Links Between Electronic Documents and a Knowledge Base of Conceptual Graphs

Philippe MARTIN

INRIA - ACACIA project
2004, route des Lucioles - BP 93
06902 Sophia Antipolis Cedex France
Phone: (33) 93.65.76.45 Fax: (33) 93.65.77.83
E-mail:phmartin@sophia.inria.fr

Abstract. In this paper, we first show that: 1) Information Retrieval (IR) and hypertext may be knowledge-based and combined to provide the user a more efficient and effective environnement for accessing and retrieving information; 2) Knowledge Acquisition Tools represent document and needs to integrate hypertext and IR techniques; 3) the Conceptual Graphs (Sowa, 1984) formalism is interesting for IR, hypertext and their integration. Then, we present our integration of a structured document editor with hypertext capabilities to a CG workbench, namely Grif (Quint & Vatton 1992) to CoGITo (Haemmerlé, 1995). The result of this integration is a KAT named CGKAT, which enables to use and mix three kinds of document access/organization techniques: those relying on hypertext navigation, those using the document abstract structure, and those using semantic queries. Our integration relies on typed hypertext links between document elements and knowledge in the KB, which enables to use the KB of CGs which is built during the knowledge acquisition process, as a semantic index on documents. We show how browsing, queries, document generation can be done and multiple kinds of views on the documents and on the knowledge can be built. Our tool is dedicated to knowledge acquisition and could not be used as a large-scale hypertext or IR system.

Keywords. Knowledge Acquisition, Knowledge Representation, Knowledge Documentation, Conceptual Graphs, Hypertext, Information Retrieval, Structured Document.

1 Introduction

In Knowledge Acquisition (KA), electronic documents are used and represented because they are *expertise sources*, e.g. interview transcriptions and technical reports, or because they are *generated* partly from a Knowledge Base (KB), e.g. specification documents, technical documentation and documents given by the Knowledge-Based System (KBS) to explain its knowledge or its reasoning. A Knowledge Acquisition Tool (KAT) should then include some hypertext capabilities for enabling its users to retrieve and structure information from documents. This is useful for the knowledge extraction and modelling process, or for the KB validation, since for example various pieces of information scattered in documents may detail different aspects of the same knowledge or may detail this knowledge at different levels. Query mechanisms, that is Information Retrieval (IR) mechanisms, are also necessary to provide a direct access to information. Additionally, queries can be composed and user tailored views can be built by queries filtering the hypertext network.

Besides, knowledge-based approaches have gained much attention recently both in IR systems (Croft, 1987) and in hypertext systems (Nanard & al., 1994). In IR systems, instead of keyword-based representations of document and user queries, more semantically-based representations help to reduce the gap between the semantics of documents and the actual representation to be processed by an IR system and make automatic reasoning more amenable (Myaeng, 1992). Formal representations are also interesting in hypertext systems for similar reasons. Second generation hypertext systems enable to type document elements and to set between them typed links for expressing the intended semantics of the elements and of their relationships. The hypertext networks of Aquanet (Marshall & al., 1991) and Sepia (Streitz & al., 1992) rely on frame-based knowledge representation which still does not offer reasoning capabilities, while the hypertext network of MacWeb (Nanard & Nanard, 1993) relies on a semantic network which has some computing capabilities. More and more hypertext systems integrate a logical query language enabling to access nodes in the network according to their types, attributes and structural properties (the way they are connected to each other) (see for example (Consens and Mendelzon, 1990) and (Beeri and Kornatsky, 1990)). Conversely, hypertext techniques may supplement conventional methods of IR by allowing users to discover retrieval cues that successively can be used for query formulation.

To sum up, we think with others (e.g. (Lucarella 1990), (Myaeng, 1992), (Nanard & al., 1993), (Kheirbek & Chiaramella, 1995)) that IR and hypertext may be knowledge-based and combined to provide the user a more efficient and effective environnement for accessing and retrieving information. We have also seen that such an integration is interesting in a KA context. In this paper, we first briefly present some advantages of Conceptual Graphs (Sowa, 1984) for IR, hypertext, and their integration. Then, we present our integration of a structured document editor with hypertext capabilities to a CG workbench, which are respectively called Grif (Quint & Vatton 1992) and CoGITo (Haemmerlé, 1995). The result of this integration is a KAT named CGKAT, which enables to use and mix three kinds of document access/organization techniques (with separated tools, these techniques can only be pipe-lined): those relying on hypertext navigation, those using the document abstract structure, and those using semantic queries. We will compare CGKAT to the KA oriented hypertext system MacWeb which also enables to mix these three kinds of access/organization techniques (Nanard & al., 1994). However, CGKAT is aimed for KA and it cannot be used as classic hypertext or IR systems, i.e. as large-scale hypertext or IR systems, because at present time: 1) it does not include a natural language processing module for generating representations for document elements and their relationships (instead it provides a rich default concept type lattice and relation type hierarchy, and exploits the semantic word database WordNet for proposing various concept types given a lexical entry (see (Martin, 1995) for more details); 2) IR by queries can only be done with the question-answering system of CoGITo which can project the CG query on the CGs of the KB, and generalizes the query if no answer has been found; there is no IR suited graph matching functions which deal with the two standard measures of IR effectiveness: precision and recall; 3) it does not include hypertext navigation guiding mechanisms like paths, historics, or activity detection.

Our integration relies on typed hypertext links between document elements and knowledge in the KB. Thanks to Grif, these document elements can be not only text portions like words or sentences but also graphic elements, images, or any composition of the previous kinds of elements, e.g. a graph, a section, a chapter, and a whole article. We have distinguished two kinds of links between an element and knowledge in the KB, the link of "representation" and the link of "annotation": the first connects an element to its "representatives" (formal representations of the element and its content) and the second connects an element to its "annotations" (formal pieces of knowledge relative to the element and which does not represent it and its content only, e.g. the representation of a relationship between the element and another element). These typed hypertext links, and the indexation of representatives and annotations by various kinds of informations enables to use the KB of CGs which is built during the knowledge acquisition process, as a semantic index on documents. We will show how browsing, queries, and document generation can be done and multiple kinds of views on the documents and on the knowledge can be built.

