数理解析研究所講究録
第 593 巻 1986 年 275-294

Towards a Logical Approach to Building a Frame-Based Knowledge Representation System: Preliminary Report

伊藤 秀昭 滝沢 誠 上野 晴樹
Hideaki ITO* Makoto TAKIZAWA** Haruki UENO**

* Japan Information Processing Development Center (JIPDEC), 3-5-8, Shibakoen, Minatoku, Tokyo 1Ø5, Japan

** Department of Systems Engineering, Tokyo Denki University, Hatoyama, Saitama 35Ø-Ø3, Japan

ABSTRACT

This paper presents a formalization of a frame-based knowledge representation system which has been developed to represent knowledge and used for building knowledge-based systems. In order to formalize the knowledge representation system, we define an abstract frame-based data structure and an abstract machine which manipulates it. Then, a frame-based knowledge representation system is reconstructed with the formal semantics. The most characteristic of the conventional frame-based knowledge representation system is flexibility and freedom of representing of knowledge. Although this is a great advantage, they lack a formalism. The lack of a formal aspect of the system may cause some problems. For example, they have a mixed aspect of both a physical and a logical structure. In order to solve such problems, it is necessary to formalize the frame-based knowledge representation system. We give a formal definition of the frame-based knowledge representation system based on a first-order theory.

In this paper, definitions on the frame-based data structure are given and basic operations on them are defined. A virtual frame machine which manipulates the structure based on these operations is built. Next, over the abstract machine, a formal system based on the first-order theory is defined.

1. Introduction

Recently, various knowledge-based systems have been developed[Waterman]. The knowledge-based system is a software system to solve problems in a specific domain (e.g., diagnosis, consultation, design, etc.). In general, the knowledge-based system is composed of two basic components, a knowledge base and an inference engine[Waterman], [Hayes-Roth]. Knowledge stored in the knowledge base is represented in a certain knowledge representation model. In order to build a useful knowledge-based system, The representation of domain-specific knowledge is one of the key problems. We define a knowledge representation system to be a system which is used for developing the various knowledge-based systems. A knowledge representation model is a knowledge representation and an inference mechanism.

Already, various means of knowledge representation have been proposed. They are based on production rules[Davis], semantic networks[Brachman], logics, frames[Minsky], and combinations of some representation, e.g., LOOPS[Bobrow]. Recently, many representation system based on the frame have been developed, e.g., ZERO[Ito86], Krypton[Brachman85a], KL-ONE[Brachman85b], UNITS[Stefik79]. Reasons why frame-based systems have been developed are follows;

(1) The knowledge base is represented in a hierarchical data structure called a generalization hierarchy.

(2) Inference mechanisms can be write by means of procedures attached to the frames.

Most advantage of the frame-based knowledge representation system is that its data structure is flexible. However, the flexibility causes the following disadvantages.

(1) It is difficult to formally specify a data structure and an inference mechanisms, except that the specification is described in the language which is used for implementing the frame-based knowledge representation system.

(2) It may be difficult to verify the correctness of the system.

(3) There is no way for evaluation the frame-based knowledge-based system objectively, except that it applies to an actual problem or a certain task domain.

(4) Since semantics of the knowledge-based system are given by operations on the knowledge base in terms of a system implementation language, the operations do not have formal semantics. Therefore, in the conventional frame-based knowledge representation systems, both aspects of a physical structure and a logical one are mixed.

Furthermore, the lacking of a formal semantics of the knowledge representation system cause the problem that it is difficult to manage the frame-based knowledge base. If the application area of the knowledge-based system is extended then it is possible to consider that the knowledge base becomes larger one. Then, it is difficult to manage the large knowledge base by a user of the knowledge-based system. From this point of view, one of the most fundamental problems is integration of the knowledge representation model (frame model) and a data model of a database.

So far, the frame-based data structure has been tried, such as Krypton, KL-TWO[Vilain], and ones based on the denotational semantics[Attardi],[Reimer], etc. Krypton and KL-TWO are based on the extensional semantics and lambda-calculus, respectively. Our approach is based on a first-order theory[Enderton] and a database logic[Jacob].

Already, we have developed a frame-based knowledge representation system, called ZERO. In the ZERO knowledge representation, Horn clauses (Prolog statements[Clocksin]) can be attached into a frame. The ZERO system has three major features which are developed to embed Prolog into the system; a Prolog-based message passing, an extension of unification mechanisms and a function for non-deterministic behavior by backtracking for a frame system. Using the ZERO system, highly flexible intelligent systems could be achieved. In this paper, we attempt to make clear the semantics of the frame-based data structure based on ZERO knowledge base. Then, we try to formalize the restricted functions and the data structure based on the ZERO system.

The purposes of our work are as follows;

(1) The properties of the data structure provided with a frame-based knowledge representation system are made clear in a sense of a logical semantics. The semantics of several frame-based knowledge representation systems are defined in operational semantics which are proposed by the designers of systems, only. Hence, the system has a mixed aspect of both logical and physical structure. For this reason, it is possible to consider that its logical semantics could not be established sufficiently.

