implementations of these components. Due to the semantic modelling relations between the nodes may be formally defined in order to provide content oriented browsing facilities. The graphical retrieval system TOPOGRAPHIC (THIEL/HAMMWÖHNER 87) employs an already implemented subset of them to guide users to relevant text parts.

In this paper we outline a structure model for hypertexts based on partial representations of the meaning of text parts. Formal definitions of content oriented relations between such text units are given in terms of a logic specification language.

---

[1] This paper has been presented at Hypertext '87, published in Hypertext '87 papers, Chapel Hill, NC, University of North Carolina, 1987, pp. 155-174.

II. Text Units: Result of a Content Oriented Fragmentation of Documents

In this section, we will first present a basic model of general hypertext systems that allow users to browse in a set of "text units" (they may be presented in a multi-media environment). In order to access relevant units, the user must be guided along content oriented links connecting units that have a semantic overlap.

A concept oriented framework for modelling the semantics of text units is outlined in the second part of this section, serving as a foundation for the definition of relations between the concepts modelled. These relationships may be used in two ways: First, they provide information for the semantic parsing that is needed to obtain representations of the units' contents. (We briefly outline some problems of the parsing process that can be solved by the defined relations.) In the next chapter, the construction of relations between text units will then be based on the same relations.

Content analysis of text units provides not only a powerful browsing facility, but also the opportunity to propose an augmented hypertext model This model features "derived text units", which may be added to the original ones. These new text units may be regarded as summarizations of the contents of text fragments. They provide an overview over larger units, and surplus give rise to a special type of browsing. This navigation operator leading from the condensed abstract to detailed original information is called "informational zooming". Parts of the abstract may be weighted, thus the zooming is interest-driven.

II.1 Basic Hypertext Features: Browsing in a Set of Text Units

Most hypertext systems employ graphical user interfaces, which are part of object oriented programming environments. Windowing being a usual feature, the common technique of assigning a window (or icon as a shrunken form) to each text unit allows to choose appropriate presentation methods for each type of text unit

The surface structure of the given original document and the media used to communicate can be employed to introduce the following classification of text units:

(1) The units conveying the simplest semantic structures are formatted graphical entities like tables. Their surface structure corresponds directly to their contents.

(2) Textual units consist of sentences, therefore their surface structure is linear. The semantics of a text unit can only be partially modelled due to its complexity.

(3) Graphical units are assemblies of graphic primitives, whereas pictures and icons are sets of pixels. The analysis of pictures requires a dedicated methodology, which will not be a topic of this paper, but the (partial) modelling of pictorial semantics can be accomplished within the formal framework outlined in the following.

As speech fragments, animated graphics, software modules, and videos may also be nodes in hypertext networks, the classification above does not claim to be a complete one. However, the types of text units included in the subset can be regarded as the most important ones for our approach to construct hypertext versions from machine-readable documents.

In order to access text units that are not depicted on the screen the user is usually given the opportunity to browse in the hypertext network. There are two kinds of browsing (cf BATES 86):

a) undirected browsing: The user investigates text units in an arbitrary ordering. This type of navigation may provide a survey of what can be accessed in general, but if the number of items available increases the user will ask for a more dedicated access to text units which are relevant for him.

b) directed browsing: This way of navigation requires that the text units are interconnected by meaningful links. Thus, a selection of the appropriate link may be accomplished, which results either in the replacement of the text unit currently displayed on the screen, or in the presentation of one or more additional text units.

In this paper, we are concerned with the latter. In most hypertext systems, the selection of text units to be accessed is based on keywords or descriptors (eg WEYER 82, WEYER/BORNING 85) which are either assigned to the text units by the hypertext author or are detected in the units by string matching procedures. However, if the network of text units is to be constructed automatically, formally specified relations which refer to partial representations of the contents of the interconnected units are needed to provide content oriented browsing facilities. The semantic modelling may be restricted to topical descriptions, because in most situations it suffices to know what a text unit is about instead of having a detailed account of its contents. Therefore, the text analysis may be accomplished by selective parsing procedures that capture the meaning of nominal phrases. As far as full texts are concerned, a method which allows the determination of the "aboutness" of a text in a reasonable time is required that is applicable in situations where larger collections of texts have to be processed. (Similar restrictions of image analysis may aim at identification of the objects depicted in a picture, while neglecting deeper analysis like action detection and scene interpretation.)

II.2 Capturing the Contents of Text Units: Knowledge Representation and Semantic Parsing

In the following, we specify the properties of a knowledge representation formalism which is powerful enough to capture the contents of tables and to support a semantic parsing yielding topical descriptions of text passages on an indicative level. There is evidence for the appropriateness of similar approaches to image and speech analysis, but this hypothesis will not be discussed to further extent in this paper.

We start with the category of text units whose semantics is intuitively understood in terms of simple relationships: tables. Added to textual parts of a document, they often serve to summarize the main facts or to communicate sets of formatted data records. The semantics is constituted by the aggregation of columns or rows as a table, thus providing a framework for entering the individual items into the right place. Each column (row) is associated with its set of entries, as well as the table is constituted by its columns or rows. This obviously may be modelled by a "frame" (cf MINSKY 75) by identifying the columns (rows) with "slots", the data items contained in a column being its "slot entries".

Here is a concise verbalization of the frame construct from which first-order predicates are obtained They may be regarded as abbreviations of the informally specified structural conditions for frames, slots, and entries. (Although it is possible to give a complete axiomatic system (cf HAYES 79), for the purposes of this paper it is sufficient to adopt a descriptive view on frames treating them as an abstract data type (cf HAYES/HENDRIX 81). Thus, issues of implementation may be left outside, while the discussion concentrates on properties of (and relations between) objects that are presumed to match the frame specification.)

A frame consists of a name and a set of slots. This requirement may be formalized using the basic predicate

- **is-frame(f)** asserting that f is a frame, the function
- **fn(f)** yielding the name of the frame f, and for each slot s the condition

- **is-slot(f,s)** stating the assignment of s to the frame f. A slot has a name given by the function
- **sn(s)** and a (potentially empty) set of entries such that for each entry the proposition
- **is-entry(f,s,e)** holds.

Slot entries may either be unstructured individuals or may be frames, having a slot set of their own. The latter possibility allows a modelling of aspects (slots) of a frame by nesting the representation structures. In the following example, a frame called "Zenon-X" represents a hypothetical micro-computer which is characterized by the features "Manufacturer", "Vendor", and "CPU". A fictive corporation is assigned to the "Manufacturer"-slot: "Zeta-Machines". Thus, the meaning of the table below is captured completely.