First we detail the interest of CGs for IR and hypertext, then we see how various techniques for accessing or organizing documents may be integrated, and we present our integration and its features. An example is then showed and some additional facilities provided by CGKAT listed. Finally we compare our tools with similar ones.

2 Conceptual Graphs for Information Retrieval, Hypertext and Their Integration

We sum up here some arguments in favour of the CG formalism for implementing IR and/or hypertext techniques.

According to (Myaeng, 1992), IR is a process of identifying all and only those documents that satisfy a particular user's information needs. Myaeng lists the features of the CG framework which are advantageous for IR:
- CGs can represent the meaning of natural language sentences of a document or a problem description, and also the interconnections of these sentences. Additionally, complete information is not obligatory: CGs can be manipulated with no implication of existence of the individual being described, and discourse referents problems can be handled without coreference resolution (Sowa, 1990), e.g. with the indexical referent marker #;
- the type lattice and the abstraction mechanism using a "lambda expression" not only allows for a relatively easy modelling and handling of different levels of details of information needs and document contents but also facilitates efficient knowledge processing (e.g. by conceptual graph matching);
- CGs can be seen as a system of logic that has a model-theoretic semantics, but also allow informal, non-deductive reasoning with prototypes and schemata. This is important since information needs are seldom expressed precisely and completely, and since there are many ambiguities in natural language texts.

According to Sowa (1992), an entire hypertext network could be represented by a giant CG, where the referent field of each concept could contain text, nested CGs, or any other type of lexical object (e.g. picture, video). Browsing could take place on relations between concepts, and inference could be done since CGs have the full expressive power of logic. Sowa also notes that CGs have a graphic structure with some useful features for representing various kinds of information graphically (including Petri nets, flow charts, entity-relationship diagrams, etc.).

In order to integrate IR and hypertext techniques, Kheirbek & Chiaramella (1995) have implemented in the IR system RIME the above Sowa's idea: a document element (i.e. here often a whole document) is represented by a concept of such a type as a data type (e.g. Text, Hyperdocument, Structured-Type) and is related to others concepts by conceptual relations for expressing its external attributes (author, date, editor, title, etc.), its content attribute (a concept of type Graph which represents it), and its structural and semantic links with other document elements. The semantic network can be automatically built from SGML documents (Goldfarb, 1990) by transformation of SGML structures into CG structures. Although the interface is not discussed, it seems that the user can browse in this semantic network, in the concept type lattice, in the relation type lattice, and between them. Then these lattices can be used as indexes over the semantic network and help the users to reorient themselves when they are lost. Document elements may be retrieved using IR matching functions based on projection. In order to index a document and ensure an optimal retrieval precision, for each document, a transitive closure of the content attributes of the document elements is built: the content attribute of a document element E which is composed of the sub-elements E1, ...,En, is built by a directed maximal join (Nogier, 1991) on the content attributes of E1, ...,En.

According to (Kheirbek & Chiaramella, 1995), the CG formalism was adequate for such an integration since: 1) it has a great expressive power that allows the definition of almost any kind of knowledge; 2) it has powerful formal semantics (first order logic) and recent extentions of the theory (Wuwongse & Manzano, 1993) allow to deal with uncertain knowledge, which is a very important notion for designing models and their uncertain-matching functions; 3) it supports efficient implementation.

Thus the CG formalism seems to be a good support to implement IR and hypertext techniques. However, some extentions will have to be done for some applications. For instance, in order to enable powerful but efficient logical queries, Beeri and Kornatsky (1990) had to choose an extension of modal logic, since first-order logic requires a least fixed point for queries on structural properties of nodes, and first-order quantifiers are costly to process.

3 Integration of Document Access/Organization Techniques

Let us list the techniques classically used for organizing and then accessing documents. A document may be merely tagged or represented by some keywords. It may also be structured, i.e. represented as a logical (or abstract) structure which combines elements of different types, as in SGML (Goldfarb, 1990). The logical structure is defined in a Document Type Definition (DTD) which specifies all types of elements that can be used. In Grif (Quint & Vatton 1992), a DTD also specifies all structural and hypertext relation between elements[1]. Databases may be used to manage and retrieve information based of the logical structure of the documents. Hypertext systems enable to navigate on links between information. First generation hypertext systems only offer hard-wired links. Second generation hypertext systems such as Aquanet (Marshall & al., 1991) and MacWeb (Nanard & Nanard, 1993) rely on a knowledge representation approach[2], which enable more powerful queries and dynamic (calculated) links. Finally, text understanding approaches like those based on the CG formalism enable then some conceptual queries in more or less artificial languages.

The complementarity of these approaches and specific information management needs (e.g. concurrency control, transaction management, object versions, distribution or efficient query language facilities) have led to some integration of existing hypertext systems with current relational or object-oriented database systems (e.g. (Chen & al., 1990) and (Gallagher & al., 1990)).

Nanard & al. (1994) have realized a more knowledge-based integration of these approches within the object-oriented hypertext system MacWeb. The model of this system which enables this integration relies on a four level organization : 1) the *Information level* which is a SGML-like document structure description, 2) the *Knowledge level* which is an hypertext *and* semantic network of concepts and their relationships, 3) the *Anchoring level* which maps these last two levels by associating to each represented document element, the concept in the semantic network which represent it (this is detailed below), and 4) the *Task level* which contains document generation scripts associated to task or domain concepts of the semantic network. The user may navigate 1) at the Information level by following the logical structure or the explicit inter-references of this level; 2) at the knowledge level by following the various relations between concepts; and 3) between the Information level and the Knowledge level via the anchoring level. The user may also make explicit queries on the Anchoring level to select parts from document which directly match some given properties, or activate a script which generates a "virtual document" by collecting and structuring the answers to the queries in the script (more exactly the document parts corresponding to the concepts which are answers to the requests). Thus, this structured collection of document parts is a task-oriented view on the documents. This "virtual document" may be modified (then the original sources of the document are modified) and may be a departure point for hypertext searchs. To sum up, various kinds of document access/organization techniques may be interwined and combined.

