A Personal Universal Filing System Based on the Concept-Relation Model.

Conference Paper · January 1986

Source: DBLP

3 authors, including:

Hiromichi Fujisawa
Waseda University
82 PUBLICATIONS **3,023** CITATIONS

SEE PROFILE

A Personal Universal Filing System Based on the Concept-Relation Model

Hiromichi Fujisawa, Atsushi Hatakeyama
Jun'ichi Higashino

Central Research Laboratory,
Hitachi, Ltd.
Kokubunji, Tokyo 185

Abstract

A personal universal filing system, UNIFILE, based on a knowledge-base approach has been developed. The system supports to accumulate small pieces of information to construct organized knowledge and provides many kinds of view on the stored information to help the whole process of "filing." As the result of the analysis of "filing," acquisition of knowledge rather than retrieval and a capability of reflecting the user's view on the world are focused. A concept-relation model is adopted as a knowledge representation scheme to simplify the data model that helps the user store heterogeneous information without difficulty. A versatile view of hierarchical trees, frames and tables is provided on the data structure. Among these, the tabular-form view with semantic selection by concept matching is unique and very effective. To evaluate the effectiveness of the approach, computer related articles have been collected and the extracted information has been coded into UNIFILE. The results obtained have clarified that it is preferable to enter more information into this system rather than store it in terms of paper documents.

1. Introduction

The increasing amount and value of information today gives rise to the need for more advanced filing technology. The emergence of large storage devices such as optical disks is encouraging the development of a document filing system which stores documents as images. This kind of system can store more than 60,000 pages of documents per optical disk[15]. To meet these needs, however, merely a large capacity is not enough; functions that support the whole filing process should be cultivated.

In this paper, we analyzed "filing" from three aspects. First, by consulting a dictionary for the definition of the word "file," we looked for the essence of filing. As a result, we decided to emphasize the importance of storing and constructing organized knowledge rather than the retrieval of information as in [6,7,9,11,16,18]. For this reason, the word "filing" is used instead of "information retrieval."

Secondly, we looked into the kind of information involved in filing activities and classified it into six classes. Then, we grouped the requirements of advanced filing systems into three clusters by analysing the needs. The cluster that should be cultivated is identified as a personal universal file that helps users accumulate a piece of information from a wide range of sources and analyze it from many viewpoints.

Expert Database Systems: Larry Kerschberg, Editor.

To realize such a personal filing system, we propose a concept-relation model as a knowledge representation scheme to unite fragmental information and document images. The representation scheme is concept- (object-) oriented as in KL-ONE[20], but it provides a more simplified view of the data structure. We focus on relationships among concepts in hand rather than a complete definition of a new concept. Concepts and relationships construct a concept network. The user can express his own world-view in terms of the concept taxonomy and generic relationships.

A knowledge representation system, called the Concept Network Editor, has been developed. It supports such functions as showing many different views of stored data, browsing through the network, constructing the network, semantic retrieval by concept matching, and tabular-form retrieval like QBE[3]. Among these functions, the tabular-form retrieval with a concept matching capability is unique. The user may select any generic concept as a table and any set of relationships as a column set, and then he may set a semantic condition onto any column.

To investigate the effectiveness of this approach, we have developed a prototype of a personal universal file, UNIFILE, and carried out an experiment on the storing of information gathered from computer related articles. As a result of storing about 70 articles, more than 1000 concepts have been captured, among which more than 2000 relations have been defined. The experience has shown that the capability of browsing through the network and the capability of showing many different views encourage users to collect more information, making the proposed approach a very promising one.

2. Meaning of "Filing"

2.1 What is Filing?

In a dictionary, "file" is defined as follows[1]:

> file: 1) to put or keep, paper or cards for example, in *useful order*;
> 2) a receptacle that keeps *loose* objects or *small* objects in useful order.

We can see the importance of the act of storing rather than retrieving information and that the act of storing involves the binding together of small pieces of things (or information) which would otherwise be kept separately. This view conforms with our daily experience that information comes fragmentarily at a random order and we always worry where to store it.

The useful order will be realized by connecting pieces of information with each other through meaningful relationships and accumulating information as a knowledge base, from which any part of knowledge may be retrieved and shown in a flexible manner.

This view on filing is different from the conventional view of information retrieval systems in that the user himself gathers and stores information, rather than someone else who serves for unspecified people. Therefore, the user's world-view should be reflected in the system. The second difference is that information to be stored comes from many sources and a wide range of information should be stored in a uniform data structure; thus, it must be a heterogeneous knowledge base[11,18].

2.2 Six Classes of Information

To make clear the problems in the present filing system, it is important to look into the kinds of information involved. Conventionally, it has been classified into two classes; original documents (primary information) and bibliographic data (secondary information). We think it is not enough to discuss the problems so that we have classified information into the following six categories:

1) **Original documents:** Original information should be stored electronically as images so that the loss of information is minimized. Since it is an important concept to "freeze" a document at a certain time, storing documents as images matches the concept. Presently, the emergence of large optical storage devices inspires to develop electronic document filing systems that store document page images. Representational

schemes and data compression methods for the multi-media data including text, figures and color pictures are to be studied.

2) **Bibliographic data:** Description of the source of information. This class of information is objective and main issue is the input cost. Automatic recognition of such data from the document image will be required for a large database system to automate the input process. However, there are such cases in the daily filing activities that the title of a document has less meaning; in these cases, the following fourth class of information is more important.

3) **Abstract of contents:** Description of contents where key words and abstract are included. This class of information is sometimes subject to an indexing person's view. Originally, it should be a subjective description by the author. These second and third kinds of information correspond to the secondary information based on the conventional classification. Here, it is separated into two classes as described above because the former class is objective while the latter is rather subjective. There have been many researches to extract objective index terms from original texts automatically[22,23,24,25].

4) **Value of information:** What is understood by the recipient, its value to him, personal comments, and relationships to other objects or concepts are included here. For instance, a document may have a strong connection with an event, like "it was submitted at the meeting Mr. A attended." It is due to the importance of this category that the filing activity is different from conventional information retrieval.