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines		

The same frame may also be taken as a partial semantic representation of a sentence fragment like: ... the Zenon-X which was recently developed by Zeta-Machines...

However, for the purpose of text analysis procedures which yield such a frame representation of a given sentence, it is necessary to restrict the possible slot filling operations by means of integrity rules in order to model linguistic regularities (cf REIMER 86). A first step in this direction is the introduction of "singleton slots" which stand for properties that can only have one value at a time. The singleton slots of the frame fulfil the condition

is-single(f,s).

Further integrity rules may require that an item must be a member of a specified set of allowed entries, if it is to be assigned to the slot as a value during the parsing process. (We give no further formalisation of the notion of integrity rules here, because they are primarily important for processes that change the knowledge base, i.e. editing domain-specific knowledge and parsing the documents, and have been discussed in REIMER 86, and REIMER/HAHN 85 where a frame representation model (FRM) is presented which captures the semantics of concepts. As FRM is used as the knowledge representation formalism in the text analysis system TOPIC, the prototypical hypertext system TOPOGRAPHIC accessing the results of TOPIC may therefore assume the knowledge structures to be consistent.)

In order to capture the conceptual contents of a given text correctly the text analysis mechanism of TOPIC has to perform two main tasks: anaphora resolution and (restricted) concept learning. The solution to these problems is primarily based on providing two different kinds of frames: "A prototype frame acts as a representative of a concept class consisting of instance frames which all have the same slots but differ front the prototype in that they are further characterized by slot entries. Thus, instance frames stand for individual concepts of a domain of discourse" (HAHN/REIMER 86). The notion of prototypes can be formalized in the following definition:

Def: 1:

$$\forall f : \text{is-protype}(f) \leftrightarrow \text{is-frame} \wedge \neg\exists s : [\text{is-slot}(f,s) \wedge \exists e : \text{is-entry}(f,s,e)]$$

Example: inst

Microcomputer	Manufacturer	Vendor	CPU

$\xrightarrow{inst}$

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines		

The formal definition below is a generalized Version of the inst-predicate, holding not only between a prototype and its corresponding instance frames, but also between two instances if the second frame is a specialization of the first one:

Def 2:

$$\forall f_1,f_2 : \text{inst}(f_1,f_2) \leftrightarrow$$
$$\qquad \text{is-frame}(f_1) \wedge \text{is-frame}(f_2) \wedge$$
$$\qquad \forall s_1 \exists s_2 : \left[\text{is-slot}(f_1,s_1) \rightarrow \text{is-slot}(f_2,s_2) \wedge \text{sn}(s_1) = \text{sn}(s_2)\right] \wedge$$
$$\qquad \forall s_2 \exists s_1 : \left[\text{is-slot}(f_2,s_2) \rightarrow \text{is-slot}(f_1,s_1) \wedge \text{sn}(s_1) = \text{sn}(s_2)\right] \wedge$$
$$\qquad \exists s_2, e : \left[\text{is-entry}(f_2,s_2,e) \wedge \neg \exists s_1 : \left[\text{sn}(s_1) = \text{sn}(s_2) \wedge \text{is-entry}(f_1,s_1,e)\right]\right]$$

This can be employed in a simple but often sufficient heuristic of concept learning: If an unknown noun occurs during the parsing process and there is an indicator of what concept class it may belong to (eg if it is a compound noun containing a prototype identifier), then it can be integrated into the knowledge base as a frame inheriting the slots of its supposed prototype. The slots may then be filled with further information from the text.

In the process of anaphora resolution the inst-relation is used for identifying the instance frame that occurred in the previous text part, if a prototype frame is encountered (and there is linguistic evidence that it is used anaphorically). This method can be extended to other prototypes which are generalizations of the instance's prototype. In this case, the is-a-relation holds between the prototypes. (Note that the above descriptions of concept learning and anaphora resolution are idealized to emphasize the very ideas. More technical specifications give HAHN/REIMER 86). The formal definition of the is-a-relation is recursive due to the fact that slots may be frames themselves. To cover this case we use the extended is-a-relation e-is-a, which is the transitive closure of the union of the inst- and is-a-relations.

Def 3:

$$\forall f_1,f_2 : \text{is-a}(f_1,f_2) \leftrightarrow$$
$$\qquad \text{is-prototype}(f_1) \wedge \text{is-prototype}(f_2) \wedge$$
$$\qquad \exists s : \left[\text{is-slot}(f_2,s) \wedge \neg\text{is-slot}(f_1,s)\right] \wedge$$
$$\qquad \forall s : \left[\text{is-slot}(f_2,s_2) \rightarrow \left[\begin{array}{l}\text{is-slot}(f_1,s_1) \vee \\ \exists s' f, f' : \left[\begin{array}{l}\text{is-slot}(f_2,s') \wedge \text{fn}(f) = \text{sn}(s) \wedge \\ \text{fn}(f') = \text{sn}(s') \wedge \text{e-is-a}(f,f')\end{array}\right]\end{array}\right]\right]$$

Def 4:

$$\forall f_1,f_2 : \text{e-is-a}(f_1,f_2) \leftrightarrow$$
$$\qquad \text{is-a}(f_1,f_2) \vee \text{inst}(f_1,f_2) \vee \exists f' : \left[\text{is-a}(f_1,f') \wedge \text{inst}(f',f_2)\right]$$

In the following example, the is-a-relation holds between "Micro-Computer" and "Graphics-Workstation", because of the additional slot "Graphics-Screen" .

Example: inst

Microcomp.	Manuf.	Vendor	CPU

$\xrightarrow{\text{is-a}}$

Graphics Workst.	Manuf.	Vendor	CPU	Graphics Screen

The knowledge representation mechanism of TOPIC/TOPOGRAPHIC combines the modelling of concepts as frames with the modelling of certain relationships between frames, a technique originally devised in the area of 'semantic networks'. Furthermore, the relations are defined mathematically exploiting the structural properties of the frames involved. Due to the concise definitions, the concept hierarchy of the knowledge base is system-controlled, i.e. each new frame entered into the knowledge base will be classified automatically by computing all the relational links that connect it to modelled concepts (cf REIMER 86). The system incorporates a variety of other relations (eg parts), which support the semantic parsing procedure. Thus, the parser - organized as a lexically distributed grammar in the format of word experts (cf HAHN 86) - is not only enabled to assemble factual knowledge by recognizing concepts, filling slots or classifying sub-concepts but also to detect topical shifts by combining syntagmatic indicators (start of a new paragraph, occurrence of idiomatic phrases that indicate a new topical focus) with semantic criteria (eg if the current sentence has no semantic overlap with the previous ones.)