(2) A unified frame system is proposed. As described above, many frame-based knowledge representation systems are designed. Through experiments of several works, basic mechanisms of the system has been proposed. However, there are difference among several systems with respect to these operational semantics. It is need to propose the unified frame system in a sense of the operations on the frame-based knowledge base.

(3) When a knowledge base becomes largely, a connection of a knowledge-based system and a database system should be required. Then, the idea of the connection are not only just to combine them, but also it is possible to consider that users of the system can handle them without being hardly aware of the difference among them.

(4) This work is a logical basis for a program design problem. We have a subject of research which is a software design. For this subject, it is possible to suppose that properties of tools used for building the software design system may be made clear.

In this paper, the frame-based knowledge representation system is defined based on a first-order logic, and its data structure and operations are made clear. The data structure is defined to give the interpretation of a first order language. For this aim, we define a frame model based on a data structure, operations and constraints. Using the frame model, a virtual frame machine are built.

2. An Overview of a Frame-Based Knowledge Representation

In a frame-based knowledge representation system, knowledge are represented by a hierarchical tree of frames. A frame is composed of a set of slots. The slots described the properties or the attributes of objects represented as the frames. The slots are filled with these values. Here, a frame data model is composed of the data structure and the operations on the data structure.

The knowledge defined by means of the frame data structure can be rewritten in the form of assertions in the first-order language [Hayes],[Israel]. They say that frames and semantic networks as respect to factual knowledge can be written in a set of the assertions. In ZERO, the frames are counterparts of individuals in the first-order logic[Ito86]. And, a slot is corresponding to a two-place predicate[Ito86]. Although the slot consists of some field (e.g., datatype field to specify the type of slot value, default field to describe the default value, etc.), the ZERO system treats the slot as a two-place predicate. The reason why the representation of the slot is simplified is that it is possible to consider that using many parameters for getting something (i.e., slot value) becomes a burdensome for users of system, and although there is a lose of information (e.g., datatype, default, etc.) such information should not be used for retrieving data defined in the frame system as key parameters. A general form to represent the slot is "S (F VALUE)". Where, S, F, and VALUE are a slot name, a frame name and a slot value respectively. This expression is read as "S of F is VALUE.".

In a frame-based data structure, two kinds of knowledge are defined explicitly, they are;

(1) the facts described in slots in a frame (in forms of binary relations),

(2) the implications (rules) denoted by a generalization hierarchy.

Fig. 1 shows a generalization hierarchy and a frame, and Fig. 2 a subset of facts and implications represented in Fig. 1. The frame "Pochi" is composed of some slots. For example, the slot "Color" describes the fact that; the color of "Pochi" is white. And, a dog named "Pochi" denoted by the frame "Pochi" is a dog. A dog described by the frame "Dog" is a kind of "Mammal". In this case, we can infer that; Pochi is a mammal. These sentences can be written in the following first-order sentences.

color-of(Pochi White)
is-a(Pochi Dog)
a-kind-of(Dog Mammal)
is-a(Pochi Dog) <u>and</u> a-kind-of(Dog Mammal)->is-a(Pochi Mammal).

In general, the generalization hierarchy with no exceptions and simple inheritance must satisfy the following axiom schemata.

(1) ∀x∀y∀z is-a(x y) <u>and</u> a-kind-of(y z)->is-a(x z)

(2) ∀x∀y∀z a-kind-of(x y) <u>and</u> a-kind-of(y z)->a-kind-of(x z)

(3) For every two-place predicate P corresponding to a slot.

∀x∀y∀z P(x y) <u>and</u> (is-a(z x) <u>or</u> a-kind-of(z x))->P(z y).

Here, both the is-a and the a-kind-of links are a link which connects between two nodes (frames). Furthermore, the link a-kind-of and the is-a are represented in a slot. In the ZERO system, in order to classify the semantics of these links, every frame has a frametype. There are two kinds of the frametypes; an instance and a class. And, every frame has a named a-kind-of slot which indicates a parent frame in a hierarchical structure. Therefore, if a frame has the frametype instance (called an instance frame), there is the is-a link between the frame and the frame indicated by the value of the a-kind-of slot (called an AKO frame). The difference between the two types of a link is whether the frametype is an instance or a class. If a frame whose frametype is a class, the frame and its AKO frame are linked by the a-kind-of link.

3. Building a Frame Model

3.1. The Levels of Frame-Based System

Fig. 3 shows a basic organization to formalize a frame data model. These are organized as a hierarchical structure. The criterion of this organization is based on the development process of a knowledge-based system. The level indicates how to describe a frame-based knowledge-based/representation system. This paper is concerned with both the level 2) and 3) in Fig. 3.

The physical frame system is described in terms of a physical implementation into a computer system. At this level, for example, the system is described by constraints on memories and an internal representation of the frame data structure, etc.