5) **Knowledge about a specific domain:** This is the kind of information that is acquired from the original documents. It is the result of the recipient's digestion. Some questions may be answered by consulting this class of knowledge without retrieving the original document. At the same time, it helps the user store fragmental information and retrieve it from a semantic description as described in the following sections. These fourth and fifth categories will be the main theme of this paper.

6) **General knowledge:** This is the knowledge not specific to a domain. Though the last three kinds of information differ respectively in the meaning, the same knowledge representational scheme can be applied uniformly.

The above classification clarifies the necessary steps toward the advanced filing technologies. For example, while the conventional bibliographic database holds only the second and third classes of information, original document database systems being investigated[5,8] are the ones that store also the first kind of information, i.e. original documents as images. However, we still do not have a filing system that can handle the fourth and fifth classes of information. Ironically, the results of bibliographic database searches create a bundle of paper and we worry where to put it for the future use. Therefore, we can conceive a new system for personal use that integrates all these classes of information.

2.3 Approaches to Advanced Filing Technology

We recognize the following three categories of requirements of advanced filing systems:

1) **Large systems for information services:** On-line remote-accessible libraries and patent information services are conceived and being developed[5,8,17]. As tens of millions of documents are to be stored as images, pattern recognition technology should be applied to extract and recognize bibliographic data on the documents automatically[21]. After each document has been identified by pattern recognition, conventional information retrieval methodology can be used. This approach is also investigated by the author's group which will be presented elsewhere.

2) **Distributed office systems:** Information flow is more important than storage. This kind of system should support rather regular business activities. Technical issues are application of the relational database management system, local area networks that can support document image transfers, and management of distributed database systems. Integration of the business knowledge will be important. A typical example of this kind may be patent office systems[17] where the patent applications are examined by consulting the disclosed documents and the examination procedure has been well developed and well defined in spite of the complexity.

3) Personal filing systems: As discussed above, we need a system that can accumulate and unite fragmental information to construct organized knowledge as a result of the user's digestion of incoming information. It should integrate six kinds of information; the fourth and fifth classes are particularly important for "filing." It is noted that this kind of system is necessary even in a large system since those who interact with the large systems are individuals.

In this paper, we propose a "Personal Universal File" that meets the requirements described above.

3. Concept-Relation Model for Knowledge Representation

To integrate the fourth and fifth classes of information, we apply AI approaches. The knowledge representation scheme proposed here is concept- (object-) oriented and based on a concept-relation model. It gives a simplified view of the network representation and a flexible access path to any concept. Both abstract and concrete objects (things) are treated as concepts, including attribute values. Each concept can be given more than one name (word) for reference and may have more than one superclass concept. The exception is the concept UNIVERSAL which has no superclass concept. Superclass is a special relation that forms an taxonomic hierarchy of concepts or equivalently a subsumption hierarchy.

A generic relationship may be defined between two generic concepts as G(m,i,j) which stands for the m-th generic relationship defined between the concepts C(i) and C(j), and which has the print names R-lr and R-rl. The relationship can be read as (C(i) R-lr C(j)) and (C(j) R-rl C(i)). For instance, the generic relationship AUTHORSHIP may be defined between the concepts PUBLISHED-MATERIAL and PERSON and a triad can be read as (PUBLISHED-MATERIAL IS-WRITTEN-BY PERSON) or (PERSON HAS-WRITTEN PUBLISHED-MATERIAL), where the print names R-lr and R-rl are IS-WRITTEN-BY and HAS-WRITTEN respectively and are assigned to AUTHORSHIP. The generic relationships define generic frames of concepts. In the above example, the generic relationship AUTHORSHIP defines both

```
(PUBLISHED-MATERIAL (IS-WRITTEN-BY PERSON)) and
(PERSON (HAS-WRITTEN PUBLISHED-MATERIAL))
```

where IS-WRITTEN-BY and HAS-WRITTEN are slot names and PERSON and PUBLISHED-MATERIAL are the slot fillers of the corresponding slot. In our representational scheme, frames are a kind of view that is virtually generated.

Similarly, an instance relation may be defined between two concepts C(k) and C(l) where C(k) < C(i), C(l) < C(j), and the symbol < is read as "is subsumed by." The relation is denoted as r(n,m,k,l) which stands for the n-th relation belonging to the m-th relationship defined between C(k) and C(l). An example is (BOOK#0031 IS-WRITTEN-BY ISAAC.NEWTON), where (BOOK#0031 IS-A BOOK), (BOOK IS-A PUBLISHED-MATERIAL) and (ISAAC.NEWTON IS-A PERSON). Instance relations define instance frames virtually. In the above exapmple, defined instance frames are

```
(BOOK#0031  (IS-A BOOK)
            (IS-WRITTEN-BY ISAAC.NEWTON)) and
(ISAAC.NEWTON  (IS-A PERSON)
               (HAS-WRITTEN BOOK#0031)).
```

As can be seen, a fact represented by relations can be entered in either way, i.e. from either BOOK#0031 or ISAAC.NEWTON. This is a very important feature because a piece of information comes at random.

The basic data structure is the four relational tables of concept name C, superclass relation s, generic relation name G, and instance relation r, as shown in Fig. 1. As shown, concepts, relations and generic relationships are given unique system names like C#1065, R#1070 and RS#0051 and are used in inference. An entity-relation diagram[14] for this data model is shown in Fig. 3 where a rectangular block and a diamond-shaped block represent

an "entity" table and a "relation" table respectively. Among these four tables, only C and G are language dependent and can be easily extented to a multiple-language system.

Another representational diagram is a concept network as shown in Fig. 2, where it is a part of the network representing the fact "there is an article, ARTICLE#0014, whose subject is a superminicomputer, HP-9000, which is developed at a company called HP, which is located in Palo Alto,..." There are three kinds of arrows representing superclass relations, generic relations and instance relations, all of which are defined between one or two concepts shown by the ellipses. The concept network is another virtual view.

