Acta Mathematica Academiae Paedagogicae Nyíregyháziensis
18 (2002), 77–84
www.emis.de/journals

STRUCTURE REPRESENTATION IN OBJECT ORIENTED KNOWLEDGE REPRESENTATION SYSTEMS

KATALIN BOGNÁR

ABSTRACT. This paper deals with structure representation in object oriented knowledge representation systems. Structures can appear in various forms - spatial, composites, text, etc. - and can be used in a lot of applications. In this paper we offer a kind of representation for structures based on a formalism of pattern recognition. A reasoning method allowing manipulating the structures is defined which is based on the subsumption relation and classification.

1. Introduction

In a great number of problems it is indispensable to represent and manipulate structures with various proprieties. For example, the goal of the representation and classification of landscape structures is to make diagnosis and forecast on agriculture ([Le Ber]); in organic chemistry the representation and manipulation of molecular structure have the reason to make plans of the molecular syntheses ([Lieber]); in data mining the representation and manipulation of structured or semi-structured text documents have the goal to index the text by its content.

General problems on structures arise in:

- **representation:** to represent entities having physical reality and intern organization or structure: molecules, grounds, text documents, etc.
- **reasoning:** recognize and classify structures to solve problems on the investigated domain.

These structures can be concerned on abstract level as non (necessarily) oriented graphs, which consist of arcs and nodes. Relationships among arcs specify the environment of structures or inter structural constraints.

By author's knowledge there is no language general enough to offer the representation and manipulation of structures in the various domains. In pattern recognition the structure description has been studied in detail and a formal and practical frame was suggested, which is next to the approach of this paper.

In this paper a theoretical formalism is described which can be realized in practice, and which is based on some results in pattern recognition. This formalism is adapted to object oriented knowledge representation (OOKR later on).

The paper is organized as follows: at the beginning the formalism to represent and manipulate structures in KADS is shown, then the notion of extended structural description is given. In the next section the hierarchy of structural descriptions is

2000 *Mathematics Subject Classification.* 68T30.
Key words and phrases. Object oriented representation, structures, structure description, reasoning by classification.
This publication made in the frame of Hungarian-French scientific and technological cooperation supported by Hungarian OMFB and the French Ministère des Affaires Etrangères (Balaton F-4/98).

defined to show a reasoning method based on classification. The paper ends with a short conclusion.

2. Modeling and description of structures

2.1. **Structures in the formalism of KADS.** In [Breuker] the structure S is defined by the set of instanced relations, that is

$$S = \{r_1(x_1^1, \dots, x_n^1), \dots, r_m(x_1^m, \dots, x_k^m)\},$$

where r_i is the name of relation and x_i^j are elementary objects.

The list and graphs are special relations:

- The list $l = (x_1, x_2, \dots, x_n)$ can be seen as the structure
$$\{s(x_1, x_2), s(x_2, x_3), \dots, s(x_{n-1}, x_n)\}$$
where $s(x_i, x_{i+1})$ denotes the fact that x_{i+1} is the successor of x_i in the list l (all the relations r_i are defined by successor relation s).
- The graph $G = (V, E)$ can be considered as the structure
$$\{E(v_1, v_2), E(v_2, v_3) \dots, E(v_{n-1}, v_n)\}$$
where $V = \{v_1, v_2, \dots, v_n\}$ is the set of nodes and
$$E = \{(v_1, v_1\prime), (v_2, v_2\prime), \dots, (v_n, v_n\prime)\}$$
is the set of edges.

The elementary operations defined on the structures are composition, decomposition, transformation and selection (object or relation). The composition constructs the structure from the set of elementary objects. The decomposition is an opposite operation of the composition which consists in finding elements of the structure. The transformation allows changing a structure to another one without modifying elements and relations (sort) or modifying relations (pass from a linear structure to a hierarchic one). The selection operation of elementary objects or relationships consists of verifying the existence of an elementary object or relationship.

2.2. **Structure description in pattern recognition.** Structure description (SD) was investigated in pattern recognition to recognize and match objects. SD-s can be considered as a set of primitives connected with one another. They may be used to model a lot of forms of physical structures.

Definition 1. The *primitive of the concept* C is defined by the couple (attribute: value).

