A Visual User Interface for a Personal Information Base using a Concept Network

Hiromichi Fujisawa*, Itsuko Kiuchi*, Takuo Koguchi** and Hidefumi Kondo**

*Central Research Laboratory, Hitachi, Ltd.
**Systems Development Laboratory, Hitachi, Ltd.

Abstract: A "personal information base" software tool, Concept Browser, is presented, which facilitates easy storage and retrieval of conceptual information and associated media information. Fragmental information at user's hands can be stored as organized, structured knowledge in terms of a network of concepts, and stored information can be retrieved easily even from vague, partial memory. A knowledge manipulation software tool, Concept-Network Editor, and a visual user interface have been developed. The visual user interface features a versatile view to a concept network and a flexible query edition capability which enables the user to create an explanatory query expression even from user's vague memory. The system makes inference to retrieve information from such explanatory query expressions. The system can be applied to intelligent document filing systems where documents are stored in terms of images, and related information is stored in terms of a concept network.

1. Introduction

Being in an information age, amount of information individuals should manage daily is increasing drastically, and it is becoming more important to keep such information in order and ready to use. One of the most important tasks that accomplish it is "document filing" in offices. Emerging electronic document filing systems using optical disks are gradually improving the situation [1].

Although such electronic filing systems can store tens of thousands of documents and can retrieve documents in a few seconds, they have not yet lived up to the expectations described above. The concept of "document filing systems" should be changed. This paper describes a system concept of a personal information base which can store and manage conceptual information in addition to documents, and a prototype system called Concept Browser.

The first problem here is in storing documents to such a system. It is difficult to find proper keywords to assign to documents being stored, and to make proper classification, such that they will be easily retrieved afterwards. The second type of problem is in retrieval. In general, making an access to information is not straightforward primarily because of our memory being vague and partial to state what we want very clearly. This type of problem is experienced even in daily document filing activities, especially when we ask somebody else to find a file.

Another problem is an incapability of storing conceptual information, which we think is one of the most important problems. Since people are prone to forget details, we propose a system that stores facts and other knowledge semantically for their sake. Conventional databases are appropriate for a pre-designed, limited range of items. For instance, a relational database can be used for a huge amount of records for rather uniform data structures, but it is not capable of handling conceptual, semantic information. Such information is carried by documents, but they are just stored in terms of document images and associated keywords.

Ideally, we need a system that can semantically associate a wide variety of piecewise information each other, storing it as a network of minimal information units in an organized manner. In a way, it is a kind of hyper-text with a knowledge base. In fact, it is our aim that this kind of system may solve the first two problems at the same time.

An approach presented here is to apply knowledge representation and reasoning techniques to store and retrieve such piecewise conceptual information, and to implement a user-friendly graphical interface on top of the knowledge representation system.

The application of knowledge representation techniques to information retrieval systems has been proposed and a few prototypes have been developed so far [2-6]. Importance of good graphical user interfaces for complex information structures is also well recognized [7-9]. Cognitive science approaches are also studied [10].
A visual user interface of the Concept Browser is implemented on top of a previously developed knowledge base system [4,6,11]. It has features that come from considerations on the human memory characteristics, enabling users to formulate query expressions interactively. They can be composed of concepts and relations selected from a network of concepts shown on the screen. The system stimulates the user to recall more concrete information and reflect it to the query expression. The expression can be at any level of abstractness. The remaining gap between an abstract query and concrete information to be retrieved can be bridged by making inference.

2. Problems and Approaches

The recognized problems in the current document filing systems are the following:

1) classification and registration,
2) retrieval, and
3) storing conceptual information.

In the following sections, these problems are discussed. Approaches presented here are the features of the Concept Browser.

2.1 Intelligent thesaurus assisting classification

Making proper classification and finding proper keywords for documents to be stored are difficult. When a filing system is based on classification, the user needs to guess what the classification labels mean. The user needs to select and open "cabinets" and "binders" to see which is the right place to store. It should be repeated until a correct classification is found.

If the system is based on keywords, the user needs to find pertinent words that may have a high recall probability. Even though thoughtful users select abstract words with broader meanings, terms will form an uncontrolled collection of indexes.

A solution for this problem is an Intelligent thesaurus that has a taxonomical hierarchy of those terms. Words having the same meaning or the same concept can form a concept node, which should also hold spelling variants. Homonyms should form separate nodes. These concept nodes structure a concept tree. The system should have a capability of editing the thesaurus to enter new terms and to restructure the hierarchy at any time. When a user does not find a word in the thesaurus, s/he should be able to add it as a new concept node, a synonym, or a spelling variant at a proper place in a hierarchy. It is also important that the node may have multiple parents.