A virtual frame system indicates a virtual frame machine which has a virtual frame data structure and operations. Therefore, basic mechanisms which are provided with existing frame-based knowledge representation system must be represented by means of the virtual frame machine. In this paper, the frame data structure is formalized, and the virtual frame data structure are defined by introducing a total ordering on the frame data structure. Furthermore, the operations are given to handle the virtual frame data structure. A conceptual frame system is the system that is obtained through the definitions in terms of the virtual frame system. Moreover, the system describes properties of an application frame system. In this work, the system is represented in the form of a first-order language. For this reason, we put the restriction on the structure of an actual frame-based knowledge representation system. Therefore, it can be made the range clear that we can accept the results deduced from a frame-based knowledge-based system. The description at this level is called a conceptual representation level.

An application frame system indicates a frame-based knowledge representation system. Therefore, the application frame systems have several inherent features. For example, ZERO, KL-ONE, UNITS, etc. are at this level. Those systems have unique constraints for representing knowledge. A frame-based knowledge-based system which are built using the application frame system is called a frame-based knowledge-based system (e.g., MOLGEN[Stefik81], INTELLITUTOR[Ueno84]).

3.2. Frame Model

In order to put a logical interface (i.e., to define a conceptual representation), it is required to clear the frame-based knowledge representation. In most frame-based knowledge representation systems, since both a physical and a logical aspects are mixed, the semantics of the system is confusing. A formal frame model is defined to solve this problem. For this purpose, each aspect, i.e., physical, logical, is explicitly separated. At first, in order to obtain a frame model, a frame data structure are defined to give an interpretation based on a first-order logic. Next, we define a frame model based on the frame data structure by adding the physical aspect, i.e., the functions to retrieve the data defined in the frame model, and setting constraints.

3.2.1. Frame Data Structure

A frame data structure is composed of two kinds of objects: records (frames) and links (relationships between frames). The frame data structure can be described in a graph whose nodes and links denote the records and the links, respectively.

Fig. 4 shows the data structure which is the basis of the formalization of a frame data model. This is based on the data structure of frames in ZERO. This structure is a simplified frame data structure in the knowledge representation of the ZERO system. In Fig. 4., (a) and (b) show primitive data structure of a frame and adopted one for formalizing the frame data structure by transformation of the viewpoint how to write a frame, respectively.

We define a frame model FM to be a triple (\$\$, OO, CC). Here, \$\$ is a physical frame data structure, OO a set of operations and CC a set of constraints. \$\$ has a physical aspect. It is obtained through introducing some ordering relation on a conceptual frame data structure SS (see, latter). By abstracting the logical aspect from the frame data structure, we can get the conceptual frame data structure SS.

3.2.1.1. Conceptual Frame Data Structure

A conceptual frame data structure SS is a set of pairs ($\underline{S}$, S), each of which represents a frame structure. $\underline{S}$ is a frame structure schema ($\underline{FF}$, $\underline{RR}$) and S a frame structure instance (FF, RR). Where, $\underline{FF}$ is a set of frame schema $\underline{F}$, $\underline{RR}$ a set of link schema $\underline{R}$, FF a set of frame instance F, and RR a set of link instances R. A frame is represented as a pair of a frame schema $\underline{F}$ and a frame instance F, ($\underline{F}$, F) (see, Fig. 4). Relationships on frames are represented by a set of ($\underline{R}$, R).

We described above, the basis for a frame data structure is a schema and its instance. The schemata are a time-invariant and a logical data structure. While, the instances are a set of time-variant data. This structure are obtained through the abstraction of a frame-based data structure. The frames are composed of a collection of a sets and partial functions among the sets.

A frame schema $\underline{F}$ is a set of items which is a set of slots {@F, s1, s2, ... , sm} ($m \geq 0$), where @F is a frame-id which is an identifier in a set of frames. In Fig. 4, a frame-id is a frame name. And, si ($0 \leq i \leq m$) is called a slot item. For every slot item si in $\underline{F}$, let dom(si) denote its domain. A frame instance F is obtained through filling with actual data, which is a subset of direct product dom(@F) x dom(s1) x dom(s2) x ... x dom(sm). That is;

$$F \in FF \subseteq dom(@F) \times dom(s1) \times \ldots \times dom(si) \times \ldots dom(sm).$$

Let f be a frame instance, vi a value of a slot item (slot value), @Fi a frame-id of f, and the relationship between @Fi and vi is indicated by;

si(@Fi, vi).

And, for every slot item st, let st(f) denote st's value of f. Every f has a different value @F(f).

A set of link instance, R c F1 x F2, are obtained when actual data are given to a link schema $\underline{R}$. Here, let r (=(f1, f2)) be an element of RR. Such r is a partial function R:F1->F2. R is called an AKO link.