As every concept is treated equally and a relationship can be defined in either direction, new concepts and relationships can be registered at random, and any concept can be retrieved through the same process. For instance, "which company has developed the computer?" and "which computer has been developed by the company?" are equally well answered. A uniform mechanism is important to support both the input and flexible queries of heterogeneous information .

4. Concept Network Editor as a Knowledge Representation System

The concept network editor as a knowledge base editor is a user interface of the personal universal file. It provides the functions of browsing, registration and retrieval in addition to the editing functions. The editing functions include

1) creation and change in concept names,
2) change in taxonomical position of concepts,
3) removal of concepts,
4) creation and change in generic relationships, and
5) creation and deletion of instance relations.

By adopting the concept-relation scheme, only local changes in the tables shown in Fig. 1 are required, thus simplifying the update procedure. The features of the concept network editor that is important in filing will be described in the following subsections.

4.1 Versatile View

Four kinds of view can be shown at a concept node as shown in Fig. 4; a) subtree of the taxonomic hierarchy, b) subtree of the part-whole hierarchy, c) generic and instance frames, and d) tables. As for the part-whole hierarchy, there are as many kinds of subtrees as the number of part-whole relationships; these include, for example, spatial inclusion, subsystem, machinery parts, suborganization, science-technology subfields, etc. Other hierarchical tree will be also possible like successor relationship, obedience relationship, etc. The essence here is that the tree is constructed virtually according to a single relationship. Drawbacks and limitations of the conventional hierarchical file systems stem from usage of the mixed relationships to construct a tree.

Tabular representation is a familiar view for the general user and very effective in the concept network model (Fig. 4(d)). Instance frames of individual concepts that are subsumed by a generic concept are collected and only the slots, or relationships, that are specified by the user are extracted. Then, the set of reduced frames is converted to a tabular form. The table display routine accepts any number of columns with any length of strings. It allocates each field length automatically and folds strings that are longer than the determined length. Query functions on the tabular form will be described later.

4.2 Browsing Through the Network

With the browsing capability, a user can move around in the network by changing the current node of concept in the editor. There are four types of network traverse:

1) shifting along the taxonomic hierarchy,
2) shifting along the part-whole hierarchy,

3) shifting by specifying a substring for a concept name, and
4) shifting by selecting a slot in an instance frame.

Substring matching is used to make global searches, where matching can be limited to subconcepts that are subsumed by the current concept. For instance, at the PERSON node, matching only applies to men and women. When a substring matches more than one concept, the user may choose one from the menu as shown in Fig. 5 where showing superclass to each matched concept name is very effective to select a proper one.

Shifting to a concept associated with the current concept through an instance frame is a kind of association retrieval supporting another global search. It is noted here that the user can go back and forth through relation links, since the links can be traced from both sides as described in section 3. Semantic query described later can be also used to determine the place to which to shift. It should be noted here that interweaving usage of four kinds of browsing capabilities is very effective and the introduction of a multi-window capability will make it much more effective. The versatile view described above is the key for the browsing.

4.3 Creation of a New Concept

A new concept is defined at some point in the taxonomic hierarchy, as shown in Fig. 4(a), wherein the point is selected by browsing. Then, instance relations for the new concept are to be entered. Since relations are an instanciation of generic relationships, the system can determine which relations can be defined, by looking at generic relationships attached to superclass concepts through a property inheritance mechanism. The system then shows possible relationships and the corresponding superclass concepts as a generic frame as in Fig. 6. Here, the generic frame of concept C(i) is a frame representation of generic relationships defined to superclass concepts of C(i), i.e. $C(i) < C(i_1) < ... < C(i_n)$. The slots in the frame are filled with associated concepts that act as value restrictions.

Take the example where a user has a piece of information "Hitachi has developed the personal computer B-16." In this case, he first looks for "Hitachi" and may enter the fact as a relation if HITACHI is already registered; otherwise he can create the concept HITACHI under the concept COMPANY. When he locates HITACHI, the system identifies (HITACHI IS-A COMPANY) and it can show a generic frame for HITACHI that presumably includes the generic relationship (HITACHI HAS-DEVELOPED MACHINERY.DEVICE). After selecting the slot for this relationship, the system shifts the current node to MACHINERY.DEVICE and the user will browse in a sub-world of MACHINERY.DEVICE to determine the place for B-16. In this case, the user must have recognized that the personal computer is subsumed by the generic concept MACHINERY.DEVICE because generally there are more than one slot that have the same slot name. For example, there will also be (HAS-DEVELOPED SOFTWARE-SYSTEM). Determination of the place will complete the registration of the instance relation (HITACHI HAS-DEVELOPED B-16). During the process, he is forced to be in the sub-world so that it works as a value restriction.

Again, if he cannot find B-16, it can be newly created. If he thinks he needs a generic concept PERSONAL-COMPUTER in the process of searching for the place for B-16, he may add it under COMPUTER for example. This means that a new concept can be created at any time even during the process of defining relations. And if he has more information like "the B-16 runs under the MS-DOS," he may continue this process recursively starting with B-16. Likewise, generic relationships can also be defined any time he encounters a new kind of relationship. Browsing capability can be effectively used to specify two generic concepts.

For creation of a new concept with associated relations, the recursive call to the concept network editor and browsing capabilities with versatile view are indispensable.

4.4 Semantic Query

One of the features is semantic query formulation through a menu-driven dialog[19]. Query is semantic because the query formula preserves the semantic structure of the description of items; more specifically, concepts appearing in the formula are linked by relations. The dialog is guided by knowledge in terms of the generic relationships. Take the example where the user looks for "a computer which runs under Unix* and has been developed at a company in California" (* Unix is a trademark of AT&T Bell Laboratories).

He first changes the current concept node to COMPUTER in this case, and enters the MQ command to make a query frame. The system then responds by showing the generic frame of the concept COMPUTER (Fig. 6(b)). From among the slots, the user can select one and fill it with more specific concepts if he has more information to add. In the example, he selects the RUNS-UNDER slot and fills it with UNIX after the system automatically shifts the current concept to OPERATING-SYSTEM. Then, the internal query frame becomes (COMPUTER (RUNS-UNDER UNIX)). This process is quite similar to the above-described process of defining relations.