Furthermore, a system using an intelligent thesaurus should have a browsing capability in addition to an editing capability. It can assist both indexing and retrieval.

Other issues in this area include automatic index extraction from document images, which is discussed elsewhere [5,12,13].

2.2 Explanation-based retrieval using a visual user interface

Due to user’s vague memory, only partial, abstract descriptions can be stated. An expression like "I want the document prepared by someone who attended the important meeting a few days ago" is a typical example. Although each word does not have any concrete information, this explanation-like query expression has a significance, if the system has knowledge about things and events, and if it can identify items to be searched.

When memory is too vague, the user needs stimulation. If relevant information is presented by the system, it is easy to recognize and it is expected to stimulate the user to remember more concrete memory. It is a recognition and recall characteristic known in psychology.

The manner of recalling memory is another important issue. Our recalling process is unorganized, or random. The query formulation process should be in accordance with this characteristic.

The visual user interface presented here therefore facilitates:

1) retrieval from explanation-based query expressions,
2) flexible editing of query expressions, and
3) browsing of the knowledge base.

Explanation-based query is different from keyword-based retrieval in that the query expression consists of concepts (nouns) and relations (verbs and attributes). For instance, a query expression such as "article whose subject is workstation which is developed by ABC Corp., which runs on UX operation system" is acceptable. Here, "is-developed-by" is a semantic relation between "workstation" and "ABC Corp." for instance. Relations include active relations expressed in verbs and static relations representing attributes. By using taxonomical reasoning [14], the system can identify concepts that are subsumed by an abstract concept expressed by a query semantically.

The intelligent thesaurus described in the previous section, therefore, can be extended to the notion of a knowledge base by incorporating semantic relationships between words or concepts.

The underlying ideas of 2) and 3) are parallel to "retrieval by reformulation" realized in RABBIT [3]. Flexible query edition allows to cut and paste concepts and relations in a query expression which are shown on a screen. By browsing through the concept tree, the user can pick a concept and paste it into a query expression. Addition and deletion of a condition is also facilitated.
A browsing capability is important to stimulate the user. The system should select a proper scope of concepts to display in a window. When a concept in the query expression is pointed, a subworld of concepts corresponding to the specified concept is automatically selected. A query editing process should cooperate with this browsing process.

2.3 Conceptual information store

As already discussed, conceptual information stored in the system can assist classification and retrieval of documents. However, the same information is actually that of user's interests. It is a natural desire to keep this kind of information in an organized manner.

In offices, information that should be managed by individuals is "heterogeneous" [2]. They have to remember things, objects, people, group of people, their attributes, companies, employees, company's organizational structure, customers, products, parts, technology, projects, plans, targets, budget, activities, tasks, events, happenings, public relations, regulations, schedule, telephone number, mailing address, documents, etc. in relation to other objects, and some generic knowledge.

Here comes the notion of "Personal Information Base" we propose, which can store such heterogeneous information in terms of concepts and relations, supplemented with other media. The resulting network of concepts is a store of this conceptual information. We would like to emphasize a knowledge base aspect of the intelligent filing system in [11].

3. System Architecture for the Personal Information Base

A personal information base system consists of files for a concept network and documents, a concept network manager to edit and retrieve concepts, a document retrieval system, and a visual user interface, as shown in Fig. 1.

Incoming documents are scanned and stored into a multimedia file on optical disks. Document images may be decomposed into their elements [12-13]. The decomposed elements include pictures, diagrams, and text, where text can be converted into code information by character recognition. Document understanding is a function block to extract knowledge directly from documents, which is a future system image.

To register a new document, the user creates a new concept node that represents the document, and then defines the document node by linking it to other concepts with some semantic relations, by interacting with the Concept Browser. For instance, subjects, authors, important facts described in the document, are registered into the concept network. The document node is automatically given a document image identifier for the multimedia file which is used to retrieve images.

When concept nodes to be linked are not there, the user can create and define them. Required changes in the classification hierarchy can be also accomplished by moving nodes in the hierarchy. By continuing this process through the daily usage, the concept network is adjusted to a user's world.

To retrieve information, the Concept Browser is used again. This process is described in detail in Section 6. When document nodes are retrieved and identified, corresponding images are shown on a separate high resolution screen from optical disks. The document browser supports a user-friendly page flipping function, whose feature is a very rapid, back-and-forth page turning operation. Three pages per second is possible [11].