We summerize above definitions. The frame data structure is defined based on the frames ($\underline{F}$, F) and the relationships ($\underline{R}$, R). Hence, a frame structure schema $\underline{S}$ and its instance S are given $\underline{S}$=($\underline{FF}$, $\underline{RR}$) and S=(FF, RR), respectively. Each pair is a family of frame schemata and link schemata. In the link schema $\underline{R}$=($\underline{F1}$, $\underline{F2}$) and a link instance R=(F1, F2), $\underline{F1}$, $\underline{F2}$, F1 and F2 are called a child frame schema, a parent frame scheme, a child frame instance and a parent frame instance, respectively. The frame structure schema $\underline{S}$ can be represented in a graph. This graph is called a frame schema graph (FSG). For example, Fig. 5 shows the FSG of $\underline{\text{Mammal}}$ and $\underline{\text{Dog}}$. In the FSG, a node and a link correspond to an element of $\underline{FF}$ and an element $\underline{RR}$, respectively. Then the labels of the nodes are distinct each other. Each arcs on a FSG has a label called AKO. An extension of $\underline{S}$ is given by an instance graph (FIG) S=(FF, RR). In this graph, the nodes and the arcs correspond to the frame instances and the link instances respectively. Fig. 6 shows the FIG of the FSG shown in Fig. 5. In Fig. 5, for $\underline{R}$= ($\underline{\text{Mammal}}$, $\underline{\text{Dog}}$), a parent frame schema is $\underline{\text{Mammal}}$ and a child frame schema is $\underline{\text{Dog}}$.

3.2.1.2. Physical Frame Data Structure

\$\$ is a physical frame data structure, which is obtained through introducing an ordered relation into SS.

Now, we consider a context free grammer G=<Vn, Vt, P, Root>. Where, Vn is a set of nonterminal symbols, Vt a set of terminal symbols, P a set of production rules, Root an initial symbol. Let be V =Vn U Vt, and Root e Vn. P is a collection of rules A -> B1, B2, ... , Bn. Then, P must satisfy with the following conditions.

1) for every symbol appeared in a rule Aj, it is not the case that Aj*=>vi Aj v2 for some symbols v1 and v2 in V, where the symbol *=> denotes several applications of a derivation.

2) for every two different rules, Aj and Ai ($j \neq i$) are different each other.

3) for every rule in P and every symbol in the rules, the symbols in

the right hand side appear only at once ($\forall i \forall j$ ($Bi \neq Bj$)).

A derivation of the context free grammer G can be described in a derivation tree GT. GT is represented by (V, E), where, E is a set of directed edges <nl, n2>. The nl and n2 are called a descendent node and a parent node, respectively.

By the way, a FSG is a tree. Hence, by giving a correspondence between the FSG and the GT, the FSG could be described in the GT. There is the correspondence between them. It is clear that V and E correspond to FF and RR. Hence, there is the FSG representing a certain GT. Such a FSG and a GT are an isomorphic graph.

As a FSG and a GT has an identical structure, a FSG is an acyclic graph. For every ni∈V in a GT, if <nl,n2>∈E then nl and n2 are adjacent. Ad(n2) is defined as {ni | ni∈V, <ni, n2>∈E}. By this definition, there is an 1-to-1 correspondence between a node ni∈V and its Ad(ni), all nodes in Ad(ni) are linked as the list which is called an adjacency list.

Let's define an ordering procedure on a GT. The GT is an acyclic direct graph. The ordering procedure of the GT can be given a depth-first and right-to-left manner using the adjacency list. This property is a well known in a graph theory[Iri]. Clearly, this procedure becomes an ordering function OF. Therefor, the nodes in a GT is enumerated by the ordering function OF. Since FF corresponds to V, the ordering of the nodes in the GT correspondence to the ordering of the set of frame schemata. Hence, a set of frame schemata is defined as a totally ordered set (F*, <<). Where, for every f and f'∈F*, a set of frame instances is ordered by the depth-first and right-to-left manner on the FIG, and << is an ordered relation in a sense of the OF. By introducing the ordered relation on SS, $$ is defined. That is, $$ is a physical frame data structure.

3.2.2. Operations

We defined operations on the physical frame data structure $$. In this paper, we consider only retrieval operations. By introducing the operations, we can consider a virtual frame system which is seems to be a virtual frame machine. The frame data structure is a family of a set of frame instances FF and a set of link instances RR. In order to define operations on the virtual frame machine, we ought to put the criteria, because the semantics of the frame-based knowledge representation system is made clear by means of the behavior of these operations. The criteria are as follows;

(1) At present, the virtual operations do not have functions to update the frame instances. It is possible to consider that such function do not match with a first-order logic. Therefore, the operations on the virtual frame machine are composed

of the access functions to the frame data structure, only.

(2) The strategy of retrieving a data object is navigational manner likes CODASYL database[Olle] has by introducing a total ordering into a FSG and a FIG.

A virtual frame machine has three kinds of registers. The register Current-frame and Current-slot have a specific role. They have functions like a currency indicator in CODASYL database, and each resister stores a frame-id as these value. The register Current-frame store the currency indicator for the navigation which depends on the hierarchical structure defined in Sec. 3. The register Current-slot is used for preserving the currency, whose value is the frame-id. Its frame instances contains the slot which is specified by a parameter of some operations. Also, the navigation depends on the total ordering of the FIG described in Sec. 3. Other registers are used for preserving a temporal data.