Now, since he knows it is developed by some company and the company is located in California, he continues the process by entering the MQ command recursively. The final query frame will be

```
(COMPUTER
     (RUNS-UNDER UNIX)
     (IS-DEVELOPED-BY (COMPANY (IS-LOCATED-IN CALIFORNIA)))).
```

The search for an item is done by matching concepts. Candidate concepts are leaves of the taxonomic subtree of a generic concept, COMPUTER in this case. Then, the query frame is matched with each instance frame of the candidate concepts.

The concept matching process proceeds as follows. First, it is determined whether or not every slot of the query frame appears in the instance frame. If so, the two concepts in those corresponding slots are matched and this process is applied recursively. Otherwise the candidate concept is rejected. For example, in the case of Fig. 2, an instance frame

```
(HP-9000  (RUNS-UNDER UNIX)
          (IS-DEVELOPED-AT HP))
```

is matched against the query frame shown above. In this case, each slot of the query frame has corresponding slot in the instance frame, and the two concepts UNIX and HP will be matched against UNIX and COMPANY, respectively.

The match between two slot values succeeds in the following four cases:

1) The two are identical.
2) The concept as a slot value in the query side subsumes the concept to be matched, like (IS-DEVELOPED-AT COMPANY) and (IS-DEVELOPED-AT HP).
3) The concept to be matched is a part of the concept in the query side; for instance, matching of (IS-LOCATED-IN CALIFORNIA) to (IS-LOCATED-IN PALO-ALTO) succeeds.
4) Both concepts are explicitly defined to be equivalent by EQUIVALENCE relations.

The key to success in the last three conditions is backward chaining inference tracing through the IS-A, PART-WHOLE and EQUIVALENCE relations, where the IS-A relation is the superclass relation.

Document retrieval can be done in quite a same manner. If the user looks for an article, he can shift the current concept to ARTICLE and then qualify the concept by using the MQ command recursively, getting, for instance,

```
(ARTICLE (SUBJECT-IS (COMPUTER (RUNS-UNDER UNIX)
                               (IS-DEVELOPED-AT (COMPANY (IS-LOCATED-IN CALIFORNIA))))))).
```

The result of a search will be ARTICLE#0014 in case of Fig. 2. The DSP command can show the original document as images associaed with the concept.

4.5 Tabular-form retrieval with the concept matching

The tabular-form view described in an earlier subsection has query functions. Conditions can be applied to columns and the result of the selecton is displayed in the same manner. The acceptable conditions are 1) arithmetic conditions like =, >, <, etc., 2) substring matching, 3) set inclusion test, and 4) concept matching.

Semantic selection using concept matching in a tabular-form retrieval is a unique feature of this system. Figure 7(a) as an example is a tabular representation for the individual concepts of COMPUTERs where their manufacturer, application software and relevant article are shown. To make a semantic selection, the user first specifies a column to which a concept matching should be applied, after which a semantic query frame can be created through a dialogue as described above. The result of the semantic selection with the query condition (COMPANY (IS-LOCATED-IN CALIFORNIA)) applied to the column IS-DEVELOPED-BY is shown in Fig. 7(b). In this case, backward chaining inferences through the spatial part-whole relationships are carried out, since each company's location is a CITY while the location specified in the query is happens to be a STATE.

The user may repeat the process with other conditions, where there is an option of how a new condition is added to the current one, i.e. OR or AND; and he can also go back to the previous conditions. As a result, he can analyze the collected data from many aspects.

5. An Experience with a Personal Universal File

An experimental system, called UNIFILE-I, has been constructed to determine the basic structure of a personal universal file. The system consists of the concept network editor on a host computer, a personal computer which acts as a terminal for the host computer and the optical document file system, Hitfile-60[15] which can store up to 60,000 document images per optical disk. The personal computer interacts with the concept network editor on a time-sharing host computer and at the same time controls the optical document file by transferring messages from TSS to the Hitfile-60. The Hitfile-60 has an image scanner, a laser printer and a high resolution CRT display as input/output devices, each of which has resolutions of both 8 dot/mm and 16 dot/mm except the CRT which has 8 dot/mm resolution. The system is currently being transferred to a lisp machine to exploit the multi-window capability.

As an extension of the concept network, images of any size can be assigned to any concept and can be displayed at the DSP command. Straightforward application is to assign document images to individual concepts of ARTICLE. Another extension made is the multiple language capability. Tables of concept name C and generic relation name G have been extended to include the column LANG, which stands for language. The column has entities J and E which represents Japanese and English respectively. In case of the Japanese mode, for example, records whose LANG field value is equal to J are searched for a specified concept number. If no record is found, an English version is used. Figure 8 shows a view in the Japanese mode where the frame representation is different from the English version because of the difference in the Japanese verb order.

To investige the effectiveness of this approach, we carried out an experiment in which news articles on computer related products were collected from technical magazines and digested information was coded into the knowledge base. This experiment was conducted in English using the first version of the system.

Initially, a basic concept taxonomy was constructed by the experimenter's world model with the help of a thesaurus[2], and certain generic relationships were defined. At that time, the approximate numbers of concepts, generic relationships and relations were 365, 25 and 95, respectively. These articles were then registered one by one. In the process, other generic concepts and generic relationships were found to be necessary and were added to the concept network as required. In this way, the experimenter's world view was reflected in the generic concept network which currently consists of 197 generic concepts and 67 generic relationships.

To determine how wide the knowledge base will have to be, the status of the concept network was logged each time an article was registered. As for the cost of entering information, it took 10-30 minutes to register an article and relevant information. Much of the time was spent thinking over the generic structure of the network; after the generic concepts and generic relationships saturated, it took about 10 minutes to register an article.

Currently, the network has 66 articles stored in it and now consists of 1078 concepts, 67 generic relationships, 980 instance relations and 1078 superclass relations. Among the concepts, 197 are generic and 881 are individual. The branching factor of the taxonomic hierarchy is 5.5 and the average depth is 6.7. Some examples of acquired objects are: 138 organizations including 61 companies, 21 universities, 14 research laboratories; 70 spatial units

including 42 cities; 37 science-technology fields, 38 kinds of computers, 68 published materials, 80 persons, and so on. Figure 9 shows a part of the acquired list of companies.