The philosophy of our integration of document access/organization techniques in CGKAT is similar. But contrary to MacWeb, CGKAT integrates and exploits two different existing tools: a CG workbench (CoGITo) and a structured document editor with hypertext capabilities (Grif). CoGITo enables to represent knowledge with CGs and to make some operations (like projections) on them (the MacWeb representation language enables inheritance

---

1. According to (De Young, 1990), three primary advantages of identifying and supporting structures on hypertext are: 1) the user can more easily developp a mental model of what the system contains and can maintain orientation when navigating through the system (this is due both to reduced cognitive complexity through indentifying cohesive block of hypertext and familiarity with specific structures); 2) the system can be tailored to support specific linking structures, particularly at the user interface level, and assist the user by organizing and analyzing information; 3) more efficient implementations may be possible if the structure is known in advance.

2. The high and explicit structuring of a KB is not only a good support for hypertext navigation but also prevents user disorientation as Bernstein (1990) shown it.

of attributes between concepts, but has no quantifier nor graph-matching operation). Grif is dedicated to the handling and presentation of document logical structures (the Information level in MacWeb), enables to set typed hypertext links on any document element, and enables to create graphic representations of the knowledge in CoGITo[1]. Hence our way of connecting the Information level (the document elements) to the Knowledge level (the concepts and CGs of CoGITo) is different from the one of MacWeb (the Anchoring level). The Anchoring level associates to each represented document element, the concept in the semantic network which represents it, plus the context of the element (i.e. the smallest part of the document surrounding the element, which has an autonomous full meaning) and a type for this context (e.g. definition, exception, proof). In CGKAT, there is not such an Anchoring level: a bidirectional typed hypertext link may directly relate a document element to the graphic representation of the concept which represents the element, and if necessary, the context of this element (which is a document element) may be connected in the same way to a concept. Thus, all the knowledge on the document elements is uniformly represented at the knowledge level (in MacWeb, it seems that document elements cannot be composed, hence our option could not have been taken). Additionally, in the CG formalism *the content of a document element may be further described by a CG inside the concept which represent the element* (hence contexts can be handled; this is developped below). Semantic relationships between elements can be represented by conceptual relations between concepts.

Thus, the user of CGKAT may browse hypertext links at the Information level, at the knowledge level, and between them. The whole knowledge of the KB can then be used as a semantic index on the documents. CGKAT enables to navigate on the concept type lattice, on the relation type hierarchy, and also between the occurences of a given concept in the CGs of the KB. Queries can be also be done on the KB of CGs using a question-answering system that uses projection and query relaxation techniques (Carbonneill & Haemmerlé, 1994). The *result of a query may be CGs or, as in MacWeb a generated virtual document which concatenates the document elements corresponding to the CGs answering the query.* More precisely: 1) the elements of this virtual document are "inclusions" (in the sense[2] of Grif) of the previous document elements; 2) a virtual document is also built for presenting the CGs answering the query. Hence, *views can be dynamically built on the documents and on the KB*. A CGKAT virtual document element may be restructured or completed by the user, but it is initially just a collection of inclusion of document elements: we have not implemented like in MacWeb a script language which would structure the answers of a query or of many queries.

To our knowledge, the MacWeb semantic network cannot handle contexts. Then, the representation of a document element can only be a concept: without context there is no way to isolate a graph (a distinguished portion of the semantic network) which will represent the internal content of the document element. With the CG formalism, such a graph may be in the referent part of the concept, and this concept can be related to other concepts (e.g. for expressing semantic relationships between elements). In CGKAT, a document element may be

---

1. Grif enables its users to edit a document structured in various elements, the possible orders and presentation of which are formally specified by DTDs and presentation models (see "The languages of Grif" (Quint, 1992) for more details). Then the manipulation of these elements (e.g. selection, creation, attribute adding, hypertext connection and presentation change) may be done by the user via the editor, or by program via a C functional interface called the "Grif Editing Tool Kit". Our tool exploits this editor and this interface for offering a graphic interface for the CGs creation, display and update : the graphic representation of concepts and relations are document elements. Thanks to the DTDs, the presentation models may be updated easily and hypertext links may be set between them and other document elements. Of course, the graphic representations always reflect the knowledge entities in the KB, then from now, we will not make distinctions.

2. In Grif, an inclusion of a element is a copy of this target element, and an hypertext link connect the inclusion and the target element. This copy is "alive" in the sense that all change made in the target element are automatically reflected in the copy (which hence cannot be directly modified). Inclusions are a way to share and reuse information which CGKAT exploits a lot. We also exploit a lot the fact that, from an element which is the target of hypertext links, all the sources of these links may be retrieved (via the editor or by program).

connected to the list of concepts which represent it by an hypertext link of type "Representations" (we explain below why a list is needed instead of a single concept). These concepts are then called "representatives". An element may also be connected to a list of CGs (which were not built for representing it) by an hypertext link of type "Annotations". These CGs are then called "annotations". An annotation may *for example* relate representatives for representing a relationship between elements. A CG in a representative R may also contain representatives for representing a relationship between elements but only if these elements are sub-elements of the one represented by R (otherwise this CG would not be a description of the content of this element represented by R).

CGKAT enables a document element to be represented differently by various users, and differently according to various views (e.g. goals, knowledge models or knowledge categories), then a document element is not connected to a single representative but to a list of representatives, each one being indexed by its creator and the view to which it belongs (this last sense of "view" is not so different from the sense of view used in the previous page since CGKAT takes this indexation into account for building virtual documents enabling to see the parts of the KB corresponding to these views). In order to respect the semantics of the hypertext link "Representation", CGKAT allows only one representative by user and by view. The handling of views is implemented via the concept type lattice: "AnyView" is the supertype of all types of view, including the user-defined ones. "AnyView" is used when the representation is not relative to a special view, e.g. for a "literal" representation. This is the default view type proposed by CGKAT. A relation cannot be set between two representatives of two different views unless one of the views is a supertype of the other, or unless this is explicitly allowed (see next section).

Annotations are also indexed with their creator name and the view they fit in, but there is no restriction on their number. Other information are stored with annotations and representatives (see Figure 2 or 3): the creation date, the context of use of the document element (this is useful essentially when the document element is a word) and a comment. We will soon implement a filtering mechanism on the answers to a query for selecting the ones which correspond to a given creator or view. Let us note that for generating a virtual document which concatenates the document elements corresponding to the CGs answering a query, only the representation links are exploited.