We define the operations as follows. A user can handle the frame model without updating the frame data structure. Also, these operations are designed to implement the ZERO system.

(1)Frames(KB): An input is a frame data structure, and an output is a set of a frame-id defined in the input frame data structure.

(2)Slots(Frame): An input is a frame-id, and an output is a set of slot items.

(3)Slotval(Slot, Frame): An output is the slot value indicated by the arguments Slot and Frame.

(4)Current-frame(): An output is the value of the register Current-frame.

(5)Left-frame(): An output is the frame-id which indicates a brother frame. Then, the the value of the register Current-frame is changed to the brother's frame-id. Such that, @F(fc)<<@F(fb), where fc is a Current-frame instance, fb a brother frame instance. The distance between @F(fc) and @F(fb) is shorter than other brother frame instance.

(6)Child(): An output is the frame-id which indicates the child frame-id. Then the Current-frame's frame-id is changed to the child's one.

(7)First-slot(Slot): An output is a frame-id whose frame instance contains the slot item indicated by the argument Slot. Then Current-slot stores with the obtained frame's frame-id.

(8)Next-slot(Slot): An output is the frame-id which consists a slot item Slot. The distance between the frame-id of Current-slot and an obtained frame-id is shortest, such that @F(fc)<<@F(fn) and fn contains a slot item Slot. Then the Current-slot stores the obtained frame-id.

(9)Set(Frame)/Set(Slot Frame) The value of the register Current-frame or Current-slot is specified by the argument Frame of operation. And, each register's value becomes a frame-id for a frame specified by the argument Frame.

The operations (7), (8) and Set(Slot, Frame) are redundant in the operations. Because, these operations could be defined by other operations. However, these operations is useful to build a logical interface, and the set of operations dose not become ambiguous. Hence, they are defined as the operations.

We show an access program written in virtual operations to the virtual frame data structure. For example, the expression S(*F Inst) is given, then the formula is satisfied with a certain value which is obtained through finding a value of the variable *F, where S is a slot item for a slot and Inst a slot value. The program which achieves this purpose is as follows.

```
[Example]
   (first-slot s)
N: (cond ((null (current-slot)) nil)))
   (cond ((equal (slotval s
                           (current-slot))
                 Inst))
          (return (current-slot)))
   (next-slot s)
   (goto N)
```

3.2.3. Constraints

A set of constraints CC is ones for the frame data structure $$ defined in Sec. 3. Those constraints are non-logical axioms in a formal frame system based on a first-order logic. The formal axioms are given in the next section.

4. Conceptual Frame System

Already, a formal frame model FM is given in a previous section. In this section, we define a formal frame system FS based on a first-order logic. FS is given by the quadruple <S, LS, AS, RS>. Where, S, LS, AS, and RS are a frame data structure schema, a frame system description language, a set of axioms and a set of inference rules, resectively. A frame system description language LS is the first-order language.

A frame structure schema S had been given in Sec. 3.

The first-order language LS for describing the schema S is called a conceptual frame system description language. LS is given by the pair (A, WS), where A is a set of alphabets, WS a set of well formed formulas (wffs). A includes, constants, variables, function symbols, logical symbols (and, or, not, ->, <->, =, Exist, ∀), and predicate symbols.

There are four kinds of predicate symbols, i.e., ST, SP, FP, RP. The predicate symbols include unary predicate symbols ST, two-place predicate symbols SP, n-place predicate symbols FP and two-place predicate symbols RP. Each kind of predicate symbols, i.e., ST, SP, FP, RP, is called a slot item predicate symbol, a slot predicate symbol, a frame predicate symbol and a relationship predicate symbol. For every slot item predicate symbol ST, slot predicate symbol SP, frame predicate symbol FP and relationship predicate RP, there are ST predicate symbol Sti, SP predicate symbol Si, FP predicate symbol Fi and relationship PR predicate symbol Ri, respectively.

For such a set of symbols AS, terms and wffs are defined in the same as [Enderton].

An interpretation for LS is given a pair <D, M>, where D is a universe, M a mapping from symbol sets called an interpretation function. For every symbol in A, M is used for acquiring denotations of the symbols.

[Definition] M
(i) for a constant a,
  $M(a) \in D$.
(ii) for n-place function fn,
  $M(fn): D \times D \times \ldots \times D \to D$.
(iii) for n-place predicate Pn,
  $M(Pn) \in D \times D \times \ldots \times D$.
(iv) for FP predicate symbol Fi,
  $M(Fi) \in \{f\}$; f is a frame instance.
(v) for RP predicate symbol Ri,
  $M(Ri) \in \{r\}$; r is a link instance.
(vi) for SP predicate symbol Si,
  $M(Si) \in \{@F\} \times D$.
(vii) for every ST predicate symbol Sti,
  $M(Sti) \subseteq D$.

AS is a set of axioms written in LS. AS is composed of logical axioms, equality axioms and non-logical axioms. There are the following non-logical axioms.