Definition 2. The *primitive part of the concept* C is defined by a set of primitives, that is a set of

$$\{(\text{attribute}_1 : \text{value}_1), (\text{attribute}_2 : \text{value}_2), \dots, (\text{attribute}_k : \text{value}_k)\}, \quad (k \geq 1).$$

Definition 3. The *ordinary structure description* of the concept C is the pair $D = \{P, R\}$ where

- $P = \{P_1, P_2, \dots, P_n\}$ and each P_i $(i = 1, 2, \dots, n)$ is a primitive part of concept C.
- $R = \{PR_1, PR_2, \dots, PR_m\}$ is a set of relations between the primitive parts such that in $PR_i = (NR_i, R_i)$ NR_i is the name of relationship and R_i is the primitive part, that is $R_i \subseteq P^l$ for an integer l.

$D = \{P, R\}$
$P = \{P_1, P_2, P_3, P_4\}$
$P_1 = \{(\text{form: cube}), (\text{colour: black})\}$
$P_2 = \{(\text{form: cube}), (\text{colour: white})\}$
$P_3 = \{(\text{form: sphere}), (\text{colour: white})\}$
$P_4 = \{(\text{form: pyramid}), (\text{colour: black})\}$
$R = \{PR_1, PR_2\}$
$PR_1 = \{(\text{on}, R_1)\}$
$R_1 = \{(P_2, P_1), (P_3, P_2)\}$
$PR_2 = \{(\text{left}, R_2)\}$
$R_2 = \{(P_4, P_1)\}$

FIGURE 1. Example of ordinary SD.

For example, let us consider ([Napoli]) an universe consisting of: P_1 black cube, P_2 white cube, P_3 white sphere, P_4 black pyramid, and the construction where P_3 is on P_2, P_2 is on P_1, and P_4 is on the left hand side of P_1. The ordinary structure description representing this construction can be seen in the Figure 1. The four primitive parts (P_1, P_2, P_3, P_4) describe two pairs of attributes, where the attributes are the form and the colour. The relationships among the primitive parts are "on" and "left". The set R describes the configuration of the ordinary SD.

The ordinary structure description can be considered as a composite object, where the components represent the primitive parts and the relationships among components represent the relations among primitive parts. In pattern recognition the ordinary structure description is used to guide the recognition process. The base of a recognition system contains some ordinary SD of prototype objects, which can serve as a model during the recognition. The system analyses the description of candidate building its ordinary SD and comparing it with the model. The matching process is oriented in the meaning that it relies the primitive parts and relations associated to the model with the set of primitive parts and set of relations associated to the candidate. So the matching considers the primitive parts and relations at the same time.

- Let us consider P and Q as the set of primitive parts and $h\colon P \to Q$ a map from primitive parts of P to primitive parts of Q. The primitive part Q_j of the candidate matches to the primitive part P_i of prototype P, if the set of attribute-value pair in P_i is a subset of attribute-value pairs of Q_j.
- The matching between the candidate and prototype relation is based on the notion of relational homomorphism, which put into S the tuples of R_i relations. The relational homomorphism from R to S is a map $h\colon P \to Q$ satisfying $R \circ h \subseteq S$, where $S \subseteq Q^n$ and

$$R \circ h = \{(Q_1, Q_2, \ldots, Q_n) \in Q^n | \exists (P_1, P_2, \ldots, P_n) \in R : h(P_i) = Q_i, i = 1, 2, \ldots, n\}.$$

Definition 4. Let us consider $D_p = \{P, R\}$ as an ordinary SD—the prototype—where $P = \{P_1, P_2, \ldots, P_n\}$ and $R = \{(NR_1, R_1), (NR_2, R_2), \ldots, (NR_k, R_k)\}$. Furthermore $D_c = \{Q, S\}$ as an ordinary SD—the candidate—where

$$Q = \{Q_1, Q_2, \ldots, Q_m\}$$

and $S = \{(NS_1, S_1), (NS_2, S_2), \ldots, (NS_k, S_k)\}$. We say that there exists an *exact matching* from D_p to D_c (or D_c matches to D_p) if there exists a map $h\colon P \to Q$ satisfying

(1) $h(P_i) = Q_j$ implies that $P_i \subseteq Q_j$
(2) $NR_i = NS_j$ implies that $R_i \circ h \subseteq S_j$.

The consequence of the definition shows that if one of the relations R_i in D_p has the same name as one of the relations S_j in D_c then h (mapping prototype primitives D_p to candidate primitives D_c) has to be a relational homomorphism from R_i to S_j. In R and S are the same number of relations.