CGKAT provides a supplementary method to access/index the documents with the knowledge of the KB: this is an index of the represented terms. This index is a synthesis of the representations by the users of the occurrences of words or compound words in a document. This index is connected by hypertext links to the sources of information it synthesizes. Such indexes were relatively easily implemented using the inclusion mechanism, the Grif languages and editing toolkit (Quint, 1992) and the Grif index facility (Richy, 1994). More details on the indexing principles can be found in the last refered paper. An interesting property is that if a document includes other documents, its index can be built by the merge of the sub-document indexes. Lemaire (1995) details the interests of such an index for KA and knowledge sharing.

Of course, CGKAT also allows to build CGs without any links to document elements, but still indexed by their creator name and a view type. Such CGs may or may not include representatives. We now deepen some semantic restriction on the representatives and their inter-relations.

4 Semantic Constrainsts on Representation of Elements and Their Relationships

Let us first introduce some ontotological distinction with an extract of (Sowa, 1992).

> In conceptual graphs, a situation is represented by a context, which is a concept that contains one or more propositions that describe the situation. The propositions in a context could be expressed by a paragraph of English sentences or by a collection of conceptual graphs.
>
> ... The following graph represents the sentence *There exists a situation described by a proposition stated by a sentence represented by the string "A cat is on a mat"* :
> [Situation]->(Dscr)->[Proposition]->(Stmt)->[Sentence]->(Repr)->[String: "A cat is on a mat"].
>
> There are three possible contractions for simplifying this graph to just a single concept:
> [Situation]->(Dscr)->[Proposition]->(Stmt)->[Sentence: "A cat is on a mat"].
> [Situation]->(Dscr)->[Proposition: "A cat is on a mat"].
> [Situation: "A cat is on a mat"].

This makes a clear distinction between statements (sentences or graphs), the propositions they express, and the situations described by these propositions. More generally, we think that any document element which is not a word or a compound word, is a statement of a proposition which describes a situation (i.e. a real or imaginary state or process). Such an element may be for example a group of words, a sentence, a section, an image and a graph. Words might eventually also refer to propositions describing situations but most single words refer to abstract or concrete entities or properties. Since the data types of document elements (e.g Text, Image, Section or Chapter) and their structural relations are already accessible and handled in the logical structure of a structured document, and since CGKAT is a KAT and not an IR system, we think it is an useless complication to represent them with concepts as it is the case in RIME (Kheirbek & Chiaramella, 1995))[1]. Hence we decided that document elements would be directly represented by concepts which express their meaning, e.g. concepts of type Entity, Action or Property for words and compound words, and concepts of type Proposition (or a subtype of Proposition like Observation or Hypothesis) for document elements describing a situation.

A document element describing a situation could not be directly represented by a concept of type Situation, since for example it would be a non-sense to relate such concepts by conceptual relations representing semantic relationship between document elements. The CGKAT default relation type hierarchy provides and structures from various points of view, many relation types for representing such relationships: rhetorical relations types (e.g. the ones of Mann & Thompson (1987): Summary, Restatement, Concession, Antithesis, Circumstance, Evidence, Cause, Purpose, etc.), argumentation relation types (e.g. the ones of Schuller & Smith (1990): Suggestion, Answer, Objection, Contributes, Contradicts, etc.), etc. Such relations may be used for hypertext navigation, document authoring (Schuller & Smith, 1990), explanation generation (Cawsey, 1991) or for studying dialogs between experts (Baker, 1992). The CGKAT default concept type lattice also provides and structures concept types that may be used for representing document elements describing a situation: the ones of Schuller & Smith (1990) (Issue, Position, Argument, Fact), the ones of Toulmin (1958) (Datum, Claim, Warrant, Backing and Rebuttal), etc. Once representatives are connected to other concept, e.g. others representatives, they may be searched by projection. Here is an example of query: [Argument:*x]<-(Evidence)->[Proposition:*y]->(Possible) ?

We have seen in the previous section that a conceptual relation cannot be set between two representatives which are indexed by two different view types unless one of the view type is a supertype of the other, or unless this

1. Of course, when the data types of document element and their structural relations are not represented with the CG formalism, projection cannot be used to search document element according to these characteristics. But projection can be done to search document element on their semantic content and then answers can be filtered according to the data types of the retrieved elements and their structural relations. In a knowledge acquisition context, there is rarely a high number of documents to work on, and the data types of the documents or of the document elements, or their structural relations, are much less interesting than their semantic content and semantic relations. Hence, at present time CGKAT does not enable to query on the data types of document elements or their structural relations.

is explicitly allowed. The last part of this sentence refers to the types of the representatives: if these types (which may be subtypes of Proposition) are also subtypes of a same view type, then CGKAT allows that the relation be set (of course, the concept types must also match the relation signature).

We now detail on an example how document elements can be represented in CGKAT.

5 An Example

Figure 1 is a Grif editor window showing a small part of an expert interview transcription (the expertise domain is car accident analysis/diagnosis). The underlined words or compound words have been represented by concepts of type Entity, Action, or Property. CGKAT exploits the semantic word database WordNet for proposing adequate concept types (with their supertypes) for a selected word in the document (see (Martin, 1995) for more details). Colored quotes highlight the elements represented by a concept of type Proposition. Figure 2 is another Grif editor window showing the representations of the highlighted compound word (actually only one representation has been done for this occurrence of this compound word). Figure 3 shows the representations of the document element between the highlighted quotes in Figure 1: two representations are visible, a "literal" one and a second which only represents the task decomposition information that can be extracted from the represented element (this is just an example, other relations than Succ and Subtask could be used for representing a task decomposition). These two representations are then partly redundant.