(1) For every frame schema F,

(∀k∀x1∀x2 ...∀xn F(k, x1, x2, ... , xn)->

@F(k) and s1(k,x1) and ... and sn(k,xn)).

where, @F(k) indicated that frame-id is k.

(∀k∀x1∀x2 ...∀xn F(k, x1, x2, ... , xn)->

@F(k) and St1(x1) and ... and Stn(xn)).

(2) For every frame schema,

(∀k∀x1∀x2 ...∀xn F(k, x1, x2, ... xn)

and F(k, y1, y2, ... yn)->

and i=1,...,n (xi=yi))

(3) For every relation schema,

(∀f1∀f2 R(f1, f2)->F1(f1) and F2(f2)).

where, fi is a frame instance, fi∈FF

(4) For every type of AKO,

(∀f1∀f2∀f3 R(f1,f2) and R(f3,f2)->f1=f3).

(∀f1∀f2 R(f1,f2)->Q(f1)=Q(f2)).

The model of LS is an interpretation which satisfies a set of non-logical axioms described above.

There are inference rules which are modus ponens and generalization rule.

By above definitions, we can obtain next proposition for FS.

[Proposition] FS is complete and sound.

[Proof] FS is first-order theory. []

The non-logical axiom (4) describes as follows. A frame is given by (F, F). Fig. 7 shows this axiom. Let sij be a slot item in a frame schema Fi. In Fig. 7, the frame schema F1 and F2 is represented by a sequence of slot items except for a frame-id item, an A-Kind-Of slot and a Frame-type slot. For F1 and F2, the sequence of slot items.

F1:<s1Ø,s11, ... , s1n>
F2:<s2Ø,s21, ... , s2n, ... ,s2m> (m≥n).

This sequence is called a slot item sequence (SIS). If a SIS is given, the sequence of slot values obtained. This sequence is called a slot value sequence (SVS). Let Q be a function to get a slot items for a frame instance except for a slot-id, an A-Kind-Of slot and a frametype slot. Q(f1) and Q(f2) are as follows.

Q(f1)={v1Ø,v11, ... v1n}
Q(f2)={v2Ø,v21, ... v2n, ... v2m}

Therefor, properties which are inherited from its parent frame are {vØ, v1, v2, v3, ... , vn}. Hence, axiom (4) describes most restricted function of a property inheritance. That is, the corresponding SVS in the FIG must be equal. Therefor, in this formalization, if a hierarchical representation have exceptions, they must be enumerated.

5. Usage of a Database System

It is possible to expect that the application area of the knowledge-based systems are extended. In order to treat this problem, we can consider one of the treatments that a knowledge-based system uses a database system. Then it is necessary to transform a data model in a database into a knowledge base and the conversion of data representation between them. In our work, a relational view of the frame data model are defined in FS. It is possible to consider that the join between the relational database[Codd] and the frame-based knowledge-based system. Therefor, since users do not take special care to a data structure, they handle them easily. Fig. 8 shows an overview of the combination of two types of the models. Then, FS becomes an interface between a frame model and relational data model. It is available to store factual knowledge defined in a frame model using a relational database system.

6. Concluding Remarks

In this paper, we showed a first-order theory of logical part of the frame-based knowledge representation system, and made clear the semantics of the retrieval operations in it based on the navigational manner. Then, the limitation of our approach are; 1) the assumption that frame stores the factual knowledge only, 2) the restriction on the inherited properties, etc. Based on this formal system, next, we show the logical semantics of the operations to update of a frame data model, and mechanisms of message passing.

Acknowledgements

The authors would like to thank all members, especially Y. Kawashima and N. Sanai at JIPDEC who gave us the chance to write this report.

References

[Attardi] Attardi, G., Simi, M.; Consistency and Completeness of OMEGA, a Logic for Knowledge Representation. Proc. IJCAI 81, 1981
[Bobrow] Bobrow, D., Stefik, M.; The LOOPS Manual. Xerox Paloalto Research Center, 1981
[Brachman] Brachman, R.; On the Epistemological Status of Semantic Networks. In Findler, N., ed. Associative Networks, Academic Press. 1979
[Brachman85a] Brachman, R., Gilbert, V., Levesque, H.; An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of Krypton. Proc. IJCAI 85, 1985
[Brachman85b] Brachman, R., Schmolze, J.; An Overview of the KL-ONE knowledge representation System. Cognitive Science, 9(2), 1985
[Clocksin] Clocksin, W., Mellish, C.; Programming in Prolog. 2nd ed. Springer-Verlag, 1984
[Codd] Codd, E.; A Relational Model of Data for Large Shared Data Bank. CACM, Vol. 13, No. 6, 1970
[Date] Date, C.; Introduction to Database Systems. Addison-Wesley, 1981
[Davis] Davis, R., King, J.; An Overview of Production System. Elcock, E., eds. Machine Intelligence, Vol. 8, John-Wiley and Sons, 1976
[Dayal] Dayal, U., Berstain, D.; On the Updatability of Network Views - Extending Relational View. Theory to The Network Model. Inform. Systems, Vol. 7, No. 1, 1982
[Enderton] Enderton, H.; Mathematical Introduction to Logic. Academic Press, 1972
[Frish] Frish, M., Allen, F.; Knowledge Retrieval as Limited Inference, Proc. 6th Conference on Automated Deduction, 1982
[Hayes] Hayes, P.; The Logic of Frames. In Webber, B., Nilsson, N., ed Reading in Artificial Intelligence, Tioga, 1979
[Hayes-Roth] Hayes-Roth, F., Waterman, D., Lenat, D.; Building Expert Systems. Addison-Wesley, 1983
[Iri] Iri, M., et als.; Graph Theory with Exercises. Corona Pub. Co. 1983 (in Japanese)
[Israel] Israel, D., Brachman, R.; Some Remarks on the Semantics of Representation Language. In Brodie, M., et als. ed. On Conceptual Modeling, Springer-Verlag 1984
[Ito83] Ito, H., Ueno, U.; Design and Implementation of a Frame Based Knowledge Representation Language. Research Activities,