By studying the knowledge accumulation process log, the rates of increase in the number of concepts and relations per article were found to be about 11 and 15, respectively. Consequently, for a document file with 1000 items, the size of the knowledge base would be around 11,000 concepts and 15,000 relations. As for the linear growth of the network, we originally expected that at some point the growth would saturate to make a logarithmic growth, for example. However, since every article should carry new information, it will be reasonable to expect a linear growth as the number of articles increases. Another point assertained is that there is a tendency to store deeper knowledge as the size of the knowledge base increases. This experiment should be continued by all means to obtain more details regarding these points.

6. Conclusions

A personal universal filing system that can accumulate pieces of information to construct organized knowledge has been presented. The focus is placed on the storage and organization of knowledge in the framework of the user's world-view. The idea stems from an analysis of the meaning of "filing."

A concept-relation model for knowledge representation is used so that a general user can enter fragmental information without difficulty. In the model, the smallest unit of knowledge is the triad of two concepts (objects) and a relation defined between them; thus, the user can enter the fact easily by identifying the two concepts and specifying a relationship.

A prototype of the system, UNIFILE-I, has the Concept Network Editor as knowledge representation system. This editor enables the user to browse through the network, construct and edit the network, and retrieve information. The combination of a familiar tablular view and semantic retrieval has been found to be very effective.

An experiment involving the storing of many kinds of information extracted from about 70 articles has been conducted. As a result, a concept network of about 1000 concepts and 2000 relations has been constructed. Among these concepts, about 200 are generic ones so that we may conclude that the knowledge base so constructed is heterogeneous. By experiencing the experimental use of the UNIFILE-I and by dealing with the real world data, we now come to prefer to store new information into this system rather than store it as paper documents. This feeling is based on the guarantee of being able to see the stored information at any time in any form. We think that knowledge-based filing is a very effective tool for persons who must manage many kinds of information.

7. Acknowledgement

The authors are grateful to Zenji Tsutsumi and Tsuneyo Chiba for their support and encouragement in the course of this work, and also to Dr. Masakazu Ejiri and Dr. Yasuaki Nakano for their critical comments on this experimental system. They are also grateful to Prof. Tadao Saito and graduate students at University of Tokyo for their invaluable discussion. They also would like to acknowledge that the program on the personal computer to control the Hitfile-60 directly has been designed by Masaaki Fujinawa, and that the UTILISP developed at University of Tokyo has been used to write the Concept Network Editor on a host computer.

8. References

1. W. Morris, Ed., "The American Heritage Dictionary of the English Language," Houghton Mifflin Company, Boston, 1980.
2. "Roget's International Thesaurus," revised by R. L. Chapman, Fourth Edition, Thomas Y. Crowell, Publishers, New York.

3. M. M. Zloof, "Query-by-Example," AFIPS Conference Proceedings, National Computer Conference, 44, 1975, pp.431-438.
4. J. F. Sowa, "Conceptual Graphs for a Data Base Interface," IBM J. Res. Develop. July, 1976, pp.336-357.
5. F. W. Lancaster, "Toward Paperless Information Systems," Academic Press, New York, 1978.
6. G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, J. Slocum, "Developing a Natural Language Interface to Complex Data," ACM Transactions on Database Systems, Vol. 3, No. 2, Jun. 1978, pp.105-147.
7. S. Chang, J. Ke, "Translation of Fuzzy Queries for Relational Database System," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No. 3, July 1979, pp. 281-294.
8. H. Inose, T. Saito, and H. Nakagawa, "On-line Remote Access System for a Pattern Data Base Featuring Microfiche Storage and Facsimile Recording," Conf. Rec. National Telecommunication Conf., 1980, No. 16.3.
9. C. F. Herot, "Spatial Management of Data," ACM Transactions on Database Systems, Vol. 5, No.,4 December 1980, pp.493-514.
10. R. E. Fikes, "A Represetation System User Interface for Knowledge Base Designers," The AI Magazine, Fall 1982, pp.28-33.
11. F. N. Tou, M. D. Williams, A. Henderson, T. Malone, "RABBIT: An Intelligent Database Assistant," Proceedings AAAI-82, Pittsburgh, Pennsylvania, American Association for Artificial Intelligence, 1982, pp.314-318.
12. R. J. Brachman, "What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Network," Computer, Vol. 16, No. 10, 1983, pp.30-36.
13. R. J. Brachman, R. E. Fikes, and H. J. Levesque, "Krypton: A Functional Approach to Knowledge Representation," Computer, Vol. 16, No. 10, 1983, pp.67-73.
14. P. P. Chen, "English Sentence Structure and Entity-Relationship Diagrams," Information Sciences, Vol. 29, Elsevier Science Publishing Co., New York, 1983, pp.127-149.
15. Y. Tsunoda, S. Horigome, Z. Tsutsumi, and S. Abe, "Large Capacity Optical Disk File," The Hitachi Hyoron, Vol. 65, No. 10, Oct. 1983, pp.23-28. (in Japanese)
16. K. Izawa, T. Yoneyama, M. Kanno, T. Kamiyama, H. Tezuka, S. Takagi, "Visually Assisted Document File System," Proceedings of Japan Display '83, Tokyo, 1983, pp.526-529.
17. J. H. Bryant, "Hardware-Software for Very Large Information Retrieval," COMPCON '84(SPRING), pp.160-163.
18. P. F. Patel-Schneider, R. J. Brachman, and H. J. Levesque, "ARGON: Knowledge Representation meets Information Retrieval," Proceedings of the First Conference on Artificial Intelligence Applications, 1984, pp.280-286
19. Craig W. Thompson, "Recognizing Values in Queries or Commands in a Natural Language Interface to Database," Proceedings of the First Conference on Artificial Intelligence Applications, 1984, pp.25-30.
20. R. J. Brachman and J. G. Schmolze, "An Overview of the KL-ONE Knowledge Representation System," Cognitive Science 9, 1985, pp.171-216.
21. J. Higashino, H. Fujisawa, Y. Nakano, and M. Ejiri, "A Method for Understanding Document Images using a Form Definition Language," Annual Symposium of Institute of Electronics Communication Engineers of Japan, 1985, No. S10-2. (in Japanese)
22. A. Bookstein and D. R. Swanson, "Probabilistic Models for Automatic Indexing," J. American Society of Information Science, Vol. 25, 1974, pp.312-318.
23. S. F. Dennis, "The Design and Testings of a Fully Automatic Indexing-searching System for Documents Consisting of Expository Test, Information Retrieval--A Critical Review," Thomson Book, Washington D. C., 1967, pp.67-94.
24. G. Salton, "Dynamic Information and Library Processing," Prentice-Hall, 1975, p.82.
25. G. Salton, "Mathematics and Information Retrieval," J. Documentation,Vol. 35, No. 1, Mar. 1979, pp.1-29.