II.3 An Augmented Hypertext Model

Whereas the world knowledge base of TOPIC/TOPOGRAPHIC contains a taxonomic model of the discourse domain, the text knowledge base consists of "text graphs" which represent the knowledge obtained by the parsing process. Each analysed text is thus stored not only in textual form (i.e. the original text units), but also associated with its topical and, to some extent, factual content, which is organized as a conceptual graph. The following information about the analysed text can be found in the text knowledge resulting from the analysis and the subsequent condensation process:

a) A multi-hierarchical graph whose nodes contain the topical structures of the text in decreasing generality. The contents of these nodes are similar to world knowledge structures.

b) Fragments of world knowledge denoting the main topics of the text passages, i.e. the frames that match the most salient concepts in thematically coherent text parts. The frames are connected by relational links, thus a network representing the topical structure of the text unit is given.

c) The frames occurring in the networks may have 'filled' slots, i.e. there may be entries assigned to them during the process of text analysis. The filling of slots contributes to the factual information from the text by adding more precise details to the general information provided by the frames and their slots.

The semantic representations of text units offer the opportunity to support the user with interaction techniques which complement the access to original text units via browse operations. The basis for this augmentation of the dialogue facilities are artificial (or derived) text units:

- The natural language presentation of text unit contents does not necessarily depend on the original text fragments. Text generation procedures that cast the knowledge structures into predefined templates are currently under development. Thus, it will be possible to provide abstracts of the text units.
- A graphical presentation of the semantic structures - as tables or networks of nodes representing concepts - provides an automatic "text mapping" (a technique of drawing conceptual graphs in order to memorize the contents of texts (cf DANSERAU/HOLLEY 82) which enhances remembering performance). This facility is featured by our prototypical information system TOPOGRAPHIC

The presentation of artificial text units as "condensates" of the original ones entails the possibility of switching between different layers of specifity which may be assigned to the given text units (THIEL/HAMMWÖHNER 87 provide a more detailed discussion of the layered organization of text units). Especially the access to more detailed text units is supported in TOPOGRAPHIC by a general operator: informational zooming. As in optics zooming reveals more details of physical objects, the 'zoom' option in TOPOGRAPHIC can be used to access more detailed informational structures, or, in other words, to switch to a layer below. This is facilitated by navigating along the semantic relation that holds between an abstract (or conceptual network) and the text unit it has been derived from during the text analysis. (The next chapter provides an overview of the semantic relations that may additionally be used for navigation in a hypertext graph.) Thus, it is easy for the user to access an original text unit whose corresponding abstract or conceptual graph are relevant. The expansion of simple objects, usually the nodes (i.e. frames) of a network given, also fits into this model. Zooming alone, however, does not suffice for a goal oriented dialogue, because there may be too much detail information on the layer below. Therefore, a sort of focussing is needed. This is accomplished by the 'select' option which allows to mark those features of a given layer, which are to be shown in detail by the zoom operator. Selecting a topical profile from a knowledge base representing the taxonomic structure of the discourse domain to which the analysed documents pertain induces a relevance weighting on the text units. This will be employed to define pragmatic relations between text units in the next chapter.

III. Hypertextual Relations

As we have outlined in the previous chapter intertextual relations are of crucial importance for hypertext systems in order to supply the user with operators for content oriented navigation within the hypertext graph. According to the semiotic categories there are the following types of hypertextual relations:

- Syntagmatic relations are derived from the surface structure of the documents within the hypertext. The relation "next-passage-within-the-same-text" eg hold between two text units which can be found in the same document at adjacent positions. This means that our notion of syntagmatic relations excludes surface structures of hypertext presentation structures at runtime, which are generated according to semantic and pragmatic relations. Thus the computing of syntagmatic relations is straightforward.
- Semantic relations which represent paradigmatic aspects of text units depend on the content of the text units. Informational inconsistency between text units for instance induces a semantic relation. Special types of semantic relations connect original and derived text units, which result from parsing, text condensation or generation.
- Pragmatic relations represent dependencies between the dialogue context, the intentions of the user which may be given as an interest profile, the content of the hypertext and the intentions of the authors of the text units. Examples for pragmatic relations are "next-lesson" which holds between tutorial text units or "next-relevant-text" which gives an answer to a query.

III.1 Semantic Relations

Semantic relations are based on structural similarities of the semantic representations of the text units. Some of the large number of possible semantic relations will be defined in the following. For each relation a informal verbal introduction to its meaning - to show its relevance to the task of guiding the user through a hypertext graph -, a formal definition and an example will be given. A formal definition of a relation is based on properties (slot-entries) of frames which are elements of the semantic representations of text units. Thus hypertextual relations can be inferred by the means of relations between frames.

same name:

The hypertext consists of several text units belonging to different texts. Each unit has its own set of frames as semantic representation the name of each frame denoting a concept the text is about (synonyms being normalized) and the slots cumulating the facts concerning this concept as entries. A frame may be member of more than one frame set and there may be several frames (as members of different frame sets) with the same name, but with different slot fillers. Most of the semantic relations as defined below describe relations between different descriptions of the same topic, therefore a means of testing whether two frames have the same name is a prerequisite of the definition of such relations.

Def 5:

$$\forall f_1,f_2 : \text{eqn}\left(f_1,f_2\right) \leftrightarrow \text{is-frame}\left(f_1\right) \wedge \text{is-frame}\left(f_2\right) \wedge \text{fn}\left(f_1\right) = \text{fn}\left(f_2\right)$$

complement

All information contained in the first frame must also be found in the second one completed by (at least) an additional entry. The information of the first frame is confirmed and completed by the second one.