Tokyo Denki University, Vol. 5, 1983 (in Japanese)

[Ito86] Ito, H., Ueno, H.; ZERO: Frame + Prolog. To appear in LPC'85, LNCS, Springer-Verlag, 1986

[Jacob] Jacob, B.; On Database Logic. JACM, Vol. 29, No. 2, 1982

[Kowalski] Kowalski, R.; Logic for Problem Solving. North-Holland, 1979

[Minsky] Minsky, M.; A Framework for representing Knowledge. In Winston, P., ed. The Psychology of Computer Vision, McGrow-Hill, 1975

[Mylopoulos] Mylopoulos, J., Levesque, H.; An overview of Knowledge representation. In Brodie, M., et als. ed. On Conceptual Modeling, Springer-Verlag, 1984

[Olle] Olle, T.; The Codasyl Approach to Data Base Management. John-Wiley and Sons, 1978

[Reimer] Reimer, U., Hahn, U.; A Formal Approach to the semantics of a Frame Data Model. Proc. IJCAI 83, 1983

[Reiter] Reiter, R.; Towards a Logical Reconstruction of Relational database Theory. In Brodie, M., et als. ed. On Conceptual Modeling, Springer-Verlag, 1984

[Stefik] Stefik, M.; An Examination of a Frame-Structured Representation System. Proc. IJCAI 79, 1979

[Stefik] Stefik, M.; Planning with Constraints(MOLGEN part 1). Artificial Intelligence 16, 1981

[Tsichritzis] Tsichritzis, D., Lochovsky, F.; Data Models. Prentice-Hall, 1982

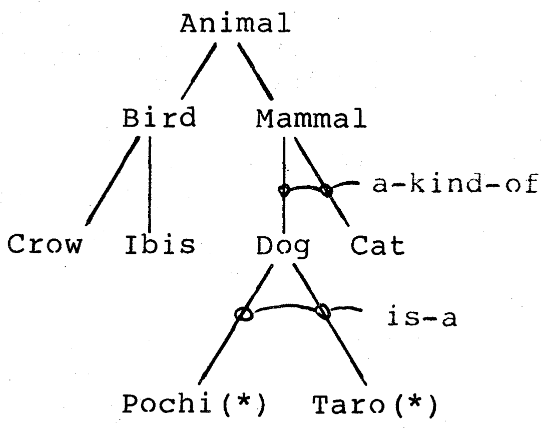
[Ueno83] Ueno, H.; An End-User Oriented Language to Develop Knowledge-Base Expert System. Compcon 83 Fall, 1983

[Ueno84] Ueno, H.; An Intelligent Programming Assistant System INTELLITUTOR - Background and Philosophy -. Knowledge Engineering and Artificial Intelligence 37-5, Japan Information Processing Society (in Japanese)

[Vilain] Vilain, M.,; The Restricted Language Architecture of a Hibrid Representation System, Proc. IJCAI 85, 1985

[Waterman] Waterman, D.; A Guide to Expert Systems. Addison Wesley, 1986

[Weiner] Weiner, J., Palmer, M.; The Design of a System for Designing Knowledge Representation System. Proc. IJCAI 81, 1981



(a) An example of a generalization hierarchy.

Frame-name	Pochi
Frame-type	Instance
Is-a	Dog
Father	Taro
Mother	Shiro
Color	white
.	
.	

(b) An example of a frame.

Fig.1 An example of a hierarchy and a frame.

```
color-of(Pochi White)
is-a(Pochi Dog)
a-kind-of(Dog Memmal)
is-a(Pochi Dog) and a-kind-of(dog Memmal)->
        is-a(Pochi Mammal)
Father(Pochi Taro)
a-kind-of(Crow Bird) and a-kind-of(Bird Animal)->
        a-kind-of(Crow Animal)
       .etc
```

Fig.2 A set of Sentences to Fig. 1.

1) A physical frame system.
2) A virtual frame system.
3) A conceptual frame system.
4) An application frame system.
5) A knowledge-based system.

