PRACE IPI PAN • ICS PAS REPORTS

Zdzisław Pawlak

Distributed information systems

370
1979
WARSZAWA

INSTYTUT PODSTAW INFORMATYKI POLSKIEJ AKADEMII NAUK
INSTITUTE OF COMPUTER SCIENCE POLISH ACADEMY OF SCIENCES
00-901 WARSAW, P. O. Box 22, POLAND

Zdzisław Pawlak

DISTRIBUTED INFORMATION SYSTEMS

370

Warsaw 1979

Pracę zgłosił Andrzej Blikle

Mailing address: Prof. dr hab. Zdzisław Pawlak
Institute of Computer Science PAS
P.O. Box 22
00-901 Warsaw

P r i n t e d a s a m a n u s c r i p t

N a p r a w a c h r ę k o p i s u

Nakład 700 egz. Ark. wyd. 1,20; ark. druk. 1,75. Papier offset. kl. III, 70 g, 70 x 100. Oddano do druku w lipcu 1979 r. W. D. N. Zam. nr 449/0/79

Abstract .Содержание . Streszczenie

In this paper the following problem is considered: Given n "local" information systems $s_1,\ldots,s_n$, with corresponding query languages $L_1,\ldots,L_n$. Find an answer to the query in a "central" language $L = UL_i$ as a function of local answers (i.e. answers of local systems). Store organization related to this problem is also briefly discussed.

Распределенные информационные системы

В работе рассматривается проблема, каким образом найти ответ в распределенных информационных системах на основе получения частичных ответов из локальных систем. Кратко обсуждается организация памяти, связана с этой проблемой.

Rozproszone systemy informacyjne

W pracy rozważa się problem, w jaki sposób znaleźć odpowiedź w rozproszonych systemach informacyjnych na podstawie uzyskiwania odpowiedzi częściowych z systemów lokalnych. Dyskutowana jest krótko również organizacja pamięci związana z tym problemem.

We shall deal in this note with the following problem:

We are given n local information systems $S_1, S_2, \ldots, S_n$. With each system S_i there is associated an information language L_{S_i} (or shortly L_i). The language L_i is used to ask queries about some informations contained in the system S_i. If t is a query in L_i, then the answer to this query will be denoted by $\sigma_{S_i}(t)$ (or simply by $\sigma_i(t)$).

We may combine systems $S_1, S_2, \ldots, S_n$ in one system, which we shall denote by S, and call global information system. A global information system S consisting of local systems S_i may be assumed to be for example a net of the systems S_i. With the global system S we may associate a global language L_S (or in short L). The global language L may be viewed as a certain combination of the languages L_i (local languages).

The question arises whether we can obtain an answer to the query in global language (global query), by means of combining the local answers to local queries, that is whether the answer to the query t may be presented in the form

$$\sigma_S(t) = \varphi(\sigma_{S_1}(t_1), \ldots, \sigma_{S_s}(t_2), \ldots, \sigma_{S_n}(t_n)),$$

where t_i is a "projection" of the query t on the language L_i.

---

1) The paper has been presented on the seminar led by prof. H. J. Schneider in Technische Universität Berlin

That is to mean that in order to answer a global query t we replace the query t by some queries $t_1, t_2, \ldots, t_n$, referring to corresponding local systems, and afterwords we form the global answer from the local answers obtained in this way. For example, let us assume that we have information systems owned by insurance company, medical care, service bank office, police etc. In each such a system we may answer specific queries related to the needs of the owner of the system, like "list all persons with the saving account greater than 10 000 $" or " list all persons which had a car accident in 1977" etc. Each such a query is related to specific information system, in which the relevant informations are stored. However, it may happen that some system user may be also interested in obtaining informations from several various systems, for example he may ask"whether there are persons who have caused a car accident while being treated with some drugs". This kind of queries cannot be answered by searching files in only one information system. This information is distributed at least in two systems: medical care system and insurance company system (or police system). Thus in order to get an answer to such a query we have to search for some information in more than one system. Of course the query language of each local system does addmit such general queries as given in the recent example. So we must also have an appropriate query language tailored properly to admit general queries.