Def. 6:

$$\forall f_1,f_2 : \text{compl}\left(f_1,f_2\right) \leftrightarrow \text{eqn}\left(f_1,f_2\right) \wedge \exists f : \left[\text{inst}\left(f,f_1\right) \wedge \text{inst}\left(f,f_2\right)\right] \wedge \text{inst}\left(f_1,f_2\right)$$

The relation is defined on the set of frames, therefore both parameters must be frames. Additionally they should have the same name (eqn def. 5) to indicate that they refer to the same topic. These presumably different frames taken from two distinct text units must have the same slot structure to be comparable, therefore the inst relation (def. 3) must hold between each of them and a third frame. f_2 must contain an additional entry in any slot and thus be an instance of f_1.

Example: compl

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines		

$\xrightarrow{\text{compl}}$

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines		68020

x-complement

The information given by these frames is disjunct. Both frames are needed to obtain the complete information.

Def.7:

$$\forall f_1,f_2 : \text{x-compl}\left(f_1,f_2\right) \leftrightarrow \text{eqn}\left(f_1,f_2\right) \wedge \exists f : \left[\text{inst}\left(f,f_1\right) \wedge \text{inst}\left(f,f_2\right)\right] \wedge$$
$$\forall s,s' : \left[\begin{array}{l} \text{is-slot}\left(f_1,s\right) \wedge \text{is-slot}\left(f_2,s'\right) \wedge \\ \forall e : \left[\text{is-entry}\left(f_1,s,e\right) \wedge \text{is-entry}\left(f_2,s',e\right) \rightarrow \text{sn}\left(s\right) \neq \text{sn}\left(s'\right)\right] \end{array}\right]$$

The initial conditions of Def. 6 apply as well. Corresponding slots of f_1 and f_2 (the slots have the same names) must contain disjunct sets of entries. Thus, if a concept is entry to both frames, it must be entry to slots with different names.

Example: x-compl

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines		

$\xleftrightarrow{\text{x-compl}}$

Zenon-x	Manufacturer	Vendor	CPU
			68020

add-inf

This relation is similar to the complement relation, but the focus is on a special property of the frames, therefore the relation has a slot as third argument.

Def. 8:

$$\forall f_1,f_2,s:\text{add-inf}(f_1,f_2,s)\leftrightarrow \text{eqn}(f_1,f_2)\wedge\exists f:\left[\text{inst}(f,f_1)\wedge\text{inst}(f,f_2)\right]\wedge$$

$$\exists s':\begin{bmatrix}\text{is-slot}(f_1,s)\wedge\text{is-slot}(f_2,s')\wedge\text{sn}(s)=\text{sn}(s')\wedge\\ \forall e:\left[\text{is-entry}(f_1,s,e)\rightarrow\text{is-entry}(f_2,s',e)\right]\wedge\\ \exists e':\left[\text{is-entry}(f_2,s',e')\rightarrow\neg\text{is-entry}(f_1,s,e')\right]\end{bmatrix}$$

The specified slot *s* must be a member of f_1. f_2 must have a slot with the same name. All entries that are assigned to the slot *s* of f_1 must be assigned to the corresponding slot of f_2 as well. The latter must have at least one additional entry.

Example: add-inf

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines	Harrods	

$\xrightarrow{\text{add-inf}}$

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines	Harrods Tiffany	

alt-inf

The relation alt-inf is derived from add-inf in the same way as x-compl is derived from compl.

Def. 9:

$$\forall f_1,f_2,s:\text{alt-inf}(f_1,f_2,s)\leftrightarrow \text{eqn}(f_1,f_2)\wedge\exists f:\left[\text{inst}(f,f_1)\wedge\text{inst}(f,f_2)\right]\wedge$$

$$\exists s':\begin{bmatrix}\text{is-slot}(f_1,s)\wedge\text{is-slot}(f_2,s')\wedge\text{sn}(s)=\text{sn}(s')\wedge\\ \exists e:\left[\text{is-entry}(f_1,s,e)\right]\wedge\exists e':\left[\text{is-entry}(f_2,s',e')\right]\\ \forall e:\left[\text{is-entry}(f_1,s,e)\rightarrow\text{is-entry}(f_2,s',e)\right]\wedge\\ \forall e':\left[\text{is-entry}(f_2,s',e')\rightarrow\neg\text{is-entry}(f_1,s,e')\right]\end{bmatrix}$$

Each of the corresponding slots must have at least one entry. The set of entries of these slots must be disjunct.

Example: add-inf

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines	Harrods	

$\xleftrightarrow{\text{alt-inf}}$

Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines	Tiffany	

conflict

The frames contain inconsistent information.

Def .10:

$$\begin{aligned}&\forall f_1,f_2,s:\text{confl}(f_1,f_2)\leftrightarrow \text{eqn}(f_1,f_2)\wedge\exists f:[\text{inst}(f,f_1)\wedge\text{inst}(f,f_2)]\wedge\\&\quad\exists s,s':[\text{is-slot}(f_1,s)\wedge\text{is-slot}(f_2,s')\wedge\text{sn}(s)=\text{sn}(s')\wedge\\&\qquad\text{is-single}(f_1,s)\wedge\text{is-single}(f_2,s')\wedge\\&\qquad\exists e_1,e_2:[\text{is-entry}(f_1,s,e)\wedge\text{is-entry}(f_2,s',e_2)\wedge\\&\qquad\quad\neg\text{is-frame}(e_1)\wedge\neg\text{is-frame}(e_2)\wedge e_1\neq e_2\vee\\&\qquad\quad\text{is-frame}(e_1)\wedge\text{is-frame}(e_2)\wedge\\&\qquad\quad[\text{fn}(e_1)=\text{fn}(e_2)\wedge\text{confl}(e_1,e_2)\vee\\&\qquad\quad\ \text{fn}(e_1)\neq\text{fn}(e_2)\wedge\neg\text{e-is-a}(e_1,e_2)\wedge\neg\text{e-is-a}(e_2,e_1)]]]\end{aligned}$$

A conflict may be detected if two frames have corresponding slots (same names) which allow only one entry each. If these slots contain differing factual data, the conflict is detected for sure. If the entries are frames themselves, then there is no conflict, if these frames represent the same concept and there is no conflict between these representations. If an inheritance relation holds between these frames, then there is no conflict as well.

Example: confl

Zenon-x	Manufacturer	Vendor	CPU		Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines	Harrods	**8080**	←confl→		Zeta-Machines	Tiffany	**68020**

property coincidence

Similar properties are assigned to two distinct objects. The passages may be read in order to compare these objects with respect to other properties.