Let us call E the element represented in Figure 3, and E1 and E2 its two represented sub-elements. The first shown representative of E has a CG in its referent part which includes some representatives of E1 and E2 and relates them to represent a rhetorical relation between them (Elaboration). Each of these representatives is an individual concept of type Proposition (it represents an unique document element) and has a referent whose name is automatically built by concatenating the name of the source document, the identifier of the represented element, the user name of the representative creator and the name of the view (this encoding provides an unique individual referent and eases searching and filtering). Let us call R the first shown representative of E, and R1 and R2 the shown representatives of its sub-elements. In Figure 3, R1 and R2 are graphically included in the CG of R by the Grif inclusion mechanism (the same mechanism is used for the concepts in the CGs of R1 and R2). Then, the target elements of these inclusions may be directly accessed. Conversely, from one of this target element, all the CGs which includes it may be retrieved (this enables to access all the uses of a concept). The CGs in R1 and R2 are visible but they need not to be present in the CG of R. An expanded linear form for R is then:

[Proposition: #JLInterview_L466_phmartin_AnyView]->(Stmt)->[Graph:
    [Proposition:#JLInterview_L380_phmartin_AnyView]->(Elaboration)->[Proposition: #JLInterview_L413_phmartin_AnyView]].

Then, we see that CGs need not to be nested on more than one level for representing nested document element. This is important for search by projection.

In the CG in R2, we find a concept of type Task and the description of this task *à la* KADS (Wielinga & al, 1992). The relation types DI, DO and SI respectively mean "Dynamic Input", "Dynamic Output" and "Static Input" and come from the relations which usually link "knowledge sources" and "role" in a KADS inference structure. We could also have used the relations that the KOD knowledge acquisition methodology (Kuntzmann-Combelles and Vogel, 1988) advices for representing knowledge. We claim that the knowledge representation scheme of some knowledge acquisition methods can be defined within the CG framework and that then most KA methodologies may be followed with CGKAT (However, (Lukose & al., 1995) lists some requirements that the CGs should fulfill in order to be more useful in KA). We are presently defining within the CGKAT default concept type lattice and relation type hierarchy, the concept types, relation types and generic model needed to follow the KADS and/or the KOD methodologies.

A represented elements is directly connected to the list of its representatives. Another hypertext link connect each representative to the element it represents.

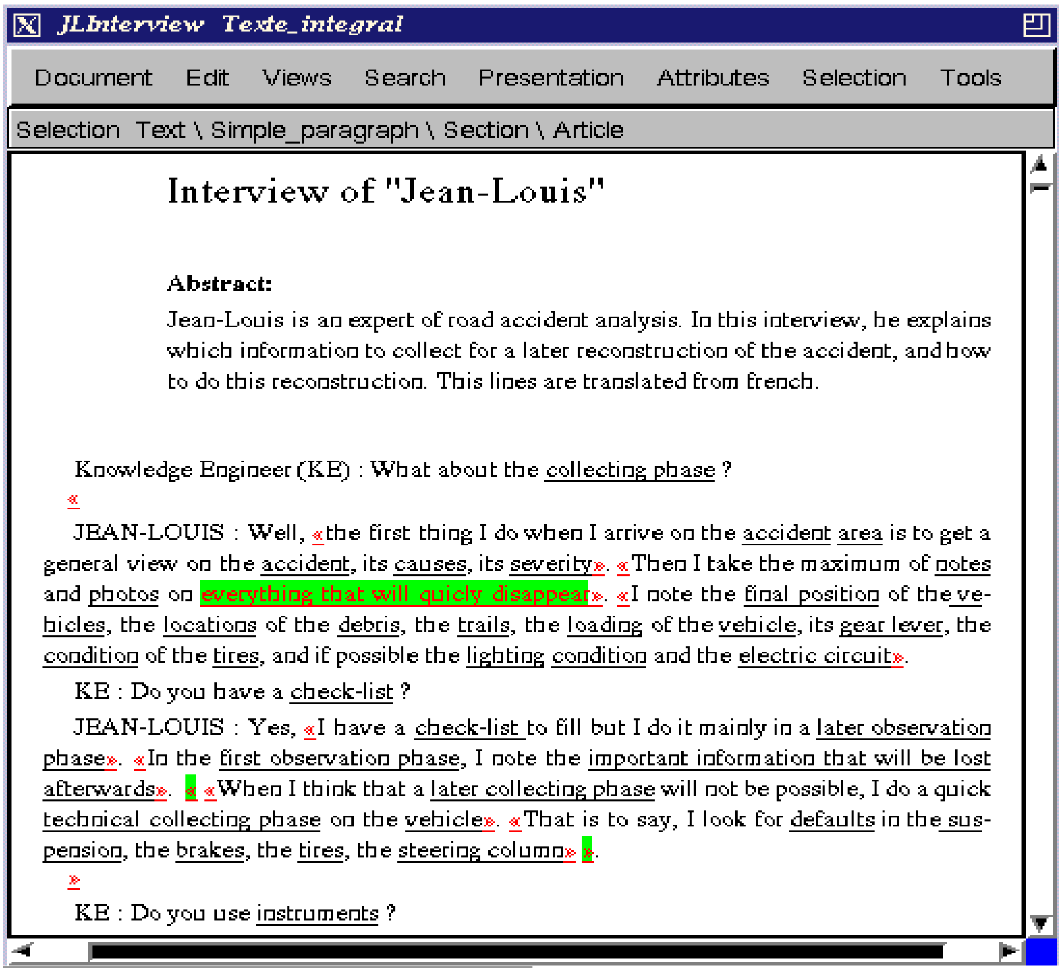

Fig. 1. An expert interview retranscription, some element of which have been represented

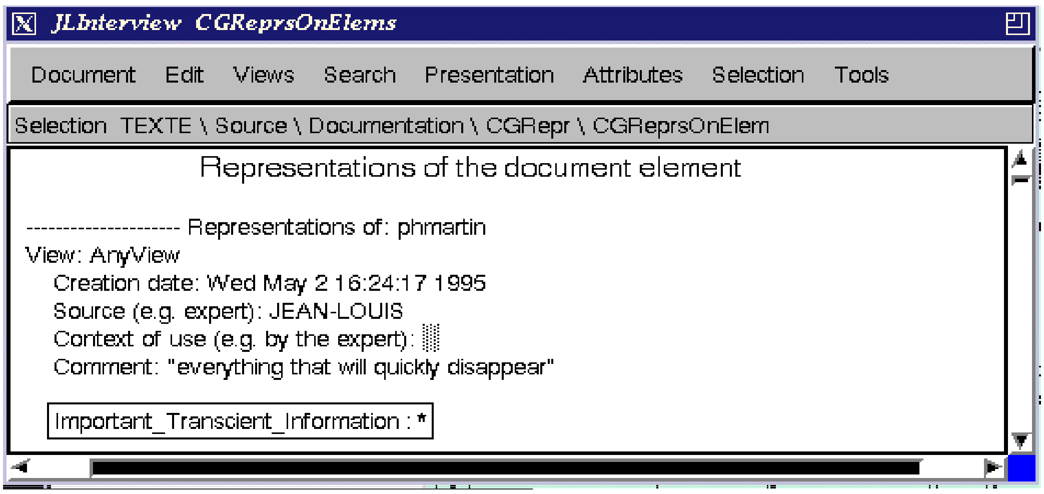

Fig. 2. A representation of the highlighted compound word in Figure 1.