ENT#	CONCEPT
C#1065	Picon
C#1064	LAMBDA/PLUS
C#1063	ELECTRONICSWEEK-840827
C#1062	Lisp machine provides a shell for industrial AI applications in one of first expert systems to go to work
C#1061	ARTICLE#0106
C#1060	PRESIDENT
C#1059	Britton, David
C#1058	IDM-PC
C#1057	Britton's IDM line
C#1056	Britton Lee intelligent database machine
C#1055	DATABASE-MACHINE
C#1054	ELECTRONICSWEEK-841126
C#1053	Britton Lee fortifies position in dormant data-base market
C#1052	ARTICLE#0121
C#1051	VICE-PRESIDENT
C#1050	Smith, John Miles

C

(a) Concept name table

CHILD	PARENT
C#1065	C#0882
C#1064	C#0376
C#1063	C#0703
C#1062	C#0201
C#1061	C#0409
C#1060	C#0254
C#1059	C#0019
C#1058	C#1057
C#1057	C#0085
C#1056	C#1055
C#1055	C#0225
C#1054	C#0703
C#1053	C#0201
C#1052	C#0409
C#1051	C#0254
C#1050	C#0019
C#1049	C#0019

S

(b) Superclass relation table

REL#	RELSHIP#	CONCEPT-L	CONCEPT-R	CLASS
R#1070	RS#0051	C#0026	C#0046	INST
R#1069	RS#0021	C#1061	C#1062	INST
R#1068	RS#0029	C#1061	C#0723	INST
R#1067	RS#0008	C#1063	C#1061	INST
R#1066	RS#0007	C#1061	C#1064	INST
R#1065	RS#0007	C#1061	C#1065	INST
R#1064	RS#0022	C#1064	C#0723	INST
R#1063	RS#0063	C#0723	C#1064	INST
R#1062	RS#0039	C#1064	C#0445	INST
R#1061	RS#0036	C#1064	C#1065	INST
R#1060	RS#0063	C#0723	C#1065	INST
R#1059	RS#0056	C#1065	C#0307	INST
R#1058	RS#0022	C#1065	C#0723	INST
R#1057	RS#0051	C#0039	C#0139	INST
R#1056	RS#0022	C#0270	C#0071	INST
R#1055	RS#0063	C#0072	C#0270	INST
R#1054	RS#0039	C#0270	C#0544	INST

r

G

(c) Instance relation table

RELSHIP#	RELATIONSHIP	LR	RL	REL#
RS#0010	ACADEMIC-TITLE	HAS-TITLE-OF	IS-GIVEN-TO	R#0031
RS#0011	AFFILIATION	WORKS-AT	HAS-EMPLOYEE-OF	R#0036
RS#0029	ANNOUNCEMENT	IS-ANNOUNCED-BY	HAS-ANNOUNCED	R#0171
RS#0049	ATTACH	ATTACHES	IS-ATTACHED-TO	R#0555
RS#0001	AUTHORSHIP	IS-AUTHOR-OF	AUTHOR-IS	R#0001
RS#0028	AUTHORSHIP2	IS-AUTHOR-OF	AUTHOR-IS	R#0152
RS#0052	COAUTHORSHIP	IS-COAUTHOR-OF	COAUTHOR-IS	R#0699
RS#0053	COAUTHORSHIP2	IS-COAUTHOR-OF	COAUTHOR-IS	R#0700
RS#0060	CONFERENCE	IS-PROCEEDINGS-OF	HAS-PROCEEDINGS-OF	R#0848
RS#0016	DATE-OF-EVENT	DATE-IS	IS-DATE-OF	R#0074
RS#0062	DELIVER	DELIVERS	IS-DELIVERED-BY	R#0984
RS#0063	DELIVER2	DELIVERS	IS-DELIVERED-BY	R#0985
RS#0022	DEVELOPMENT1	IS-DEVELOPED-AT	HAS-DEVELOPED	R#0130
RS#0023	DEVELOPMENT2	IS-DEVELOPED-AT	HAS-DEVELOPED	R#0131
RS#0038	DEVELOPMENT3	IS-DEVELOPED-AT	HAS-DEVELOPED	R#0307
RS#0061	DO	CARRIES-OUT	IS-CARRIED-OUT-	R#0865