4. Knowledge Representation by Concepts and Relations

A concept-relation model used for knowledge representation is a kind of semantic network. To allow easy knowledge manipulation, the model is kept...
simple and understandable [4]. A well-known semantic network knowledge representation language, KL-ONE [15] is powerful but complex. Here, bidirectional relationships among concepts are focused. Based on this model, knowledge is represented in terms of concept nodes and relation links as shown in Fig. 2. Nodes under the top node of UNIVERSAL represent physical objects, place, time, events, abstract concepts, properties, etc., and each may have a number of character strings assigned as a name. Names include synonyms and spelling variants.

Nodes constitutes a taxonomical tree which may be tangled, and higher level nodes are generic, representing a class of objects, while the ones at leaves are the most specific, individual concepts. The links to construct the taxonomical tree are superclass relations, also called Is-A relations. A node can have more than one superclass; therefore, the hierarchical tree is actually a tangled tree.

Relations other than superclass relations represent arbitrary, semantic relations, which can be created by the user. The relationships include properties of things as well.

A part-whole relationship is special such that semantic relations can be declared to be a part-whole (par of) relation as well. For instance, "inclusion" relations for organizations (Fig. 2) and geographical entities, etc. are part-whole relations at the same time. In taxonomical reasoning, superclass relations and part-whole relations are specially treated.

There is a distinction between generic and instance in the semantic relations. A generic relation represents a possibility of relationships among the subsumed concepts, defining a semantic data model. For instance, the link between MANUFACTURER and PRODUCT in Fig. 2 shows that a manufacturer may have a "production relationship" with a product. The bidirectional relationship is given two print-names, e.g. "produces" and "is produced by," and the relationship can be displayed as both "Manufacturer produces product" and "Product is produced by manufacturer."

Instance relations represent concrete facts. In Fig. 2, a fact "Hitachi produces Creative WS 2050" is represented. If an instance relation is defined at an intermediate level, the relationship is inherited by the lower concepts, where multiple inheritance is supported. Individual concepts and instance relations correspond to data or records in an ordinary database.

A sub-network of generic concepts and generic relations is "kernel" in the sense that it corresponds to a semantic data model and it serves as a container of the conceptual information. It can be supplemented by the user during the course of use.

5. Concept Network Editor

The concept network editor is a software tool to enable the user to:

1) create generic/individual concepts and generic/instance relations,
2) restructure the taxonomy by moving concepts and adding/deleting superclass relations,
3) add and edit aliases of a concept,
4) edit generic relationships,
5) remove concepts and relations,
6) browse through the network, and
7) search concepts.
A versatile view onto the concept network is one of the features. Five kinds of virtual views are prepared: i.e. a taxonomic hierarchical tree, partonomic hierarchical tree, concept frame, table, and natural language representation. The taxonomical tree is constructed by superclass relations, while the partonomical tree is constructed by part-whole relations. The concept frame is a view to show all properties or relations defined to a single concept, while a table is a list of individual concepts with their relations and properties. Although it is a limited capability, facts can be shown in a natural language style.

These multiple views are all virtual and dynamically generated from a single data structure based on the concept-relation model. Therefore, local updates are reflected to all views.

A new concept node can be created at a point in the taxonomy with multiple names. The user specifies whether it is generic or individual. Attributes and relations of the concept are defined by binding other concepts by instance relations. To do so, the system identifies possible relations for this concept by looking for the generic relations attached to the superclass concepts of this concept. Then, to guide the user, the system displays the possible relations in a frame structure, called generic frame, showing which relations and properties can be defined. Each slot of the frame is a relation and the slot value is a most generic concept. The slot value specifies which concepts can be connected by that relation.

To create an instance relation, the user first selects one of the slots in the generic frame, and specifies a related concept from among the concepts subsumed by the generic concept shown as a slot value. The system shows a subworld automatically to which the browsing scope is limited, acting as "value restriction." By following these steps recursively, semantic information can be encoded.

The search functions of the editor are used for information retrieval as well. Together with flexible string matching for concept names, a "concept matching" capability is provided for semantic information retrieval from explanation-based query expressions.

A typical query example is shown in Fig. 3. It is more or less like an explanation. Restrictions to the concepts can be cascaded and the words are abstract. Image about an article encircled in Fig. 3 will be articulated as a query expression in a box, and the concept matching using taxonomical reasoning is applied. Retrieval results are displayed by substituting matched concrete concepts to corresponding ones in the query expression.