The problem stated above is connected with another one: whether every local user is allowed to ask general queries or, in other words, whether any local user has access to

informations stored in another system. If not and this is widely used practice the question is how restrict access to protected informations against an unauthorised user. This problem will not be considered here, however we shall make some remarks concerning this subject.

Similar problem was also investigated by Marek [1], however there are some essential differences in our approach. We are mainly aiming at a logical analysis of distributed systems and their languages, whereas Marek's paper was devoted mainly to the problem of organizing "the data base machine".

We shall use standard mathematical notation throughout the paper.

We shall use the basic notions concerning information system after [2].

1. Information system

In this paper we shall use the notion of an information system introduced in [2]. Although the relevant definition is rather simple and does not cover some interesting cases of information system, however it expresses the most fundamental features of information systems and allows to investigate many basic properties of such systems.

By an information system we shall mean a 4-tuple

$$S = \langle X, A, V, \varrho \rangle ,$$

where

X - is a finite set of objects,

A - is a finite set of attributes,

$V = \bigcup_{a \in A} V_a$ is the domain of attribute a (or the set of

ribute a).

We assume that each attribute has at least two values.

ϱ - is an information function from $X \times A$ into V, such that $\varrho(x,a) \in V_a$ for every $x \in X$ and every $a \in A$;

$$V_a = \left\{ v \in V : \bigvee_x \varrho(x,a) = v \right\}.$$

Any pair (a,v), $a \in A$, $v \in V_a$ will be called a descriptor of attribute a.

For any $x \in X$, by ϱ_x we denote a function from A into V such that $\varrho_x(a) = \varrho(x,a)$. This function will be called the information about x in S. (Information about x in S may be also called the document about x in S or data about x in S).

For a given information system we introduce two binary relations $\widetilde{a}$, $\widetilde{S}$ defined as follows:

$$x \widetilde{a} y \text{ iff } \varrho(x,a) = \varrho(y,a)$$

$$x \widetilde{S} y \text{ iff } \varrho_x = \varrho_y.$$

One can simply check that both $\widetilde{a}$ and $\widetilde{S}$ are equivalence relations on X and

$$\widetilde{S} = \bigcap_{a \in A} \widetilde{a},$$

where $\bigcap$ is a product of partitions (equivalence relations) defined in the usual way.

The relation $\widetilde{S}$ will be also denoted by $\widetilde{A}$. Similarly, for any $A' \subset A$, $\widetilde{A'}$ is to mean $\bigcap_{a_i \in A'} \widetilde{a_i}$.

The family of equivalence classes $X/\widetilde{S}$ will be denoted by E_S (when X is fixed) and the equivalence classes will

be called elementary sets in S (or, when S is fixed, - elementary sets).

Similarly, for any $a \in A$, E_A is to mean the family of equivalence classes of the relation $\widetilde{a}$.

Empty set ϕ and every set being an union of some elementary sets in S will be called describable set in S. The class of an describable sets in S will be denoted by D_S.

2. Attributes

Let $a,b \in A$ be two attributes in an information system $S = \langle X, A, V, \varrho \rangle$.

Attribute b is said to be dependent on a ($a \Rightarrow b$) iff $\widetilde{a} \subset \widetilde{b}$.

Attributes a,b are said to be equivalent ($a \approx b$) iff $\widetilde{a} = \widetilde{b}$.

Attributes a,b are called independent iff neither $\widetilde{a} \subset \widetilde{b}$ nor $\widetilde{a} \supset \widetilde{b}$ holds.

Similarly we introduce the relations $B \Rightarrow a$, $a \Rightarrow B$, $B \Rightarrow C$, where B, C are subsets of A.