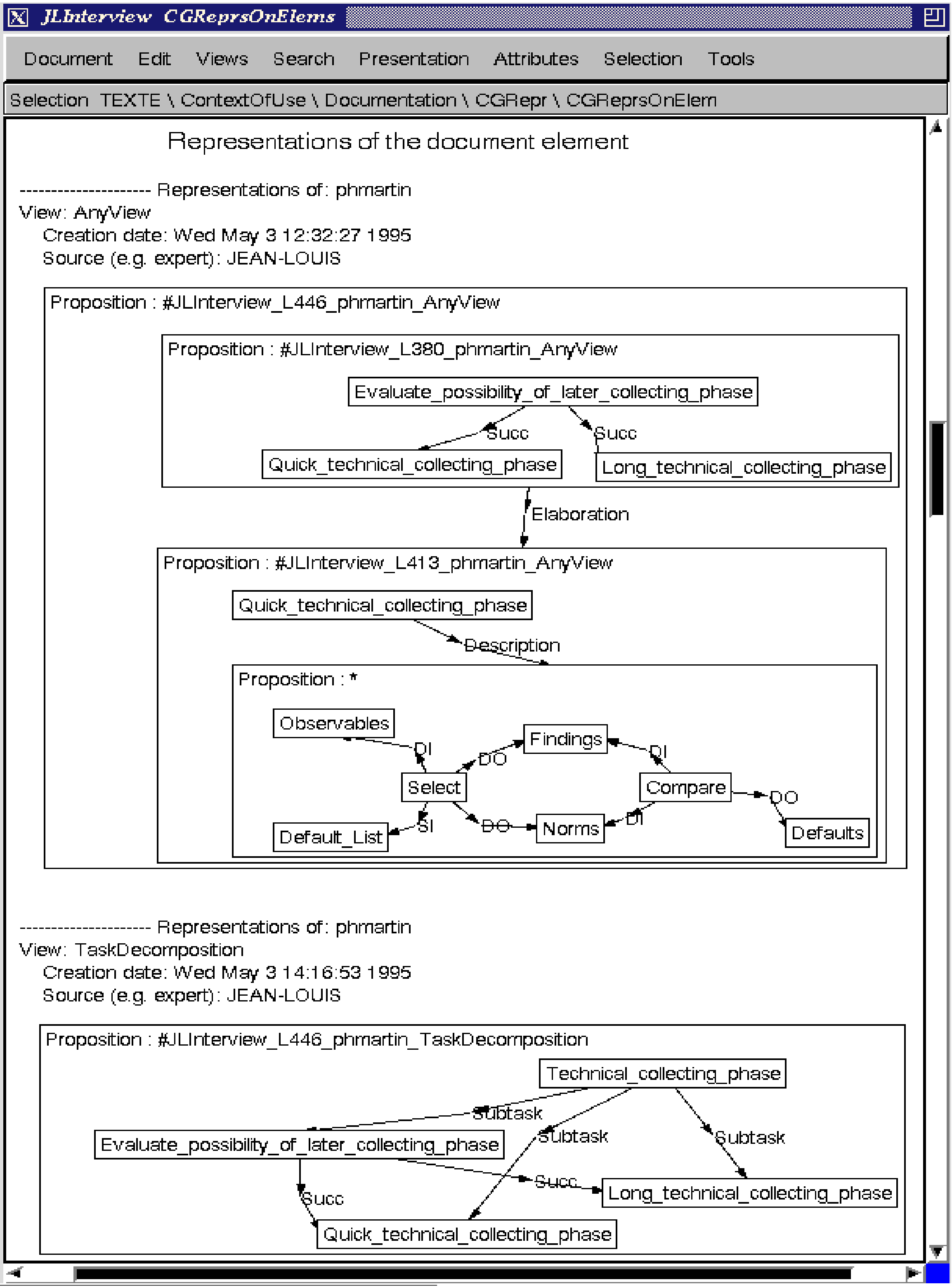

Fig. 3. A representation of the element between the highlighted quotes in Figure 1.

6 Additional Facilities for Knowledge Acquisition

We have seen that the Grif inclusion mechanism enables to share information between CGs. In addition to the CG context mechanism, this enables the knowledge engineer to easily create modules. Furthermore we added a view index to each CG. This index may be used to class knowledge in various categories, e.g. the KADS knowledge levels (Domain, Inference, Task and Strategy). Bürsner & Schmidt (1995) have deeply studied how view mechanisms like the CGKAT one can be used during the knowledge extraction and modelling phases of KA, in a bottom up approach and/or in a top down approach.

Let us note that in the CGKAT framework, the notion of view introduces a new dimension to the rhetorical or argumentative relationships: it is possible to represent a relationship between elements according to a given point of view. For example, it is possible to represent the fact that a document element is a summary of another document element *for information related to such subject, task or knowledge category.* This is especially useful when the user cannot or does not want locate in the document element the distinct parts which would entirely be relevant for the focused view.

Given a view V and a document element E which has represented sub-elements, CGKAT can propose representatives for this element by doing a maximal join on the CGs of the representatives for the view V (or a subtype of V) of the sub-elements of E. This helps the knowledge engineer to build views by proceeding incrementally from small document elements to bigger ones.

We will also soon implement a shortcut for simplifying the work of the knowledge engineer: instead of building representatives and then synthesize them in a single structure, s/he will be able to work only on this structure and to activate the generation of representatives by selecting a document element and a part of the structure which represents this document element. This way of building the KB seems easier in a KA context.