2.3. **Generalization of the ordinary structure description.** The formalization presented above gives us the frame to define the structure formalization on the domain of OOKR, generalizing the definition of structure description to typed SD and generalized SD.

2.3.1. *Typed structure description.* The first extension of SD introduces the typed SD allowing attribute-type instead of attribute-value pair. Let τ denote the set of primitive types of the system. Thus $P_i \subseteq (A \times \tau)^k$ where A is the set of attribute names.

Definition 5. The *primitive type of concept* C is defined by the set

$$\{(\text{attribute}_1 : \text{type}_1), (\text{attribute}_2 : \text{type}_2), \ldots, (\text{attribute}_k : \text{type}_k)\}, \quad (k \geq 1).$$

Definition 6. The *typed structure description* of concept C is the pair $D = \{P, R\}$ where

- $P = \{P_1, P_2, \ldots, P_n\}$ and each P_i $(i = 1, 2, \ldots, n)$ is a primitive type of concept C.
- $R = \{PR_1, PR_2, \ldots, PR_m\}$ is a set of relations between the primitive types such that in $PR_i = (NR_i, R_i)$ NR_i is the name of relationship and R_i is the primitive type, that is $R_i \subseteq P^l$ for an integer l.

We can establish a parallelism between the notions of the ordinary SD - typed SD and instances-classes. The typed SD corresponds to the notion of classes while ordinary SD corresponds to the notion of instances.

The matching between typed SD-s can be defined as the generalization of matching between ordinary SD-s. The typed SD $D_c = \{Q, S\}$ matches to the typed SD $D_p = \{P, R\}$, if there exists a map $h \colon P \to Q$, that is $h(P_i) = Q_j$ implies that Q_j is a subtype of P_i.

Definition 7. Let us consider a typed SD prototype $D_p = \{P, R\}$ where $P = \{P_1, P_2, \ldots, P_n\}$ and each P_i $(i = 1, 2, \ldots, n)$ is a primitive type. Furthermore

$$R = \{(NR_1, R_1), (NR_2, R_2), \ldots, (NR_k, R_k)\}$$

is a set of relations existing between the primitive types. Similarly, $D_c = \{Q, S\}$ is a typed SD candidate where

$$Q = \{Q_1, Q_2, \ldots, Q_m\}$$

is a set of primitive types and

$$S = \{(NS_1, S_1), (NS_2, S_2), \ldots, (NS_k, S_k)\}$$

is a set of relations between the primitive types. We say, that there exists an *exact matching* from D_p to D_c, if there exists a map $h \colon P \to Q$ such that

(1) $h(P_i) = Q_j$ implies that Q_j is a subtype of P_i,
(2) $NR_i = NS_j$ implies that $R_i \circ h \subseteq S_j$ considering types, that is the type of elements of S_j is a subtype of the type of corresponding elements of R_i.

2.3.2. *Generalized structure description.* If we want to represent a large number of various structures by the SD formalism, we have to extend the definition to allow recursivity, that is an SD might be defined with respect to other SD's. Intuitively, the generalized structure description (GSD in future) is a structure description where the primitive part itself can be a structure description, so the set of primitive parts becomes a set of substructures.

Definition 8. The *generalized structure description* of concept C is defined by the pair

$$D = \{P, R\},$$

where

- The set $P = \{P_1, P_2, \ldots, P_n\}$ contains the substructures of concept C such that each P_i (generalized primitive part) is either primitive part or primitive type (or their combination), or a generalized structure description.
- The set $R = \{PR_1, PR_2, \ldots, PR_k\}$ represents the relations among substructures, that is it is a set of relations between primitive parts such that in $PR_i = (NR_i, R_i)$ NR_i is the name of relation and R_i denotes generalized primitive part in the relation.

It is obvious, that typed structure description is itself a generalized structure description.

Definition 9. The GSD $D_c = \{Q, S\}$ *matches* to the GSD $D_p = \{P, R\}$, if there exists a map

$$h \colon P \to Q$$

such that

a.) If Q_j and P_i are primitive types (class to class match), then
  - $h(P_i) = Q_j$ implies that Q_j is a subtype of P_i.
  - $NR_i = NS_j$ implies that $R_i \circ h \subseteq S_j$ considering types, that is type of elements of S_j is a subtype of the type of corresponding elements of R_i.

b.) If Q_j is a primitive part and P_i is a primitive type (instance-class)
  - $h(P_i) = Q_j$ implies that Q_j is an instance of the type P_i,
  - $NR_i = NS_j$ implies that $R_i \circ h \subseteq S_j$ considering types, that is the elements of S_j are instances of the corresponding elements of R_i.

c.) If Q_j and P_i are primitive parts (instance-instance)
  - $h(P_i) = Q_j$ implies that $P_i \subseteq Q_j$,
  - $NR_i = NS_j$ implies that $R_i \circ h \subseteq S_j$.