(d) Generic relationship name table

Figure 1. Knowledge representation by four types of tables

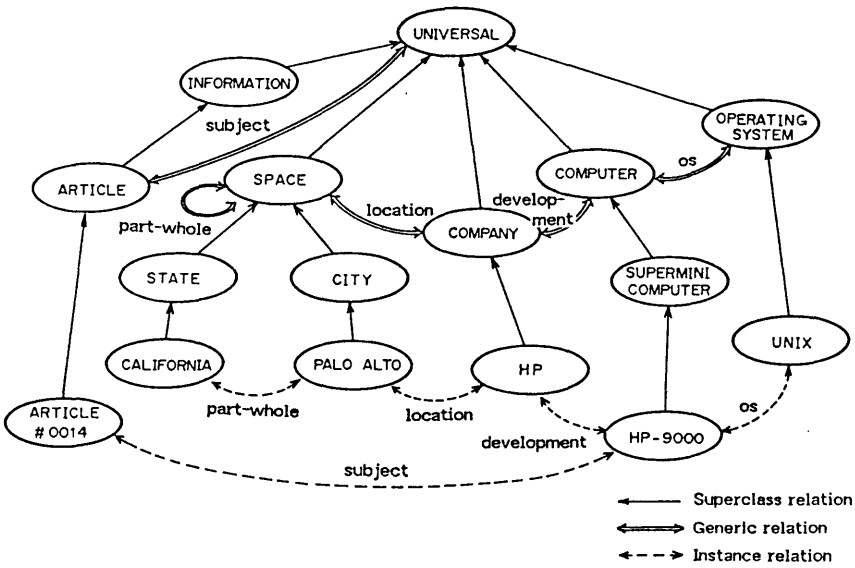


Figure 2. Example of the concept network

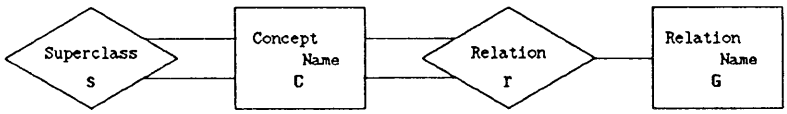
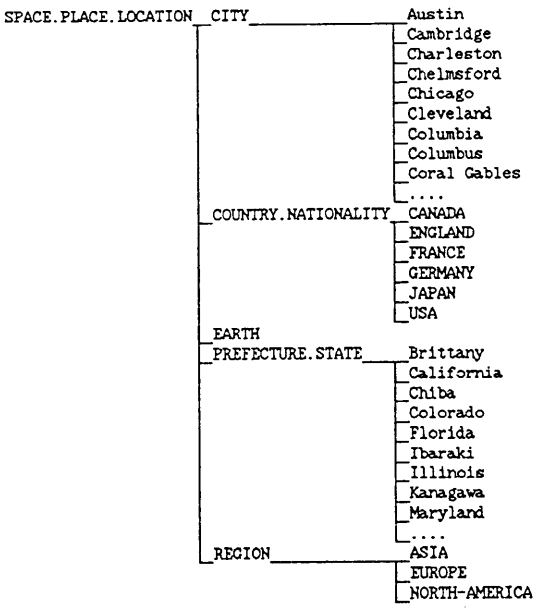
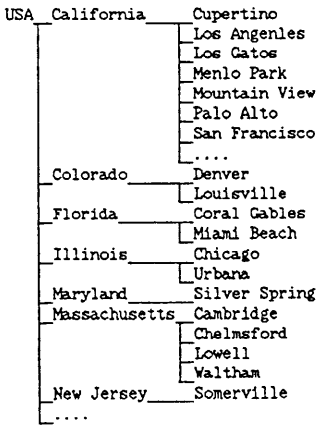


Figure 3. Entity-relation diagram for the concept-relation model



(a) Taxonomic hierarchy



(b) Partonomic hierarchy

Lisp Machine Inc.	CONCEPT	NO
IS-A	COMPANY	1
ADDRESS-IS	6033 W. Century Blvd., Los Angels, Calif. 90045	2
DELIVERS	Picon	3
DELIVERS	LAMBDA/PLUS	4
HAS-ANNOUNCED	ARTICLE#0100	5
HAS-ANNOUNCED	ARTICLE#0106	6
HAS-DEVELOPED	LAMBDA-4X4	7
HAS-DEVELOPED	Picon	8
HAS-DEVELOPED	LAMBDA/PLUS	9
IS-LOCATED-IN	Los Angenles	10

(c) Frame view

NEWS-ARTICLE	TITLE-IS	IS-ANNOUNCED-BY
ARTICLE#0093	Macintosh software combines filing, drawing	Telos Software Products
ARTICLE#0100	High-performance system that runs under lisp brings down the cost per user	Lisp Machine Inc.
ARTICLE#0014	HP taps Unix for all its lines	HP.Hewlett Packard Co.
ARTICLE#0010	Minicomputer adds networking clout to one of the smallest personal computers	GRID-SYSTEMS-CORP
ARTICLE#0003	New Apples might compete head on	Apple Computer Inc.
ARTICLE#0136	Engineering work station has internal Ethernet interface	Versatec
ARTICLE#0106	Lisp machine provides a shell for industrial AI applications in one of first expert systems to go to work	Lisp Machine Inc.

(d) Tabular view

Figure 4. Versatile view on concepts

NO	*LISP	SUPERCLASS
1	A Summary of MacLISP Functions and Flags	TITLE.HEADLINE
2	COMMON-LISP	LISP
3	Fujitsu claims its Lisp machine is fastest processor	TITLE.HEADLINE
4	Fujitsu set to ship fast lisp machine	TITLE.HEADLINE
5	FRANZLISP	LISP
6	High-performance system that runs under lisp brings down the cost per user	TITLE.HEADLINE
7	Interlisp-D	LISP
8	INTERLISP	LISP
9	Lisp machine provides a shell for industrial AI applications in one of first expert systems to go to work	TITLE.HEADLINE
10	Lisp Machine Inc.	COMPANY
11	LISP	PROGRAMMING-LANGUAGE
12	LISP-MACHINE	COMPUTER
13	LISP: A Gentle Introduction to Symbolic Computation	TITLE.HEADLINE
14	MACLISP	LISP
15	UTILISP	LISP
16	ZETALISP	LISP
17	ZETALISP-PLUS	ZETALISP

Figure 5. Menu showing the result of substring matching

COMPANY	CONCEPT	NO
ADDRESS-IS	MAILING-ADDRESS	1
DELIVERS	MACHINERY.DEVICE	2
DELIVERS	SOFTWARE	3
HAS-ANNOUNCED	NEWS-ARTICLE	4
HAS-DEVELOPED	MACHINERY.DEVICE	5
HAS-DEVELOPED	COMPUTER-SOFT	6
HAS-DEVELOPED	SYSTEM.STRUCTURE	7
HAS-EMPLOYEE-OF	PERSON	8
HAS-PART-OF	ORGANIZATION.WORKPLACE	9
IS-LOCATED-IN	SPACE.PLACE.LOCATION	10
IS-MENTIONED-IN	ARTICLE	11
IS-PART-OF	ORGANIZATION.WORKPLACE	12
PRODUCES	MACHINERY.DEVICE	13
PUBLISHES	PUBLISHED-MATERIAL	14

