Journal of Computing and Information Technology - CIT 2, 1994, 3, 167–179 

# HETHI – Heterogeneous Hierarchical Knowledge Base

Slobodan Ribarić and Berislav Lastrić

Faculty of Electrical Engineering, University of Zagreb, Croatia

The structure of a knowledge base designed by using an extension of the knowledge representation scheme KRP originally developed for the knowledge representation in computer vision systems, is presented in this paper. The HETHI — heterogeneous hierarchical knowledge base consists of one level of the Kanerva-like sparse distributed memory (SDM) and knowledge base designed by the KRP knowledge representation scheme based on the Petri Net theory. The lowest level of the HETHI implemented by Kanerva-like SDM performs associative retrieval information process and supports initialization of the inheritance process at the higher levels of the HETHI. Higher levels of the HETHI are semantic and rule-based levels. In this paper we discuss different reasoning procedures which are supported by the HETHI: "pure" associative inference or the so- called recall procedure which is performed only by means of the first level of the HETHI; inheritance procedure defined at semantic level, and finally, inheritance procedure based on the cooperation of the associative recall and inheritance procedure defined at semantic level.

## 1. Introduction

Intelligent systems require a large amount of knowledge as well as some mechanisms for manipulating this knowledge in order to perceive, organize and summarize observations, stimulus and information from the problem domain, and support a useful degree of problem-solving ability [1]. The knowledge is stored in the knowledge base that can be defined as an abstract representation of the "world" in which the system has to solve tasks. The representation and reasoning facilities in intelligent systems must be able to integrate different kinds of knowledge [2]: objects and relationships, collection of facts and uncertain facts, constrains, rules of the "world" and decision rules, problem solving methods, procedures and heuristics, behaviour descriptions and typical situations, descriptions of the motivations and goals of the system, collection of the states of the world and states of an agent, process descriptions and knowledge about knowledge or meta- knowledge.

In many cases the knowledge in the knowledge bases of intelligent systems can be considered as naturally structured systems depicted as a succession of levels of representation. For example, a knowledge base of a computer vision system can be represented as a multilevel hierarchical system. At lower levels (image oriented or iconic), which are domain-independent, the knowledge of the physics of the imaging process and the illumination is required. At lower levels the knowledge supports resolving process of the ambiguity associated with transformation from a three- dimensional world to a two-dimensional image. At higher levels, knowledge from models is used for grouping intrinsic characteristics into surfaces and volumes, recognition objects and bodies as well as for iconic to symbolic transformation. According to Barrow and Tenenbaum [3] the highest level of a general-purpose vision system can, by using knowledge about intentions and causalities, recognize events and generate events descriptions.

The other examples of the multi-level knowledge-based robot or computer vision systems are given in [4], [5], [6].

In computer vision systems the hierarchical knowledge representation is used. The existence of conceptual gap between lower levels (sensor, image formation, segmentation) and higher levels (scene interpretation) of the model, influences the organization and form of knowledge representations. Computer vision systems use different knowledge representation schemes for different model levels. For example, system VISION [4] uses a hierarchical modular approach to represent knowledge and control: relaxation approach to organize edges into boundaries and pixel clusters into regions, rule-based object hypothesis for initial iconic to symbolic transformation, and (at the highest level of representation) scene schemes. For scene independent knowledge, VISION uses semantic networks, where nodes represent primitive entities (objects, concepts, situations, etc.) and labelled arcs represent relationships between them.

The paper is organized as follows. The components of the HETHI are described in Section 2. Inheritance procedures based on co-operation of the first level of knowledge base which is implemented at associative level and higher levels implemented by the KRP scheme are represented in Section 3. In Section 4, an example of inheritance procedure is given.

## 2. Components of HETHI

Components of the proposed heterogeneous hierarchical knowledge base are : hierarchical knowledge base designed by KRP knowledge representation scheme and one level Kanerva-like sparse distributed memory model [7]. The main reason for such heterogeneous hierarchical structure of knowledge organization lies in some limitations and drawbacks of the inheritance process of the "pure" KRP scheme (only one source of activity in the inheritance process, limitation of the storage capacity of hardware designed part of the knowledge base). These limitations and drawbacks are not only features of the above mentioned scheme, but also many other knowledge representation schemes (e.g. semantic networks, frame-based schemes, production systems) have similar characteristics.

In this section we give only a brief overview of the two main organizational components of the HETHI, Kanerva-like sparse memory model and the KRP scheme [8], [9], necessary for understanding the structure of the HETHI.

The lowest level of the HETHI (Figure 1) is an *associative level* that is implemented by means of Kanerva-like sparse distributed memory model [5].

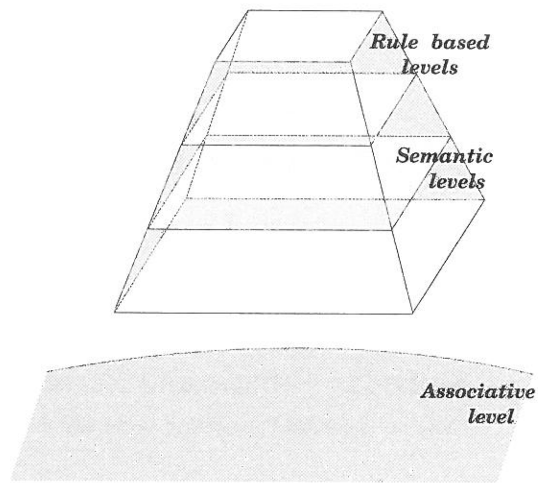



*Fig. 1.* Two main organizational components of the HETHI.