Def. 11:

$$\begin{aligned}&\forall f_1,f_2,s:\text{same-prop}(f_1,f_2,s)\leftrightarrow\text{is-frame}(f_1)\wedge\text{is-frame}(f_2)\wedge\neg\text{eqn}(f_1,f_2)\wedge\\&\qquad\exists f:[\text{inst}(f,f_1)\wedge\text{inst}(f,f_2)]\wedge\\&\qquad\exists s':\begin{bmatrix}\text{is-slot}(f_1,s)\wedge\text{is-slot}(f_2,s')\wedge\text{sn}(s)=\text{sn}(s')\wedge\\\exists e:[\text{is-entry}(f_1,s,e)\wedge\text{is-entry}(f_2,s',e)]\end{bmatrix}\end{aligned}$$

Example: same-prop

Zenon-z	Manufacturer	Vendor	CPU		Zenon-x	Manufacturer	Vendor	CPU
	Zeta-Machines	Harrods	8080	←same−prop→		**Zeta-Machines**	Tiffany	68020

These relations which are defined on sets of frames can be used to define relations on semantic representations of text units which are sets of frames. Thus, the semantic relations are completely independent from the surface structure of texts. Two interrelated text units may therefore be part of the same text or belong to different ones. The predicate unit-rep tests whether a set of frames represents a text unit. A complete definition of unit-rep would require a deeper understanding of the parsing process which is out of the scope of this paper. The following lemma which can be derived from the complete definition suffices our purpose.

Lemma

$$\forall K:\text{unit-rep}(K)\rightarrow\forall f\in K:[\text{is-frame}(f)\wedge\neg\exists f'\in K:\text{eqn}(f,f')]$$

A relation between two text units holds iff the corresponding frame oriented relation holds between two frames which are members of the units (def. 12). The semantic relations which hold between frames with respect to special properties (eg def. 8) are used to define relations on text units according to def. 12a. A relation on text units, which depends only on the interrelation of two frames may be a too weak restriction. A relation of partial identity (id_n) between text units demands that the intersection of their frame sets must have at least n elements (def. 13). The intersection between this relation and a frame based semantic relation may be used to enforce stronger restrictions (def. 14, def. 14a).

Def. 12:

$$\forall K_1, K_2 : rel_K(K_1, K_2) \leftrightarrow \text{unit-rep}(K_1) \wedge \text{unit-rep}(K_1) \wedge \exists f_1 \in K_1, f_2 \in K_2 : rel_f(f_1, f_2)$$

Def. 12a:

$$\forall K_1, K_2 : rel_K(K_1, K_2, f, s) \leftrightarrow \text{unit-rep}(K_1) \wedge \text{unit-rep}(K_1) \wedge f \in K_1 \wedge \exists f_1 \in K_2 : rel_f(f, f_1, s)$$

Def. 13:

$$\forall K_1, K_2 : id_n(K_1, K_2) \leftrightarrow \text{unit-rep}(K_1) \wedge \text{unit-rep}(K_1) \wedge |K_1 \cap K_2| \geq n$$

Def. 14:

$$\forall K_1, K_2 : rel_{k_n}(K_1, K_2) \leftrightarrow rel_k(K_1, K_2) \wedge id_n(K_1, K_2)$$

Def. 14a:

$$\forall K_1, K_2, f, s : rel_{k_n}(K_1, K_2) \leftrightarrow rel_k(K_1, K_2, f, s) \wedge id_n(K_1, K_2)$$

III.2 Pragmatic Relations

Pragmatic relations between text units reflect the situational context in which the dialogue between user and hypertext system takes place. This can be described by a variety of parameters - eg models of the discourse or the intentions of users or hypertext authors. In the following we will restrict our interest to two aspects of dialogues which are important for the design of hypertext systems and can be tackled by the formal instruments we have introduced above:

1. the amount of details the text units contain (D_U)
2. the specifity of the user's wishes - formulated as a query (S_Q).

The combination of these aspects allows to distinguish several dialogue situations. Weyer's "Dynamic Book" (WEYER 82) supplies the user with original text units, therefore D_U cannot be manipulated. Variations of specifity are gained on a syntagmatic level by alternatively presenting titles, subtitles or text passages. According to different S_Q's two prototypes of dialogue situations may be defined.

1. If S_Q is high - several terms are selected from the subject index - the system behaves like an encyclopaedia from which the user may derive information by dialogue.

2. Unspecific (or unknown) queries enforce browsing on the syntagmatic level, eg skimming the headlines of the next chapters.

Systems which are able to present information with several degrees of abstraction allow to adjust D_U to the dialogue situation. Intelligent tutoring systems adapt the level of abstraction heuristically based on an explicit model of the student and his presumable information needs that may be derived from this model (SLEEMAN 83). Information retrieval systems - on which we will focus our interest in the following - allow the specification of D_U - by selecting more general or more special index terms -and S_Q - eg by ranking the search terms. To what extent text units match a query - with respect to the content and the degree of abstraction - can be defined by a relevance relation relev(Q,TU,r), where Q is a query, TU a text unit and r the degree of relevance. In our retrieval model we use the explicit representation of the semantics of a text unit to define the degree of relevance by relating the text unit to the query as shown in the following definition.

$$\forall Q,F,r : \mathrm{relev}_{eqn}(Q,F,r) \leftrightarrow \exists T : F \in T \wedge r = \sum_{f_1 \in Q} \sum_{f_2 \in F} g_{f_1} \cdot \mathrm{chr}_{eqn}(f_1,f_2) + \sum_{L \in T} \sum_{f_1 \in Q} \sum_{f_2 \in L} g_{f_1} \cdot \mathrm{chr}_{eqn}(f_1,f_2)$$

In this formula *Q* represents the query, *F* the semantic representation of a text unit and *r* the degree of relevance.

(1) The relevance of a text unit is computed by summing up the weights g of those frames, which are related (by a predefined relation, here : eqn) to a frame of the query. The weight g, which can be defined by the user during the dialogue (see dialogue example in section IV.), is in the range from l to 10. chr is the characteristic function of this relation mapping all pairs (f_1,f_2) which are elements of the relation on l and all other pairs on 0.

(2) The relevance of a text unit can't completely be separated from the relevance of the text the unit is taken from. Therefore the overall relevance of the text (a set of text units denoted by T) is added to the relevance of the text unit itself.