Figure 6 (a) Generic frame COMPANY

COMPUTER	CONCEPT	NO
IS-A	ELECTRONIC-MACHINERY	1
ATTACHES	ELECTRONIC-MACHINERY	2
HAS-INTERFACE-OF	INTERFACE-DEVICE	3
HAS-PART-OF	MACHINERY.DEVICE	4
IS-ATTACHED-TO	COMPUTER	5
IS-DELIVERED-BY	COMPANY	6
IS-DEVELOPED-AT	ORGANIZATION.WORKPLACE	7
IS-EQUIVALENT-TO	ARTIFACT	8
IS-EQUIVALENT-TO	ARTIFACT	9
IS-MENTIONED-IN	ARTICLE	10
IS-PART-OF	MACHINERY.DEVICE	11
IS-PREDECESSOR-OF	ARTIFACT	12
IS-PRODUCED-BY	ORGANIZATION.WORKPLACE	13
IS-PROPOSED-BY	PERSON	14
IS-SUBJECT-OF	PUBLISHED-MATERIAL	15
IS-SUBJECT-OF	CONFERENCE	16
IS-SUBJECT-OF	ARTICLE	17

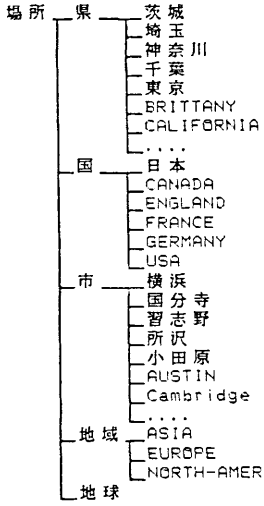
Figure 6 (b) Generic frame COMPUTER

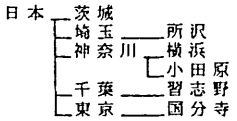
COMPUTER	IS-DEVELOPED-AT	RUNS	IS-SUBJECT-OF
APPLE-LISA	Apple Computer Inc.	.	ARTICLE#0003
APPLE-MACINTOSH	Apple Computer Inc.	Filevision	ARTICLE#0003 ARTICLE#0093
B-16	Narashino Works Narashino Works	.	
CADMUS-9000	Cadmus Computer Systems Inc.	.	ARTICLE#0057
CMU Work Station	Information Technology Center	.	
COMPASS-CENTRAL-MINICOMPUTER	GRID-SYSTEMS-CORP	.	
Dolphin	PARC.Xerox Palo Alto Research Center	LISP RABBIT SMALL-TALK Interlisp-D	
Domain II	Apollo Computer Inc.	.	

Figure 7 (a) Table COMPUTER before selection

COMPUTER	IS-DEVELOPED-AT	RUNS	IS-SUBJECT-OF
APPLE-LISA	Apple Computer Inc.	.	ARTICLE#0003
APPLE-MACINTOSH	Apple Computer Inc.	Filevision	ARTICLE#0003 ARTICLE#0093
COMPASS-CENTRAL-MINICOMPUTER	GRID-SYSTEMS-CORP	.	ARTICLE#0010
HP-3000	HP.Hewlett Packard Co.	X-ample PWD-GEOGRAPHIC-DATABASE-SYSTEM	
HP-9000	HP.Hewlett Packard Co.	HPRL.Heuristic Programming & Representation Language	ARTICLE#0014
LAMBDA-4X4	Lisp Machine Inc.	LISP ZETALISP-PLUS LM-PROLOG	ARTICLE#0100
LAMBDA/PLUS	Lisp Machine Inc.	LISP Picon	ARTICLE#0106

Figure 7 (b) Table after applying semantic condition (IS-LOCATED-IN CALIFORNIA) to the column IS-DEVELOPED-AT, i.e. COMPANY





No.	日立製作所	
1	会社	である
2	ARTICLE#0115	で言及されている
3	東京都	にある
4	HITACHI-PERSONAL-TERMINAL	を開発した
5	特許部	を持つ
6	神奈川工場	を持つ
7	中央研究所	を持つ
8	システム開発研究所	を持つ
9	マイクロ電子機器開発研究所	を持つ
10	習志野工場	を持つ
11	小田原工場	を持つ
12	HITACHI-REVIEW	を出版した
13	日立評論	を出版した
14	M-200H	を生産する
15	HITACHI-PERSONAL-TERMINAL	を生産する
16	ARTICLE#0015	を発表した
17	B-16	を販売する

(a) Taxonomic hierarchy (b) Partonomic hierarchy (c) Instance frame HITACHI

Figure 8. Extension to Japanese language

COMPANY	ADDRESS-IS
TI.Texas Instruments Inc.	PO Box 226015, MS 238, Dallas, Texas 75266
Cadmus Computer Systems Inc.	600 Suffolk St., Lowell, Mass. 01853
Landmark Software Systems	155 W. Main St., Somerville, NJ 08876
CGI.Carnegie Group Inc.	650 Commerce Court, Station Square, Pittsburgh, Pa 15219
Lisp Machine Inc.	6033 W. Century Blvd., Los Angels, Calif. 90045
Academic Press Inc.	111 Fifth Avenue, New York, NY 10003
Britton Lee, Inc	14600 Winchester Blvd., Los Gatos, Calif. 95030
CCA.Computer Corporation of America	575 Technology Square, Cambridge, MA 02139
Three Rivers Computer Corp.	720 Gross St., Pittsburgh, Pa 15224
Apollo Computer Inc.	330 Billerica Road, Chelmsford, Mass. 01824
Versatec	2710 Walsh Ave., Santa Clara, Calif. 95051

Figure 9. Example of knowledge acquisition - Table of companies and their addresses