Attribute a is said to be dependent on the set of attributes $B \subset A$, $B = \{b_1, b_2, \ldots, b_n\}$ iff $\widetilde{B} \subset \widetilde{a}$, and $a \Rightarrow B$ iff $\widetilde{a} \subset \widetilde{B}$. In general we may write $B \Rightarrow C$ iff $\widetilde{B} \subset \widetilde{C}$.

The meaning of the "depending" relation $B \Rightarrow C$ is obvious. It simply means that values of the left hand side attributes determine values of the right hand side attributes.

That is to say, if $B \Rightarrow C$, then there exists exactly one function

$$f : \prod V_b \to \prod_{c \in C} V_c$$

such that

$$(\varrho(x,c))_{c\in C} = \mathbf{f}\,((\varrho(x,b))_{b\in B}).$$

In other words there exists exactly one set of functions $(f_c)_{c\in C}$, such that

$$(\bigwedge_{c\in C})\ (\varrho(x,c) = f_c((\varrho(x,b))_{b\in B})),$$

and

$$\varrho(x,c) = f_c(\varrho(x,b))_{b\in B})$$

iff

$$X_c, \varrho(x;c) \quad \supseteq \quad X_b, \varrho(x,b_1) \cap X_{b_2}, \varrho(x,b_2) \cap \ \ldots$$

$$\ldots \cap X_{b_r}, \varrho(x,b_a), \qquad \text{for all} \quad x\in X,$$

where

$$X_{c,r} = \left\{x\in X : \varrho_x(c) = v\right\}.$$

Besides the functions f_c, $c\in C$ we shall also use functions f_c^*, with the following domains and codomains

$$f : \prod_{b\in B} E_b \longrightarrow E_c$$

defined as follows:

$$X_c, \varrho(x,c) = f_c^*((X_b, \varrho(x,b))_{b\in B})$$

iff

$$\varrho(x,c) = f_c(\varrho(x,b))_{b\in B}),$$

for all $x\in X$.

3. REDUCED SYSTEMS

As we have stated in the previous section, same attributes in the information system may be superfluous, in this sence that their values can be "derived" from the values of other attributes in the system. A natural question, arises whether we can remove all dependent attributes from the system and what properties has the system obtained in this way. Such a system will be called a reduced information system. In this paper we shall only give a definition of a reduced system; some basic properties of reduced systems will be published alsewhere.

Let $S = \langle X, A, V, \varrho\rangle$ be an information system.

A subset $B\subseteq A$ is called independent in S iff, for every $B'\subsetneq B$ $\widetilde{B} \neq \widetilde{B'}$.

If $B\subseteq A$ is independent in S, then for every $B'\subset B$ and every $a\in B - B'$ $B \not\Rightarrow a$.

A subset $B\subseteq A$ is said to be dependent in S iff there exists a subset $B'\subsetneq B$ such that $\widetilde{B'} = \widetilde{B}$.

If B is dependent in S then there exists $B'\subset B$ and $a\ B - B'$ such that $B \Rightarrow a$.

The set A is said to be derivable from A' iff $A'\subset A$ and $\widetilde{A} = \widetilde{A'}$.

If $A'\subset A$ and A is derivable from A' in S then $A' \Rightarrow A - A'$.

A set $A' \subset A$ will be called a reduct of A iff $\widetilde{A} = \widetilde{A'}$ and there does not exist a proper subset B of A' such that $\widetilde{B} = \widetilde{A'}$.

The corresponding system $S = \langle X, A', V, \varrho' \rangle$ will be called a reduced system (ϱ' is the restriction at the function S to the set $X \times A'$).

Naturally, a given system may have more than one reduct.

In the sequel we shall consider only reduced information system, unless stated to the contrary.

4. SUBSYSTEMS AND APPROXIMATIONS

Let $S = \langle X, A, V, \varrho \rangle$ and $S' = \langle X', A', V', \varrho' \rangle$ be two information systems.