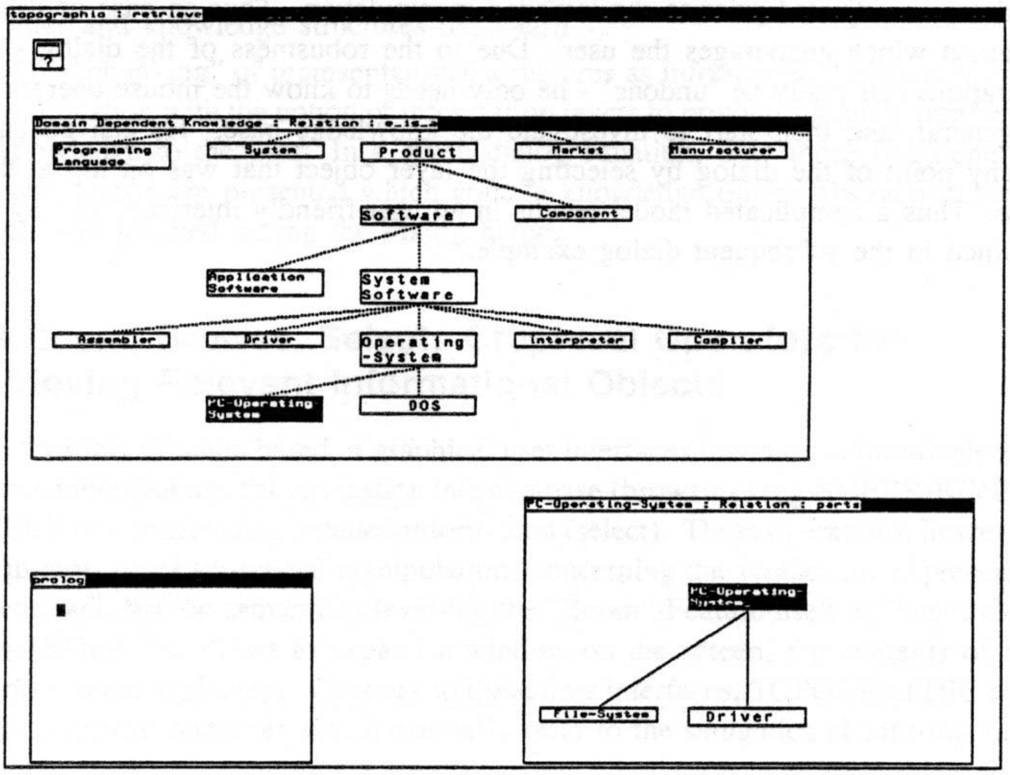

Figure 1

Recall and precision of the query may be adjusted by choosing an appropriate relation, which must hold between the frames of query and text unit The relation of name similarity (eqn see def. 5) is the most elementary of the possible tests. Instead of chr_{eqn} the characteristic function of the union of eqn and e-is-a may be applied in the relevance measure yielding a controlled expansion of the recall. (This is comparable to downposting operations in thesaurus based retrieval systems.)

IV. Guiding the User from Search Terms to Relevant Text Contents: A Dialog Example

After discussing the hypertext model we want to give some insight to the experimental information system TOPOGRAPHIC which supports knowledge based interaction facilities which provide content oriented access to text knowledge bases. The results from the text analysis and condensation process stored therein may be regarded as text units in the sense mentioned above.

We illustrate the essential features of the user interface - which supplies the user with a 'graphical retrieval language' meeting the needs of hypertexts and is based on the representation structures as defined above - by means of a (slightly simplified) dialogue. This example shows all layers of information that can be accessed in a series of zooming operations (which means switching from a more general to a more special text unit) in order to give an overview of the system's capabilities. On each layer shown the zooming is prepared by selection operations that facilitate focussing on relevant sections of the layer below. If the items to be selected are not visible due to the limited size of the screen, browsing is used to access them. (A real life dialogue may not have such a straightforward zooming structure, there might be 'loops' in it in cases the user returns to higher levels to change his focus up there and then zooms again. Thus a feedback facility for query refinement is given.)