By using the editor, we have encoded more than 3500 concepts and 7000 relations into an experimental concept network. About 400 news articles were digested by an experimenter, and important concepts and related facts were selected and registered. A kernel network of about 600 generic concepts and 90 generic relationships were constructed as a result of a bottom-up knowledge acquisition process. The deepest depth of the hierarchy was 13, and the average branching factor of the nodes was about 15.

To provide this system for end users, a kernel concept network must be developed from which they can accumulate their own information without much modification in the kernel part. We expect that several thousand generic concepts and more than one thousand generic relationships need to be prepared for the kernel.
6. Visual User Interface

A visual user interface for the concept network editor has been developed to facilitate user-friendly information manipulation methods. By interacting with the interface, the user can update the contents of the knowledge base and formulate query expressions flexibly. In this paper, we focus on the query editing function whose approach was discussed in Section 2.2.

6.1 Window organization and general operations

As shown in Fig. 4, Concept Browser has following three main windows and some other auxiliary windows.

1) Concept tree window,
2) Query editing window, and
3) Superconcepts window.

Frames and tables are displayed in a separate window. One of the problems to be solved here is how to manage limited space for display, since most of the time contents to be displayed is too large for the available space.

In the concept tree window, a subtree is displayed, whose location in the whole concept tree is specified in the superconcepts window. Each node of the subtree being displayed has attributes of a maximum width and a maximum depth of branches for display. Such maximum values can be changed by the user dynamically. Therefore, only a selected part of the tree is displayed. The user can also add and delete branches on the display at any time. When a selected subtree is too large for the window, scroll bars appear and the tree can be scrolled.

A concept highlighted by a rectangular is called "current concept." The current concept is shown in both the concept tree window and the query editing window. It is a kind of pointer to which the user is paying attention.

The superconcepts window shows the hierarchy of the superconcepts of a current concept, and controls the scope of a subtree to be displayed in the concept tree window. The circle in the superconcepts window (Fig. 4) shows the scope being displayed, and the dot (in a circle in this case) shows the most generic concept that can become a "current concept." The concept with the circle is the top most concept in the concept tree window. By picking one of the superconcepts, the user can change the position of the circle.

To zoom into the fine detail of the concept tree, the user can pick a superconcept at a lower level and increase the numbers of maximum width and maximum depths for the branches. To zoom out, a higher superconcept and smaller numbers are specified.

A concept frame window may have a scroll bar when contents are too large. Furthermore, slots with the same relation name are suppressed by showing only the first one. The user may know how many relations are suppressed by the skipping slot numbers. (See a large frame window in Fig. 9) When hidden relations are to be shown, the user selects them from a pop-up menu, and the selected relations appear in the original concept frame.

Throughout the windows, character strings representing concepts and relations are mouse-sensitive, in trees, frames, tables, and sentences. Depending on where a selected concept is located, a list of possible operations is displayed as a pop-up menu.
In the concept tree window, possible operations for a selected concept are browsing operations and editing operations. The browsing operations include those for shifting "current concept," changing maximum width and depth, addition of subconcepts to the display, hiding concepts from the display, displaying a concept frame, and displaying a table. Editing operations include creation of a new node, movement of a concept in a hierarchy, addition/deletion of superclass relation links, update of aliases, and creation of instance relation links.

In the superconcepts window, the operation is to change the scope of the concept tree, as described above.

Operations for concepts in the query editing window will be described in the following section.

6.2 Scope of the concept tree window associated with the query expression

A query expression being edited is shown in the query editing window in a limited natural language style. The phrases consist of nouns (concepts) and relative pronouns followed by a relational phrase. It is one of the virtual views to an internal representation. Character strings showing concepts and relations in the expression are mouse-sensible. By clicking a mouse on a corresponding character string, the user can invoke a possible operation.

When a concept is clicked in the query editing window, it becomes a current concept and highlighted. Then the contents of the concept tree window is changed such that the subtree contains the new current concept. The subtree is automatically determined by constraints posed on the current concept.

Take an example shown in Fig. 5 where the selected concept is "LISP MACHINE." It is constrained by two relationships in the query expression. They are "is-developed-at" and "runs-on" relationships in this case. Concepts that can replace "LISP MACHINE" should be also constrained by the same relationships. This means that they should be concepts that are subsumed by both "COMPUTER" and "ARTIFACT." To determine those concepts, the system first identifies generic relation links for those constraining relations and then selects one that is nearest to LISP MACHINE node. It is "runs-on" generic relation in this case which is connected to COMPUTER node. Therefore, the subworld to be displayed is concepts subsumed by COMPUTER, which is shown by a shaded area in Fig. 5.