We say that S' is a subsystem of S $(S' < S)$ if $X' \subset X$, $A' \subset A$, $V' \subset V$, and

$$\varrho' = \varrho \,/_{X' \times A'}.$$

If $S' < S$ and $X' = X$ we shall say that S' is an attribute restricted subsystem of S (in short a.r. system).

If $S' < S$ and $A' = A$, $V' = V$, and

$$\varrho' = \varrho \,/_{X' \times A'}$$

then S' will be refered to as an object restricted subsystem of (in short o.r. system).

If $S' < S$ and S' is a.r. system then $\widetilde{S'} \supset \widetilde{S}$.

If $S' < S$ and S' is o.r. system then

$$\widetilde{S'} = \widetilde{S} \cap (X')^2$$

or, in other words, for any $e' \in E_{S'}$, there exists $e \in E_S$ such that $e' = e \cap (X')^2$.

Let $S = \langle X, A, V, \varrho \rangle$ and $S' = \langle X, A', V', \varrho' \rangle$ be two information systems and let S' be a.r. subsystem of S. Let $Y \in D_S$ and $Z \in D_{S'}$. Thus Y and Z are sums of some elementary sets in S and S', respectively.

Z will be called an upper approximation of Y in S' if $Y \subset Z$.

If Z is an approximation of Y in S' and there does not exist a proper subset $Z' \subset Z$ such that $Y \subset Z'$, $Z' \in D_{S'}$, then Z will be called the best upper approximation (b.u.a.) of Y in S'.

If Z is the b.u.a. of Y in S' we shall write

$$Z = \left| Y \right|^{*}_{S'}.$$

It is easy to prove the following property:

Let $S = \langle X, A, V, \varrho \rangle$ and $S' = \langle X, A', V', \varrho' \rangle$ be two information systems, and let S' be an attribute restricted subsystem of S. Then for every $Y \in D_S$ there exists exactly one b.u.a. Z of Y in S', namely,

$$Z = \left| Y \right|^{*}_{S} = \bigcup_{i=1}^{k} f^{*}(Y_i),$$

where

$$Y = \bigcup_{i=1}^{k} Y_i, \qquad Y_i \in E_S,$$

and

$$f^{*} : E_S \longrightarrow E_{S'}.$$

is such that for any $e \in E_S$, $e' \in E_{S'}$ $f^*(e) = e'$, iff $e \subset e'$.

The proof is by induction on the number of elementary sets in Y.

We shall also need the notion of the best lower approximation (b.l.a.) of Y by Z in S'.

Let $S = \langle X, A, V, \varrho \rangle$ and $S' = \langle X', A, V, \varrho' \rangle$ be information systems, and let S' be object restricted subsystem of S. Let $Y \in D_S$ and $Z \in D_S$. Z will be called the best lower approximation of Y in S' iff $Y \supset Z$ and there does not exist $Z' \supset Z$ such that $Z' \in D_S$ and $Z' \subset Y$.

The b.l.a. Z of Y in S' will be denoted as

$$Z = |Y|_{*S'}.$$

The following property is valid:

Let $S = \langle X, A, V, \varrho \rangle$ and $S = \langle X', A, V, \varrho' \rangle$ be two information systems, and let S' be object restricted subsystem of S. Then to every $Y \in D_S$ there exist exactly one b.l.a. Z of Y in S', and

$$Z = |Y|_{*S'} = \bigcup g^*(Y_i),$$

where

$$Y = \bigcup_{i=1}^{h} Y_i, \qquad Y_i \in E_S$$

and

$$g^* : E_S \longrightarrow E_{S'} \cup \{\emptyset\}$$

is defined as follows:

$$g^*(e) = e \cap (X')^2$$

The proof is also by induction on the number of elementary sets in Y.

5. JOIN OF INFORMATION SYSTEMS

If $S_1, S_2, \ldots, S_k$ are information systems, we can define a "common" information system S such that $S_1, S_2, \ldots, S_n$ are subsystems of S. The system S will be called join of S_i and will be denoted as $S = \bigcup_{i=1}^{k} S_i$, $S_i = \langle X_i, A_i, V_i, \varrho_i \rangle$.