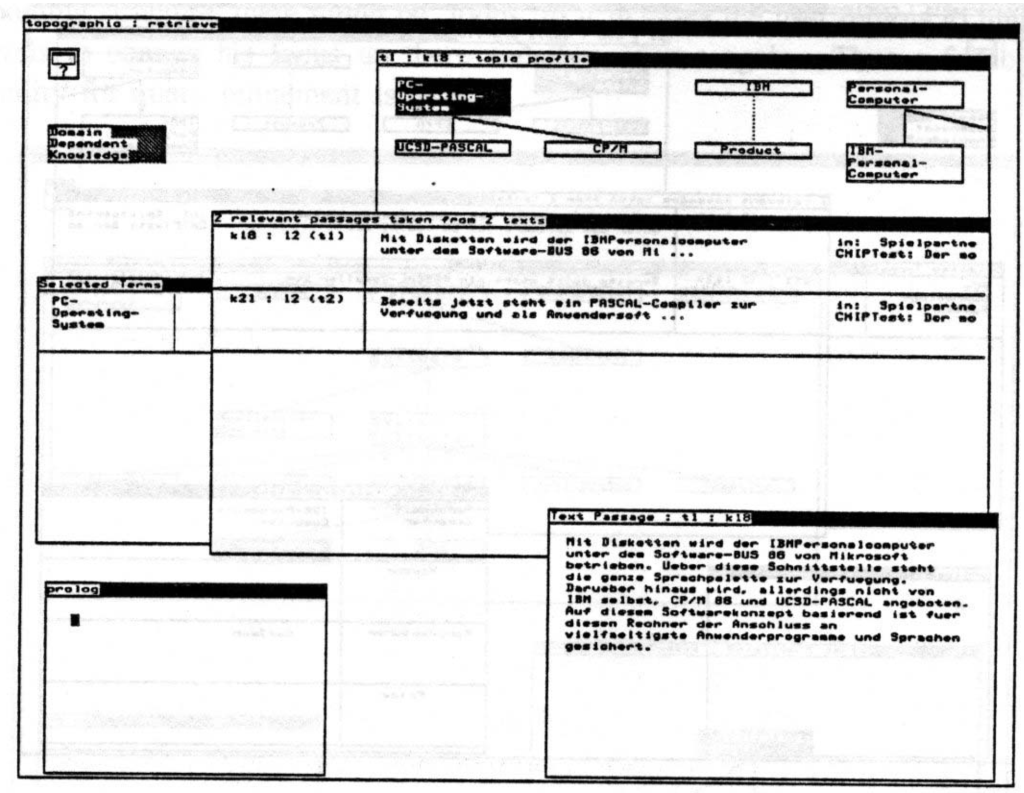

Figure 2

At the beginning of the dialogue the most general concepts of the world knowledge base (which can be thought of as a representation of a hypothetical text unit defining the terminology of the domain of discourse) are presented to the user so that he is informed about the domain of discourse. The user starts to explore this conceptual hierarchy by applying the browse option to the concepts 'Product', 'Software', 'System Software' and 'Operating System' (cf fig. 1). (He needn't know that they are frames, he only operates on graphical items.) To shorten the process of investigation, the user can enter search terms tentatively, which are not offered by the system at the time being. The command "find('Operating System')." entered via the "Prolog" window is equivalent to the browse-sequence mentioned above. Additional to the 'e-is-a' relation connecting the concepts (see def. 4) other relational dependencies of one concept can be shown on demand (such as the 'parts' relation which is similar to but not identical with the relation holding between an object and its parts (cf fig. 1)). While browsing the user constructs a query by selecting relevant terms (selected terms are presented in inverted mode). Zooming the window which presents the domain dependent knowledge on the taxonomic level yields a list of all selected terms and their activation weights indicating their relevance for the further retrieval process (cf fig. 2). (The weights may be increased or decreased if necessary.) A subsequent zooming of the 'selected terms' object produces a list of text passages which are related to the query by the relevance relation (see def. 15). The passages are ranked according to descending values of r. At the same time the graphical representation of the knowledge base shrinks to the format of a box due to the shift of the user's attention to information layers below. Bibliographical information (title etc.) about the text and a short textual extract of the beginning of each passage are given. In fig. 2 the semantic representation of the most relevant passage (topic profile of passage k18 of text t1) is shown. Applying the zoom operator to other list elements would reveal their representation, respectively. Zooming the node 'UCSD-PASCAL' in the topical network of the most relevant text part reveals the factual information about this PC-Operating-System that was extracted from this particular text part during the analysis process (cf fig. 3), whereas zooming the whole window results in the corresponding text passage (cf fig. 2).