7 Comparison With Other Tools

CGKAT integrates three kinds of techniques: those relying on hypertext navigation, those using the document abstract structure, and those using semantic queries. However, CGKAT is tailored to KA and we listed in introduction the reasons why it could not be used as a classical (i.e. large-scaled) IR or hypertext systems. Myaeng (1992) notes that in (large-scaled) IR systems, the main criterion for judging the quality of retrieved text is "aboutness", and hence 1) "the IR process is different in nature from knowledge processing for question-anwering", 2) "the limitations of the domain-dependent nature of state-of-the art natural language techniques can be leessened to a great extent" that is the representations of documents need not to be as detailled as for knowledge retrieval (as opposed to Information Retrieval). This note also applies (to a certain extent) to large-scale hypertext sytems. In large-scale IR or hypertext systems, the semantic indexation of text is done by natural language processing techniques. In knowledge acquisition, the knowledge extraction and modelling from textual knowledge sources is manual and often guided by complex models.

We have seen how RIME (Kheirbek & Chiaramella, 1995) relies entirely on CGs for integrating IR and hypertext techniques. However the goal of RIME is still to be an IR system, not a KAT. CGKAT relies on CGs but also on the structured editor Grif which handles the logical structures and presentations of the documents. Thanks to this and to the Grif inclusion mechanism, CGKAT is able to produce virtual documents like in MacWeb, but it can also handle and represent embedded document elements. An originality of CGKAT is to handle many representations of document elements (one by user and by view) and to enables the represention of relationships between document elements according to views. Hence the notion of view is well exploited, and Bürsner & Schmidt (1995) have shown that this if of a great help in KA. Finally, a lexicon is provide for synthesizing the representations by users of the occurrences of words or compound word in a document.

KATs have generally very restricted hypertext capabilities and no IR mechanism, e.g. Shelley (Anjewierden & Wielemaker, 1992) and the K-Station (Albert & Vogel, 1990) which respectively enables to follow the KADS and KOD methodologies. These KATs only enable to associate parts of unstructured documents to entities of the KB. and rhetorical or argumentative relations between document elements cannot be represented. Actually the hypertext system of these KATs is only designed for documenting their KB. However they include various dictionnaries which in CGKAT correspond to the concept type lattice and the lexicon. Conversely, we claim that the CG formalism enables to represent (and eventually execute) most knowledge acquisition models. However, (Lukose & al., 1995) lists some requirements that the CGs should fulfill in order to be more useful in KA.

8 Conclusion

We have described a KAT which integrates a structured document editor with hypertext capabilities to a CG workbench, in order to use and mix three kinds of document access/organization techniques: those relying on hypertext navigation, those using the document abstract structure, and those using semantic queries. The KB of CGs which is built during the knowledge acquisition process, is used as a semantic index on documents. Multiple kinds of views on the documents and on the knowledge can be built by the various users. Our tool could not be used as a large-scale hypertext or IR system, but seems to have adequate features for knowledge acquisition. We will further assess this with the "car accident analysis" expertise on which our example was based.

In future works, we will study how to use queries by projection for answering common explanation demands like e.g. "In which situation X is used ?" or "Why doing X ?". We will also try to compare the knowledge asserted by many expert and represented by many users.

9 Acknowledgements

The author thanks the members of the ACACIA team, and especially Dr Rose Dieng and Dr. Olivier Corby for their advices on the redaction of this article and the building of the tool.

10 References

Albert P. & Vogel C (1989). *KOD-STATION un environnement intégré pour le génie cognitif.* In "Génie logiciel et systèmes experts", No 19, June 1989.

Amann B., Christophides V. & Scholl M. (1993). *HyperPATH/O2: Integrating Hypermedia Systems with Object-Oriented Database Systems.* In Proc. of DEXA'93, 4th Int. Conf. on Database and EXpert systems Applications (Editors: Varik V., Lazansky J., Wagner R.R.), Prague, September 1993.

Anjewierden A. & Wielemaker J. (1992). *Shelley - computer-aided knowledge engineering*. In Knowledge Acquisition (1992) 4.

Baker M. (1992). *Analysing rhetorical relations in collaborative problem solving dialogues*. In Proc. of the NATO Workshop "Natural Dialogue and Interactive Student Modelling", Varenna (Italy), October 1992.

Beeri C. & Kornatsky Y. (1990). *A Logical Query Language for Hypertext Systems*. In Proc. of DEXA'90, International Conference on Database and Expert Systems Applications, Vienne, Autriche, August, 1990.

Bernstein M. (1990). *Hypertext and technical writing*. In Proc. DEXA'90, Int. Conf. on Databases and EXpert systems Applications, Vienn (Austria), August 1990.

Bürsner S. & Schmidt G. (1995). Building views on conceptual models for structuring domain knowledge. In Proc. of the ninth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'95), Gaines, B.R.

Eds, University of Calgary, Banff, Alberta, Canada, February 26-March 3, 1995.

Campbell B. & Goodman J.M. (1988). *HAM: A General Puropse Hypertext Abstract Machine*. In Communications of the ACM 31(7): 856 861, July 1988.

Carbonneill B. & Haemmerlé O. (1993). *Implementing a CG Platform for Question/Answer and DataBase capabilities*. Proceedings of the Second International Workshop on Peirce, Québec, August 1993.

Carbonneill B. & Haemmerlé O. (1994). *ROCK: un système de Question/Réponse fondé sur le formalisme des Graphes Conceptuels*. In Actes du 9ème Congrès Reconnaissance des Formes et Intelligence Artificielle, Paris, Janvier 1994.

Cawsey A. (1991). *Generating Interactive Explanations*. In Proc. of the AAAI'91 Workshop on Comparative Analysis of Explanation Planning Architectures, Anaheim, California, July 1991.

Chen J.C., Ekberg T.W. & Thompson C. (1990). *Querying an Object-Oriented Hypermedia System*. In "Hypertext: State of the Art" (Editors: R. Mc. Aleese and C. Catherine), Billing & Songs, 1990.

Consens M.P. & Mendelzon A.O. (1990). *Low complexity aggregation in Graphlog and Datalog*. In International Conference on Database, 1990.