The join S of S_i is defined in the following way:

$$S = \langle X, A, V, \varrho \rangle,$$

where

$$X = \bigcup_{i=1}^{k} X_i$$

$$A = \bigcup_{i=1}^{k} A_i$$

$$V = \bigcup_{i=1}^{k} V_i$$

$$\varrho / X_i \times A_i = \varrho_i \qquad i = 1, \ldots, k$$

or $\varrho_x = \bigcup_{i=1}^{k} \varrho_{ix}$, for all $x \in X$.

The join S of S_i is well defined if the following two

1°. $(\bigwedge_{i,j})(X_i \cap X_j \neq \emptyset$ and $A_i \cap A_j \neq \emptyset \rightarrow$

$$\varrho_i/[(X_i \cap X_j) \times (A_i \cap A_j)] =$$

$$\varrho_j/[(X_i \cap X_j) \times (A_i \cap A_j)] ,$$

and

2°. $(\bigwedge_{x \in X})(\varrho_x = \bigcup_{i=1}^{k} \varrho_{ix}$ is defined for an' $a \in A$.

The first condition is obvious and the second need some explanations.

Let S_1 be a system with only one attribute, say colour and S_2 a system also with one attribute for example length and assume that $x_1 \cap x_2 = \emptyset$. The condition 2° says that we are not allowed to define join S of S_1 and S_2 because we do not have any information about lengths of objects in S_1 and about colours of objects in S_2. Thus we are unable to define for all x X the information ϱ_x about colour and length of x. In other words, we are not able to define the function ϱ_x for the join $S = S_1 \cup S_2$.

This seems to have natural justification in reallife systems. If we have two information systems, say first concerning insurance and the second medical care with different sets of population (for example one in London and the second in Warsaw), then combining these two systems into one joint system is justified only in the case when we have insurance data in medical system and conversely. Otherwise we are unable to define for all $x \in X$ the information ϱ_x about insurance and medical care and, consequently, according to our definition the join of these two systems in not an in

Evidently if $S = \bigcup_{i=1}^{k} S_i$, then

$$\widetilde{S} = \bigcap_{i=1}^{k} \widetilde{S}_i .$$

Moreover, every S_i is a subsystem of S.

We shall distinguish two kinds of joins S.

In the first case we assume that every subsystem S_i of the system S is attribute restricted and such a joint system will be called an <u>attribute joined system</u> (or an attribute join of $S_1,\ldots,S_n$).

The second case refers to the system S in which every S_i is object restricted subsystem of S. This kind of system we will refer to as an <u>object joined system</u> (or an object join of $S_1,\ldots,S_n$).

Thus in an attribute joined $S = \bigcup_{i=1}^{k} S_i$

$$S = \langle X, A, V, \varrho \rangle$$

we have the subsystems

$$S_i = \langle X, A_i, V_i, \varrho_i \rangle ,$$

and in the object joined system $S = \bigcup_{i=1}^{k} S_i$

$$S = \langle X, A, V, \varrho \rangle$$

as subsystems we have

$$S_i = \langle X_i, A, V, \varrho_i \rangle .$$

6. QUERY LANGUAGE OF INFORMATION SYSTEM

With every information system S we associate a query language L_S which will be used for asking queries about informations contained in the system S. For the sake of simplicity we consider a very simple language, which is apart of a query language defined in .

The query language considered in this paper will consist only of terms, which are logical combinations of descriptors. That is to say a term (a query) in the considered language is a descriptor, or any expression built up from descriptors by means of logical connectives "or", "and", "not".

So the alphabet of L_S consists of the following symbols:

1° 0,1 - constans

2° A,V - attributes and values of attributes

3° $\cup$, $\cap$, $\sim$ - symbols for logical connectives "or", "and", "not" respectively

4° (,) - parentheses.