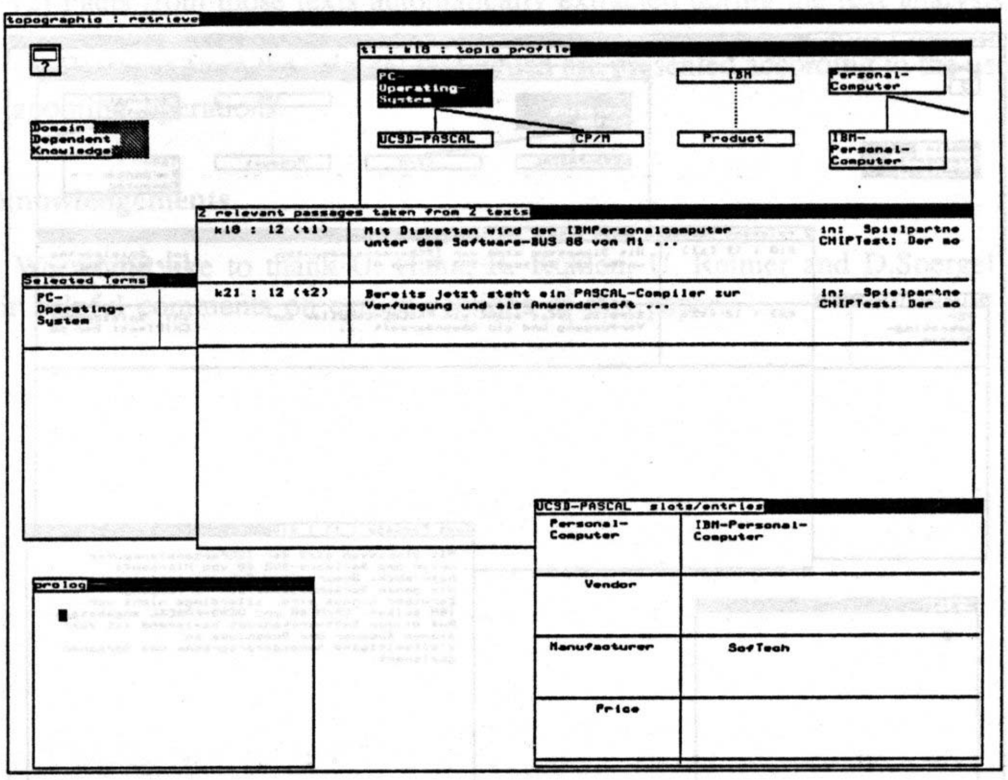

Figure 3

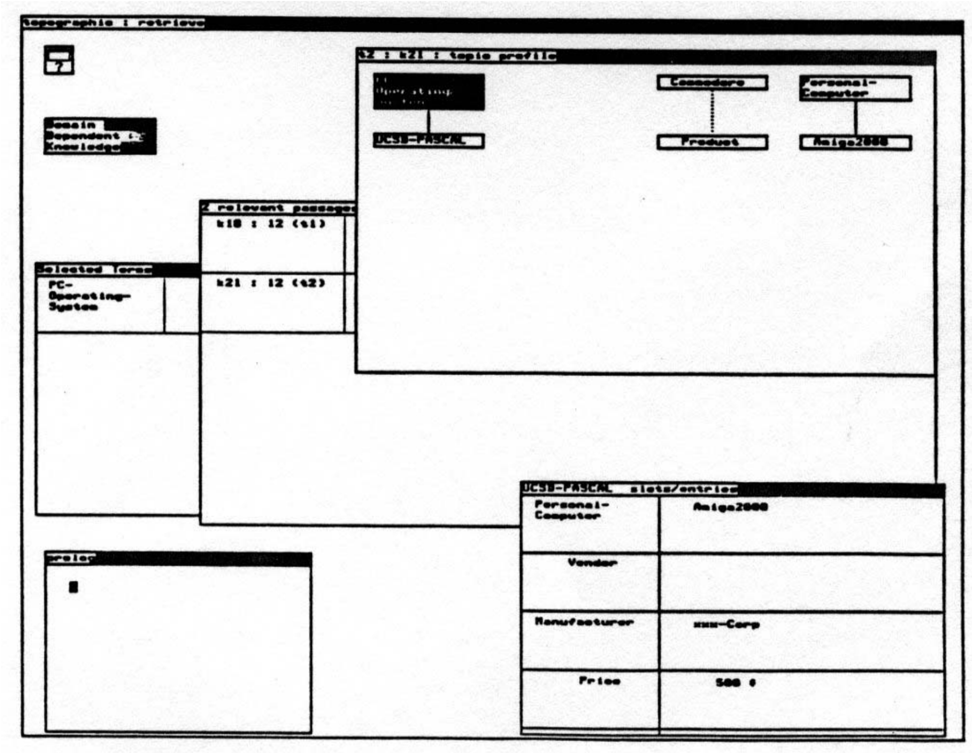

Figure 4

Switching between topic profiles may be accomplished not only by zooming from the list of text units but by browsing the current text profile as well. The user is then asked (a pop up menu will be displayed (cf fig. 3)) whether he wants to see the next relevant passage or another relevant one which is connected to the current text unit by a semantic relation (cf fig. 3). Only those relations will be offered which promise a successful continuation of the browsing process, i.e. lead to other text units. Thus, the combination of pragmatic and semantic or syntagmatic information about possible successors of the current text unit allows to restrict the set of new objects to be presented efficiently. In our example, the conjunction of requirements selected (all items of the menu in fig. 3 yields one salient text unit (cf fig. 4)) containing a frame named UCSD-Pascal as well. These relations may be inferred from the properties of the two frames according to def. 14 and def. 14a (definition of hypertextual relations by frame relations) and:

a) x-compl: all entries are different (def. 6).

b) alt-inf: the sets of entries of two corresponding slots are disjunct, i.e. the product is sold by different vendors (def. 9).

c) conf: there are different entries to corresponding slots which are singletons, i.e. supposed a product has no more than one manufacturer there is a conflict between the two frames (def. 10).

Again, this dialogue fragment is somewhat idealized, in real live dialogues a set of more or less qualifying text units might be obtained, which would then be presented in a table similar to the list of relevant text passages retrieved by the query (fig. 2).

(a remark on the example: TOPOGRAPHIC supports the retrieval of German texts, therefore the text example is taken from a German (computer-) magazine. This text is about software products available for the IBM-PC. For convenience, all identifiers occurring in the example have been translated.)

V. Implementational Remarks

The development of the TOPOGRAPHIC system is supported by BMFT/GID under contract '1020018 r. It is implemented in Prolog and C on a CADMUS 9200 with UNIX. The Prolog-system as used in TOPOGRAPHIC was developed as part of the project by augmenting the IF-Prolog interpreter with new built-in predicates. Additional to common features of Prolog it supports access to frame based knowledge bases and graphical tools for interface management, which are implemented in C for the purpose of efficient execution. The basic frame predicates used in the definitions above are a (small) subset of the predicates provided for knowledge base access. TOPOGRAPHIC's graphics-predicates include multi window and mouse interaction techniques as well.

References

CHRISTODOULAKIS ET AL. 86: Christodoulakis, S.; Ho, F.; Theodoridou, M.: The Multimedia Object Presentation Manager of MINOS: A Symmetry Approach. In: SIGMOD Record, Vol. 15, No. 2,1986, pp. 295-310.

DANSEREAU/HOLLEY 82: Dansereau, D. F.; Holley, C. D.: Development and Evaluation of a Text Mapping Strategy. In: Flammer/Kintsch (eds.): Discourse Processing, Amsterdam, North Holland, 1982, pp. 536-554.

HAHN 86: Hahn, U.: On Lexically Distributed Text Parsing: A Computational Model for the Analysis of Textuality on the Level of Text Cohesion and Text Coherance. In: Kiefer, F. (ed.): Linking in Text, Dordrecht, D. Reidel, 1986.

HAHN/REIMER 86: Hahn, U.; Reimer, U.: TOPIC Essentials. In: Coling 86: Proc. of the llth Conf. on Computational Linguistics, August, 25-29,1986, Bonn, FRG, ACL1986.

HAYES79:Hayes,P.J.: The Logic of Frames. In: Metzing, D. (ed.), Frame Conceptions and Text Understanding, Berlin, New York: De Gruyter, 1979, pp. 46-61.

HAYES/HENDRDC 81: Hayes, P. J.; Hendrix, G. G.: A Logical View of Types. In: Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling, SIGART Newsletter Nr. 74, Jan., 1981, pp.128-130.

MINSKY 75: Minsky, M.: A Framework for Representing Knowledge. In: Winston, P. (ed.): The Psychology of Computer Vision, New York: McGraw Hill, pp. 211- 277.

REIMER 86: Reimer, U.: A System-Controlled Multi-Type Specialization Hierarchy. In: Kerschberg, L. (ed.): Expert Database Systems. Proceedings of the 1st International Workshop, Menlo Park/CA: Benjamin/Cummings, 1986, pp. 173-187.

REIMER/HAHN 85: Reimer, U.; Hahn. U.: On Formal Semantic Properties of a Frame Data Model. In: Computers and Artificial Intelligence 4.1985. No. 4., pp. 335-351.

SLEEMAN 83: Sleeman, D. H.: Inferring Student Models for Intelligent Computer Aided Instruction. In: Michalsky et al. (eds.), Machine Learning. An Artificial Intelligence Approach, pp. 483-510, Palo Alto, CA: Tioga, 1983.

THIEL/HAMMWÖHNER 87: Thiel, U.; Hammwöhner, R.: Informational Zooming: An Interaction Model for the Graphical Access to Text Knowledge Bases. In: Yu; van Rijsbergen (eds.), Proc. of the lOth Annual Int. ACMSIGIR Conf. on Research & Development in Information Retrieval. New Orleans, Louisiana, 1987, pp. 45-56.

WEYER82:Weyer, S.A.: The Design of a Dynamic Book for Information Search. In: Int. J. of Man-Machine Studies, Vol. 17,1982, pp. 87-107.

WEYER/BORNING 85: Weyer, S. A.; Borning, A.H.: A Prototype Electronic Encyclopedia. In: ACM Transactions on Office Information Systems, Vol. 3, No. l, 1985, pp. 63-88.

WOELK ET AL. 86: Woelk, D.; Kim, W.; Luther,W.: An Object Oriented Approach to Multimedia Databases. In: SIGMOD Record, Vol. 15, No. 2,1986, pp. 311-325.