Then the user may pick any concepts below COMPUTER displayed in the concept tree window to replace the current concept in the query expression.

If a memory is vague, s/he may select more generic concept in the displayed subtree; and if it is exact, s/he may select a more specific concept on the other hand. In this way, the level of ambiguity for the retrieval can be reflected to the query expression, and the formulated query expression is semantically valid all the time.

6.3 Query edition process

Query editing functions are:

1) replacement of a concept,
2) replacement of a relation,
3) addition/deletion of a restriction, and
4) addition/deletion of a root concept.

Starting with a single concept, the user can "assemble"
a query expression. As in Fig. 6 (a), a concept X can be constrained by a restriction R1-Y where R1 is a relation between X and Y. It is also possible to add a root concept Z to concept X through relation R2 as in Fig. 6 (b), enabling creating a query in a reverse order. In this way, the user can edit query expressions flexibly, according to a user's random recall process.

If the expression is too abstract resulting in many items retrieved, it can be changed to a more specific one by replacing a generic concept in the expression to a more specific one, or by adding more restrictions. If it is too narrow on the other hand, it can be broadened by opposite operations. It is also possible to change a relation when a retrieval result is not satisfactory; for instance, relation "produces" can be changed to "has developed."

By using these functions, in the case of Fig. 5, concepts "COMPUTER SOFTWARE" and "LISP MACHINE" are changed to concepts "EXPERT SHELL" and "WS", respectively; relation "is-developed-at" is changed to relation "is-produced-by". And a root concept "ARTICLE" is inserted afterwards.

Figures 7 and 8 show the editing process on the visual interface more concretely.

In Fig. 7, the query expression is "EXPERT SHELL which runs on WORKSTATION" where
WORKSTATION is a "current concept" and is going to be constrained by a relation. The system is showing the generic concept frame for WORKSTATION which has slots of:

1) "is subject of DOCUMENT"
2) "runs under OPERATING SYSTEM"
3) "is produced by ORGANIZATION"
4) "is developed at ORGANIZATION"
5) "runs PROGRAMMING LANGUAGE"
6) "has interface of INTERFACE-DEVICE"
7) "runs COMPUTER SOFTWARE".

Assuming relational phrase "is developed at some company" is being added, the slot No. 4 is selected and highlighted.

Then the system shows a subworld of ORGANIZATION in the concept tree window, corresponding to the highlighted slot, as shown in Fig. 8. At the same time, the query expression has a new line of "which is developed at ORGANIZATION" added, and the current concept is changed to "ORGANIZATION". Then, the user may select a more specific concept for the current concept.
This process can go on, resulting in a query expression such as:

"ARTICLE
whose subject is EXPERT SHELL
which runs on WORKSTATION
which is developed at COMPANY"

as shown in Fig. 4. In this example, the result of retrieval is shown as:

"ARTICLE#1010
whose subject is ES/KERNEL
which runs on Creative WS 2050
which is developed at Hitachi, ltd."

If any concept in the retrieval result window (the shaded concept in Fig. 4 for instance) is picked, the concept frame for the picked concept is displayed showing the attributes and relations.

More generally, in the Concept Browser, concepts shown in the concept tree window, retrieval result window, frames, and tables, can be asked for more information. When they are picked, relevant information is shown in the frame format. It can be done repeatedly. This is a powerful feature, supporting associative browsing as shown in Fig. 9.

7. Conclusions

An architecture for a personal information base and its visual user interface were presented, aiming at the improvement of the personal information environment in offices. The Concept Browser, a prototype system, can manage pieces of conceptual information in terms of a concept network supplemented with document images. The kernel of the network defines a "container" of concrete information.

The user can interact with the visual interface implemented on the concept network editor to update and retrieve conceptual information. Explanation-based query expressions can be formulated by cut-and-paste operations while browsing through the contents of the knowledge base. Concrete information from such queries is retrieved by using taxonomical reasoning.

An experience with this experimental system has suggested effectiveness of this approach, especially for heterogeneous information, through which a concept network with the size of more than 3500 concepts and 7000 relations has been constructed. Although the size of the experimental concept network was not as big as that for a real use, the fact that this size could be realized means its practicality and effectiveness.

Still, a more systematic study is required to evaluate the effectiveness through test runs by a wide variety of people. The size of a kernel concept network required for the real use will be ten times as big as that of the experimental network we have constructed.

8. References