Terms (queries) in the language L_S are defined in the following manner

1° 0,1 are terms

2° any pair (a,v), $a \in A$, $v \in$ a is a term

3° If t,t' are terms, so are $\sim(t)$, $(t \cup t')$, $(t \cap t')$.

The meaning of a term t (an answer to the query t) is a function

$$\delta_S : \tau_S \to \wp(X),$$

where:

τ_S - is the set of all terms in the language L_S.

The meaning is defined in the following way:

1° $\delta_S(0) = \emptyset$, $\delta_S(1) = X$,

2° $\delta_S(a,r) = \{x \in X : \varrho_x(a) = v\}$,

3° $\delta_S(\sim t) = X \setminus \delta_S(t)$

$\delta_S(t \cup t') = \delta_S(t) \cup \delta_S(t')$

$\delta_S(t \cap t') = \delta_S(t) \cap \delta_S(t')$.

A term t will be called an <u>elementary term</u> if it is of the form

$$t = (a_1, v_{i_1}) \cap (a_2, v_{i_2}) \cap \ldots \cap (a_n, v_{i_n}),$$

$$A = \{a_1, a_2, \ldots, a_n\}, \quad v_{i_j} \in V_{a_j}.$$

If τ_E is the set of all elementary terms in L_S, then

1° $(\bigwedge t,t' \in \tau_E)(\delta_S(t) \cap \delta_S(t') = \emptyset)$

2° $\bigcup_{t \in \tau_E} \delta_S(t) = X$.

Thus the set of all elementary terms generates a partition of the set X (i.e. defines an equivalence relation on X).

It is easy to see that this equivalence relation coincides with relation $\tilde{}_S$.

Namely, if $\delta_S(t)$ is not empty, then $\delta_S(t)$ is just an equivalence class of the relation $\underset{S}{\sim}$. Thus elementary terms may be considered as unique names with nonempty values (in the language L_S) of elementary sets, (i.e. if e,e' are two different elementary sets in S then the corresponding elementary terms are different two).

A term t is in <u>normal form</u> if it is a sum of some elementary terms, i.e.

$$t = t_1 \cup t_2 \cup \dots \cup t_l,$$

where t_i - are elementary terms.

The following is true:

For every term t in L_S there exists $t' \in L_S$ such that t' is in normal form and

$$\delta_S(t) = \delta_S(t').$$

The subset $Y \subset X$ is called <u>describable</u> in L_S iff there exists a term t in L_S such that

$$\delta_S(t) = Y$$

From the normal form property we can deduce that all nonempty describable sets in L_S are unions of some elementary sets in S. That means the set of all describable sets in L_S is identical with the set $D_S(X)$ introduced in the first paragraph of this paper.

If $S' \prec S$, then we shall say that the language $L_{S'}$ is a <u>sublanguage</u> of L_S.

If S' is attribute restricted subsystem of S, then $L_{S'}$ will be called <u>syntactically restricted sublanguage</u> (s.r.s.) of L_S.

If $L_{S'}$ is a s.r.s. of L_S, then for every term t in L_S, there exists a term t' in $L_{S'}$ such that

$$\delta_S(t') = \left|\delta_S(t)\right|^*_{S'}.$$

If S' is object restricted subsystem of S, then both languages L_S and $L_{S'}$ are identical, but the meanings of terms t in these languages are different. (Because both systems' have different sets of objects).

Namely, for any term t in L_S (or in $L_{S'}$)

$$\delta_{S'}(t) = \left|\delta_S(t)\right|_{*S'}.$$

If S is an attribute join of systems S_i, then each sublanguage L_{S_i} is naturally a syntactical restriction of the language L_S.

The language L_S corresponding to the join S of systems S_i will be called the <u>join of languages</u> L_{S_i} and denoted by

$$L_S = \bigcup_{i=1}^{k} L_{S_i}$$

The join of languages L_{S_i} is obtained simply by admitting in the definition of the languages L_S an attributes and values from all sublanguages L_{S_i}.