Fig.3 An organization for the formalization of a frame-based knowledge-based system.

F2

AKO	F1
A	a
B	b
C	c
D	d

(a) An actual frame.

Frame-name	AKO	A	B	C	D
F2	F1	a	b	c	d

(b) A frame structure for formalization.

Fig.4 A basic frame structure.

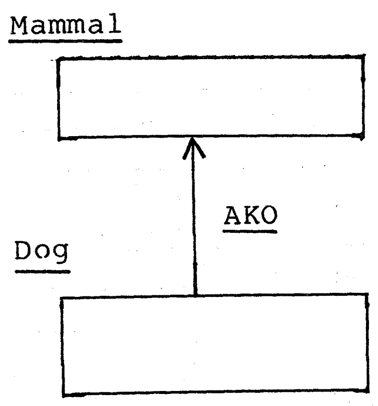


Fig. 5 An example of a frame schema graph.

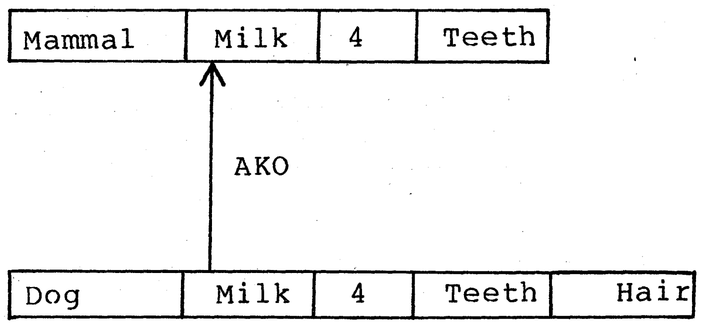


Fig. 6 An example of a frame instance graph.

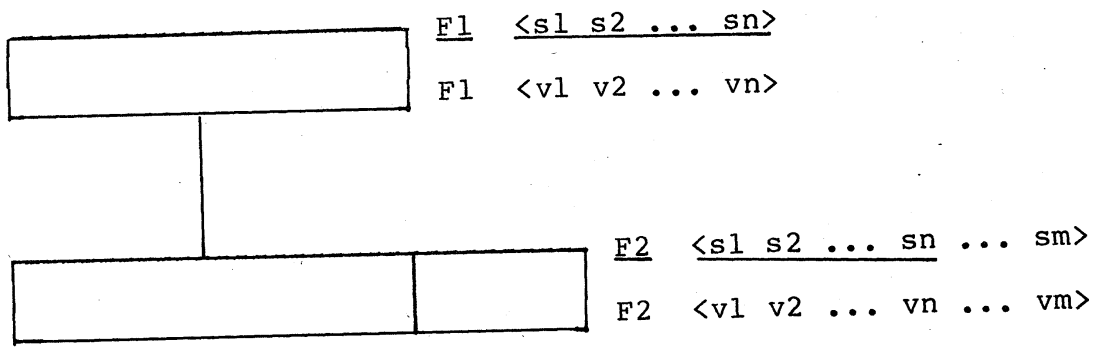


Fig. 7 An overview of the inheritance mechanism.

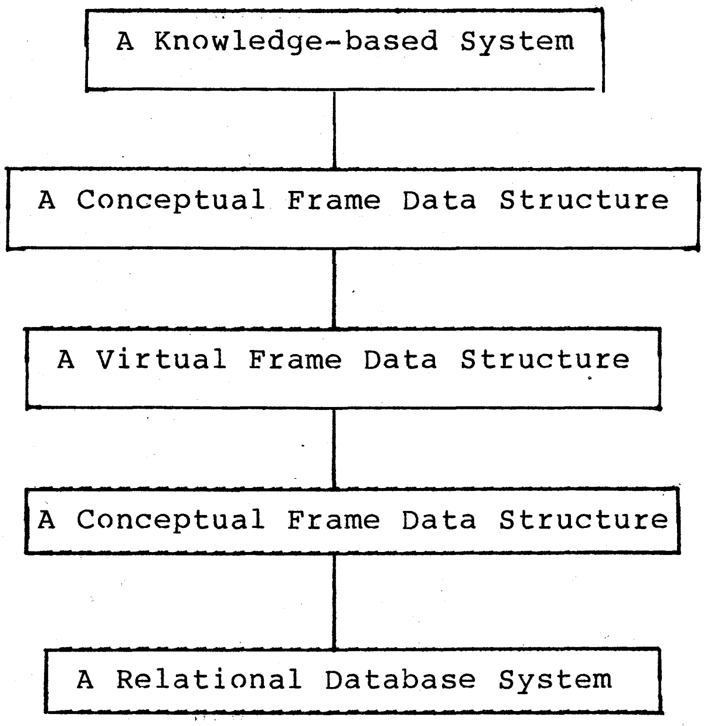

A Knowledge-based System
A Conceptual Frame Data Structure
A Virtual Frame Data Structure
A Conceptual Frame Data Structure
A Relational Database System

Fig. 8 A usage of a relational database system.