2.4. **An example of the generalized structure description.** Let us suppose that an auto consists of car-body, motor and four wheels. The car-body consists of 3 doors and bonnet, and the wheel consists of tyre and rim. The auto is represented by the GSD in Figure 2, where

$$PR_1 = \{(\text{composition}, R_1)\}$$

and $R_1 = \{(P_2, P_1)\}$ means that P_2 is a component of the substructure of P_1. The composed structure is represented by a GSD, where we can see components and configuration of these components. The name of the component comes from its attribute and it is possible that more components have the same type (e.x. 4 wheels). In this case the number of components is given by the number of attributes. The name of the component corresponds to the root of the hierarchy of composition which is attached to this component.

$D = \{P, R\}$
$P = \{P_1, P_2, D', D"\}$
$P_1 = \{(\text{name: Auto})\}$
$P_2 = \{(\text{name: Motor}), (\text{number-of-cylinders: } 4)\}$

$D' = \{P', R'\}$
$P' = \{P_3, P_4, P_5\}$
$P_3 = \{(\text{name: Car-body})\}$
$P_4 = \{(\text{name: Bonnet}), (\text{colour: Colour})\}$
$P_5 = \{(\text{name: Door}), (\text{number: } 3), (\text{colour: Colour})\}$
$R' = \{PR_2\}$
$PR_2 = \{(\text{composition}, R_2)\}$
$R_2 = \{(P_4, P_3), (P_5, P_3)\}$

$D" = \{P", R"\}$
$P" = \{P_6, P_7, P_8\}$
$P_6 = \{(\text{name: Wheel}), (\text{number: } 4)\}$
$P_7 = \{(\text{name: Tyre}), (\text{dimension: Number}), (\text{material: Rubber})\}$
$P_8 = \{(\text{name: Rim}), (\text{dimension: Number}), (\text{material: Steel})\}$
$R" = \{PR_3, PR_4\}$
$PR_3 = \{(\text{composition}, R_3)\}$
$R_2 = \{(P_7, P_6), (P_8, P_6)\}$
$PR_4 = \{(\text{under}, R_4)\}$
$R_4 = \{(P_6, P_3)\}$

$R = \{PR_1\}$
$PR_1 = \{(\text{composition}, R_1)\}$
$R_1 = \{(P_2, P_1), (D', P_1), (D", P_1)\}$

FIGURE 2. Generalized Structure Description.

3. STRUCTURE DESCRIPTIONS IN OOKR SYSTEMS

In this section we introduce briefly the object oriented knowledge representation (OOKR) systems, then the representation of GSD in such a frame. Later on the subsumption relation will be defined in order to organize GSD's into a hierarchy and realize a classification to build structures and to reason on theses structures.

3.1. **Brief Introduction to OOKR.** The OOKR system is based on a class-hierarchy $\mathcal{H} = (\chi, \tau, \sqsubseteq)$, where the subsumption relation is defined on the classes in χ and τ is the root of hierarchy (the top element regarding the subsumption). Any class represents a concept of the given domain and consists of the set of attributes. These attributes describe properties and behaviors of the represented concept. The class can be instantiated. Furthermore, the class can be described by conjunctions: $C = (a_1, s_1) \sqcap (a_2, s_2) \ldots, \sqcap (a_n, s_n)$ where a_i denote attributes (properties of the concept, they are pairwise different) and s_k denotes the specification attached to the attributes and they precise for example, type, domain, cardinality of values of the attribute. The class C specializes the class D by adding new attributes to already defined attributes. If we want to determine if D subsumes C, we have to verify the existence of attribute a_k in C (that is with the same name) for each attribute a_k in D and verify associated specifications.

At the end, the instance i of class C is denoted by

$$i = (a_1, v_1) \sqcap (a_2, v_2) \ldots, \sqcap (a_n, v_n),$$

where v_k is the value associated to a_k in the instance i (eventually it is possible that $v_k = 0$).