In an object joined system S all sublanguages L_{S_i} are the same and the language $L_S = L_{S_i}$.

In this case the syntax of each language L_{S_i} is exactly the same, but the meaning of two identical terms belonging to different languages is different.

7. DISTRIBUTED INFORMATION SYSTEMS

Now we shall return to our original problem stated at the beginning of this paper.

Assume we are given some information systems $S_1, S_2, \ldots, S_k$, together with the languages $L_{S_1}, L_{S_2}, \ldots, L_{S_n}$ associated with them.

We may think of this systems as separated units, not connected together, and used independently.

We can assume however that that there is a kind of link between those systems so that they form one distributed information system, which may be viewed as a join of individual systems S_i. The language L_S of the join system S may be also considered as a join of all sublanguages L_{S_i}.

So we may come back now to the original question:

How to find an answer to the query in the language L_S by retrieving informations in subsystems S_i.

We shall consider in this paper two basic types of information systems.

S will be called a <u>language distributed system</u> if

$$S = \langle X, A, V, \varrho \rangle ,$$

and

$$S = \bigcup_{i=1}^{k} S_i ,$$

where

$$S_i = \langle X, A_i, V_i, \varrho_i \rangle$$

$$\varrho_i = \varrho / X \times A_i$$

and

$$A_i \cap A_j \neq \phi \rightarrow \varrho_i / X \times (A_i \cap A_j) = \varrho_j / X \times (A_i \cap X_j)$$

S will be called <u>data distributed system</u> if

$$S = \langle X, A, V, \varrho \rangle$$

and

$$S = \bigcup_{i=1}^{k} S_i$$

$$S_i = \langle X_i, A, V, \varrho_i \rangle ,$$

$$\varrho_i = \varrho / X_i \times A$$

and

$$X_i \cap X_j \neq \phi \rightarrow \varrho_i / (X_i \cap X_j) \times A = \varrho_i / (X_i \cap X_j) \times A .$$

8. LANGUAGE OF DISTRIBUTED SYSTEM

Let $S = \langle X, A, V, \varrho \rangle$ be a language distributed system, i.e.

$$S = \bigcup_{i=1}^{k} S_i$$

and

$$S_i = \langle X, A_i, V_i, \varrho_i \rangle$$

If $t \in E_S$ (i.e. t is an elementary term in L_S), then by t/A_i we shall mean the elementary term $t' \in L_S$ obtained from t after removing all descriptors which do not belong to L_{S_i} (if after this removal no descriptors remain, then we assume that $t/A_i = T$).

Then the value of any term t in L_S can be represent as follows

$$\delta_S(t) = \bigcup_{j=1}^{n} \bigcap_{i=1}^{k} \delta_{S_i}(t_j/A_i),$$

where

$t_1 \cup t_2 \cup \ldots \cup t_n$ is a normal form of t,

Obviously we have $\delta_{S_i}(t_j/A_i) = \left|\delta_S(t_j)\right|^*_{S_i}$

This property means simply that in order to find an answer to the query t in the language L_S, we must first translate it into a normal form $t_1 \cup t_2 \cup \ldots \cup t_n$. Then for any $1 \leq i \leq k$ and any $1 \leq j \leq n$, we remove from each elementary term all descriptors which do not belong to the language L_{S_i}, i.e. for each $1 \leq i \leq n$ we replace all elementary terms by terms of the form $t_j/A_i = t_j'$ belonging to the languages L_{S_i}. Afterwards we have to find the best upper approximation

$$\left| t_j \right|^*_{S_i} = \delta_{S_i}(t_j'),$$

for each term t_j in every subsystem S_i. Then the intersection of all approximations corresponding to a fixed elementary term t_j constitutes the proper answer to this term.

In order to obtain the whole answer to the query t we have to "add" the answers to all elementary terms occuring in the normal form of t.

The situation may be depicted as shown in figure 1.