Croft W.B. (1987). *Approaches to intelligent informationretrieval*. In Information Processing and Management, Vol 23, no. 4, 1987.

De Young L. (1990). *Linking Considered Harmful*. In Proc. of the European Conference on Hypertext (ECHT'90), (Editors: Rizk A., Streitz N. & André J.), INRIA, Versailles, France, November 1990.

Gallagher L., Furuta R. & Stotts P.D. (1990). Increasing the Power of Hypertext Search with Relational Queries. In Hypermedia , 2(1):1-14, 1990.

Goldfarb C.F. (1990). *The SGML Handbook*. Clarendon Press, Oxford, 1990.

Haemmerlé O. (1995). *CoGITo: une plate-forme de développement de logiciels sur les graphes conceptuels*. Ph.D thesis, Université Montpellier II, France, January 1995.

Halasz F.G. & Schwartz M. (1990). *The Dexter Hypertext Reference Model*. In Proc. of the NIST Hypertext Standardization Workshop, Gaithersburg, National Institute of Standards and Technology, January 1990.

Hofmann M, Schreiweis U & Langendörfer H. (1990). *An Integrated Approach of Knowledge Acquisition by the Hypertext System CONCORDE*. In Proc. of the European Conference on Hypertext (ECHT'90), (Editors: Rizk A., Streitz N. & André J.), INRIA, Versailles, France, November 1990.

Kheirbek A. & Chiaramella Y. (1995). Integrating Hypermedia and Information Retrieval with Conceptual Graphs, HIM'95, Konstanz, Germany, April, 1995.

Kuntzmann-Combelles A. and Vogel C. (1988). KOD: a support environnement for cognitive acquisition and management. In Safety and Reliability Symposium, 1998.

Lemaire F. (1995). *Knowledge bases, texts and lexicon*. In Proc. of KBKS'95, Enschede, April 1995.

Liddy D.E. & Sung H.M. (1992). *DR_LINK Document Retrieval using Linguistic Knowledge, Project Description*. In Revue ACM: SIGIR Forum Vol 26 no 2, Fall 1992.

Lucarella D. (1990). *A Model for Hypertext-Based Information Retrieval*. In Proc. of the European Conference on Hypertext (ECHT'90), (Editors: Rizk A., Streitz N. & André J.), INRIA, Versailles, France, November 1990.

Lukose D., Mineau G., Kremer, Moeller J., Mugnier M, .Martin Ph. (1995). *Conceptual Structures for Knowledge Modelling*. Supplementary proceedings of ICCS'95, 3rd International Conference on Conceptual Structures, University of California, Santa Cruz, August 14-18, 1995.

Mann W. & Thompson S. (1988). *Rhetorical Structure Theory : Toward a functional theory of text organization*. In Text, 8, 3, 243-281.

Martin Ph. (1995). *Ontology Building Using a Semantic Word Database.* Submitted to ICCS'95, 3rd International Conference on Conceptual Structures, University of California, Santa Cruz, August 14-18, 1995.

Marshall & al. (1991). *Aquanet, a hypertext tool to hold your knowledge in place*. In Proc. of the 3rd ACM Conf. HTX'91, ACM Press, San Antonio (Tx), 1991.

Myaeng S.H. (1992). *Conceptual Graphs as a Framework for Text Retrieval*. Conceptual Structures: current research and practice (Editors: Nagle T.E., Nagle J.A., Gerholz L.L., and Eklund P.W.), England, Ellis Horwood Workshops, 1992.

Nanard J. & Nanard M. (1993). *Should anchors be typed too ? An experiment with MacWeb*. Proc. HTX93, 5th ACM Conf. on Hypertext, ACM Press, Seattle, Nov. 1993.

Nanard J., Nanard M., Massotte A-M., Djemaa A. Joubert A., Betaille H. & Chauché J. (1993). *Integrating Knowledge-base Hypertext and Database for Task-oriented Access to Documents*. In Proc. of DEXA'93, 4th Int. Conf. on Database and EXpert systems Applications (Editors: Varik V., Lazansky J., Wagner R.R.), Prague, September 1993.

Nogier J.F. (1991). *Génération automatique de langage et de graphes conceptuels*. Hermès, Paris, 1991.

Quint V. & Vatton I. (1992). Hypertext Aspects of the Grif Structured Editor: Design and Applications, num. R.R. 1734, INRIA, Rocquencourt, July 1992.

Quint V. (translated by E. Munson) (1992). The languages of Grif, IMAG, 2 rue de Vignate, 38610 Gières - France, 1994.

Richy H. (1994). A hypertext electronic index based on the Grif structured document editor. In Proc. of Electronic Publishing -- Origination, Dissemination and Design, vol. 7, num. 1, pp. 21-34, March 1994.

Schuler W. & Smith J.B. (1990). *Author's Argumentation Assistant (AAA): A Hypertext-Based Authoring Tool for Argumentative Texts*. In Proc. of the European Conference on Hypertext (ECHT'90), (Editors: Rizk A., Streitz N. & André J.), INRIA, Versailles, France, November 1990.

Sowa J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.

Sowa J.F. (1990). *Towards the expressive power of natural language.* In Proc. of 5th Annual Workshop on Conceptual Structures, sponsored by AAAI in conjunction with AAAI-90, Boston, MA.

Sowa J.F. (1992). *Conceptual Graphs Summary*. Conceptual Structures: current research and practice (Editors: Nagle, T.E., Nagle, J.A., Gerholz, L.L., and Eklund, P.W.), England , Ellis Horwood Workshops, 1992.

Streitz N. & al. (1992). *Sepia: a Collaborative Hypertext Writing System.* In Proc. ECHT'92, 4th ACM Conf. on Hypertext, ACM Press, Milano, Dec. 1992.

Toulmin S. (1958). *The Uses of Argument*. Cambridge University Press, 1958/

Wielinga B., Schreiber G. & Breuker J. (1992). *KADS: a modelling approach to knowledge engineering*. In Knowledge Acquisition (1992) 4.

Wuwongse V. & Manzano M. (1993). Fuzzy conceptual graphs. In Proc. of ICCS'93, Quebec City, Canada, August 1993. Lecture Notes in Artificial Intelligence, 699.