3.2. **Structural subsumption.** In this section the notion of subsumption relation between the classes of structures will be defined. Such notation allows us to organize structures into hierarchy and to apply the reasoning based on classification on this hierarchy.

Definition 10. Let us consider classes A and B representing the GSD

$$D_A = \{P_A, R_A\}$$

and

$$D_B = \{P_B, R_B\}.$$

We say, that class A *subsumes* the class B (or B is subsumed by A) if D_B matches D_A (class to class match). Class A is said to be *subsumer* and class B may be refered to as *subsumed.*

Notation: $B \sqsubseteq A$.

The main properties of structural subsumption are the following:

- Reflexivity, that is $A \sqsubseteq A$.
- Anti-symmetry, that is if $B \sqsubseteq A$ and $A \sqsubseteq B$, then there exists an exact matching from A to B, that is the generalized structural descriptions are identical not considering isomorphism.
- Transitivity, that is if $B \sqsubseteq A$ and $A \sqsubseteq C$, then $B \sqsubseteq C$ (considering the composition of mappings in exact matching).

Consequently, the structural subsumption is an ordering relation and organizes GSD's into hierarchy. This hierarchy will be noted as $\mathcal{H}_{SD}$ in the future.

3.3. **Reasoning on structural descriptions.**

- The goal of classification of classes is to insert a new GSD (class X) into $\mathcal{H}_{SD}$ hierarchy. This implies seeking the most specific subsumers and most general subsumed of X.
- The classification of instance x (ordinary SD) consists of finding the GSD's, whose instance x may be.
- The partition of properties in $\mathcal{H}_{SD}$ (inheritance of properties) consists of finding which properties can be inherited by GSD in $\mathcal{H}_{SD}$. Especially, the GSD of concept X inherits a property "prop" (for example an attribute-specification pair), if $X \sqsubseteq Y$ where "prop" is attached to the concept Y.
- The selection of an object or relation in the structure leads to the classification in the following way: the element x (object or relation) and a GSD of class C are given. The question if x belongs to the class C means to find or construct a class(x) class around x, then classify class(x) into $\mathcal{H}_{SD}$.

4. Conclusion

We here established a common formalism in pattern recognition and the object oriented knowledge representation. The presented approach can be adapted and used in a great number of problems in artificial intelligence. The formalism above mentioned based on results in description logics ([Bognar]) and was inspired by requirements in measurement of structures' similarity in the course of the case-based reasoning ([Al Hulou]).

References

[Le Ber] Le Ber, F., Mangelinck, L., Napoli, A., Un treillis de relations pour la classification de structures spatiales, *Actes des Septièmes Rencontres de la Société Francophone de Classification, INRIA Nancy (1999), 61-67*

[Bognar] Bognár, K., Leíró logikák az ismeretábrázolásban, *Alkalmazott Matematikai Lapok, MTA, Budapest, 20 No.2 (2000), 183-193*

[Breuker] Breuker, J., Van de Velde, W., eds. CommonKADS Library for Expertise Modelling, *IOS Press, Amsterdam, (1994)*

[Haralick] Haralick, R.M., Shapiro, L.G., Computer and Robot Vision, *Addison Wesley, Reading, Massachusetts (1993)*

[Al Hulou] Al Hulou, R., Napoli, A., Nauer, E., Objets semi-structurés, classes polythétiques et classification, *Actes des Septièmes Rencontres de la Société Francophone de Classification, INRIA Nancy (1999), 379-384*

[Lieber] Lieber, J., Napoli, A., Using classification in case-based planning, *Proceedings of ECAI'96 Budapest, John Wiley Sons, UK, (1996), 132-136*

[Napoli] Napoli, A., Le Ber, F., Bognár, K., Une proposition pour la représentation de structures dans un système de représentation de connaissances par objet, *Actes de Conférences IC'2000 Toulouse, (2000), 145-152*

[Shapiro] Shapiro, L.G., Haralick, R.M., Structural Descriptions and Inexact Matching, *IEEE Transaction on Pattern Analysis and Machine Intelligence (1981), 504-519*

[Vosselman] Vosselman, G., Relational Matching, *Lecture Notes in Computer Science 628, Springer Verlag, Berlin (1992)*

Received March 29, 2002; July 7, 2002 revised.

Institute of Mathematics and Computer Science,
University of Debrecen,
4010 Debrecen, Pf. 12, Hungary
E-mail address: bognar@math.klte.hu