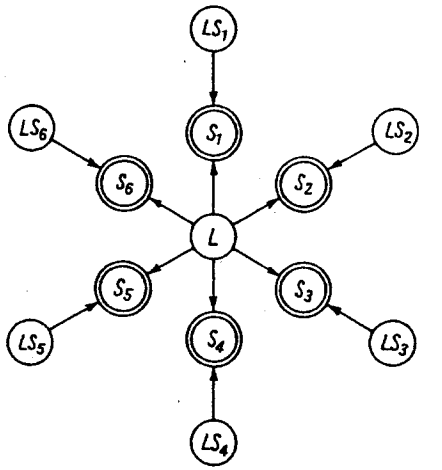

Fig. 1.

So we have in this case two kinds of users: local and global (central) ones. Local users are attached to local systems use local languages L_i, and have access only to informations in local systems. The global user can ask queries in the global language L and he has access to the informations in all local systems.

This kind of organization of distributed system has one serious disadvantage. In order to find the answer to a global

r the best upper approximations of

ystem and take the intersection of

all approximations. The intersection operation is very uneffi-cient, because it requires access to many disc memories in order to retrieve the best upper approximations of elementary terms, but only small part of thus obtained data will occur in the intersection of all approximations.

Thus another solution of this kind of systems seems to be more efficient. This solution is depicted in fig. 2.

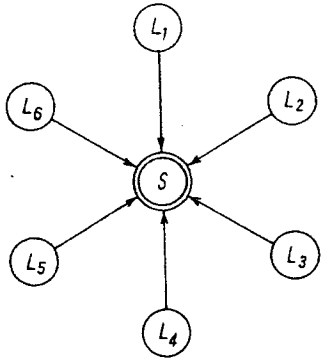

Fig. 2.

There is only one central systems and each user uses its own language L_i which can be any sublanguage of L_S. Because, in this case there is only one partition $\widetilde{S}$ generated by the global system S the answer to any query in global language or any sublanguage of the global language may be obtained directly from the central memory as a sum of some elementary sets in S. Thus in this case there is no intersection operation, which considerably slows down the retrieval

We shall call the first type of language distributed systems - local memory systems. The second kind of systems considered in this paragraph will be referred to as central memory systems.

9. DATA DISTRIBUTED SYSTEMS

In this paragraph we shall deal with the case when the subsystems S_i of the system S have the same languages L_S, but different sets of objects.

Let S be a data distributed system i.e. let $S = \langle X, A, V, \varrho \rangle$ be a system

$$S = \bigcup_{i=1} S_i ,$$

where

$$S_i = \langle X_i, A, V, \varrho_i \rangle ,$$

$$\varrho_i = \varrho \,/ X_i \times A,$$

and

$$X_i \cap X_j \neq \phi \rightarrow \varrho_i/(X_i \cap X_j) \times A = \varrho_j/(X_i \cap X_j) \times A.$$

If S is a data distributed system then for every term t in L_S

$$\delta_S(t) = \bigcup_{i=1}^{k} \delta_i(t).$$

Obviously, we have

$$\delta_{S_i}(t) = \left|\delta_S(t)\right| * S_i$$

where the normal form of t is

$$t = t_1 \cup t_2 \cup, \ldots, \cup t_n .$$

From this property it follows that in order to find the answer to the querry t in L_S we have to find partial answers $\delta_i(t)$ in all subsystems of S. This partial answers are sums the best lower approximations for t in all subsystems of the system S. The answer to the querry t is the sum of all partial answers. The operation is very simple to implement and is much faster then computing the intersection of upper approximations in the previous case.

We can also distinguish two kind of memory organizations: local memory systems and central memory systems.

The implementation in all cases is similar and it is rather simple.

Acknowledgment

Thanks are due to dr B. Konikowska for reading the manuscript and essential improvement of the paper.

I would like also to express my thanks to prof. H. J. Schneider, Dr. P. Bollmann and Dr. E. Konrad for stimulating discussion and valuable remarks and comments.

Received May 24, 1979