The associative level is based, as well as the Kanerva's sparse distributed memory, on the space $\{0,1\}^n$. Elements of the space are *n-dimensional vectors* with binary components. Total number of elements is $2^n$, that number is denoted by N. N stands also for the space itself.

Primary property of the space N is its distribution, based on distances between vectors of N. Distance among vectors $x$ and $y$ is defined as a number of dimensions or components at which $x$ and $y$ differ:

$$d(x,y) = |x - y| \tag{1}$$

where $x - y$ stand for bitwise 'exclusive or' of $x$ and $y$. The distance is called the Hamming's distance. Two vectors are similar if they are relatively close.

The number of points that are exactly $d$ bits from an arbitrary point $x$ is equal to binomial coefficient $\binom{n}{d}$. That means that distribution of the space $N$ is binomial distribution with parameters $n$ and $p = 0.5$, with the mean equals $\frac{n}{2}$ and variance equals $\sqrt{\frac{n}{4}}$. The distribution function is denoted as $N(d)$. The normal distribution $F$ with mean $\frac{n}{2}$ and standard deviation $\sqrt{\frac{n}{4}}$ is a good approximation to it:

$$N(d) = \Pr\{d(x,y) \le d\} \cong F\left\{\frac{d - \frac{n}{2}}{\sqrt{\frac{n}{4}}}\right\}. \tag{2}$$

If we divide the mean distance $\frac{n}{2}$ by the standard deviation of distance $\sqrt{\frac{n}{4}}$, we get that the distance from a point to the bulk of the space is $\sqrt{n}$ standard deviations. Property of normal distribution function is that less then $10^{-6}$ of the space lies outside of 5 standard deviations of the mean.

The concept of the *address region* is defined on a basis of the circle O and its distribution. The circle O(r, x) is defined as a set of points:

$$O(r,x) = \{y \mid d(x,y) \leq r\}, \qquad (3)$$

where point $x$ represents the address of the address region i.e. the center of the circle $O$, and $r$ is radius of the circle $O$. The circle $O(r,x)$ defined in the space $N$ has not a feature of convexity as the space itself has not.

Distribution of the circle depends on number of vectors in the circle (i.e. circle area) and their distribution inside the circle. The area of the circle $O(r,x)$ is defined as:

$$|O(r,x)| = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{r} \qquad (4)$$

or $N * N(r)$ according to a definition, and its normal distribution approximation

$$N * F\left\{\frac{r - \frac{n}{2}}{\sqrt{\frac{n}{4}}}\right\}.$$

With $r_p$ is denoted the radius that encloses portion $p$ of the space $N$ and so $N(r_p) = p$ and $|O(r_p,x)| = p * N$. Portion $p$ is function of radius $r$, while the ratio of two consecutive binomial coefficients $(n-r)/(r+1)$ is amount of area enlargement of the circle $O$ if we increase radius $r$ for one bit. So that when $r < \frac{n}{3}$, a one-bit increase in $r$ at least doubles the size of the circle, and over the half of the circle elements are maximally distanced from the center.

Concept of addresses overlapping is derived from an intersection of two circles with equal radii and well-known distance between their centres, [7]

$$I(d) = |O(r_p,x) \cap O(r_p,y)|. \qquad (5)$$

On the structure of the space $N$, model of sparse distributed memory and memory model at our associative level are built up. These models are characterized with ideas of:

**Similarity**, which is based on computing Hamming's distance between vectors. The vectors remoted $n/4$ bits apart are similar in the sense that a small portion of the space lies within $n/4$ bits [7]. Also, for any vector, we can find that almost all the space is indifferent to, i.e. about $n/2$ bits away from, given vector. This is intuitively connected with the idea that for any two unrelated concepts we can always find an intermediate concept that is related to each of the first two.

**Sparseness of memory** is defined by a concept of physical location, $x'$.

The following properties characterize sparseness of memory :

i) The storage locations are very few in comparison with $2^n$;

ii) An unique address is assigned to the storage location;

iii) The storage locations, i.e. their addresses are distributed randomly in the space $N$.

According to the above statements, and due to some limitations of the transforming concepts from the real world into their binary representation and manipulating with them, the possibility of addressing over the space $N$ is reduced to addressing over address locations of the subspace $N'$, $N' \subset N$. A location in the subspace $N'$ is represented by address vector that is $n$-binary vector, element of $N'$ and $N$.

Distance from a point $x$ to a location $y'$, $d(x,y')$, or from a location $x'$ to a location $y'$, $d(x',y')$, is defined by analogy as distance between two points. Also a distribution of distance $d(x,y')$ and a distribution of distance $d(x',y')$ which are the very same (because y' is also element of $N$) are defined accordingly as distribution of distance in the space $N$

$$N'(d) = \Pr\{d(x',y') \leq d\} = 1 - [1 - N(d)]^{(N'-1)}, \qquad (6)$$

where expression in the square brackets is probability that none of the rest $N' - 1$ independent random locations, as elements of $N$, is within $d$ bits of $x'$. $N'(d)$ is probability that at least one is within $d$ bits. $N'(d)$ can be rewritten as:

$$N'(d) \cong 1 - \left[1 - \frac{N' * N(d)}{N'}\right]^{N'} \cong 1 - \exp\{-N' * N(d)\}. \qquad (7)$$

On the basis of *circle* $O'(r,x')$ [5] and the bijective function .

$$\omega : C \rightarrow N', \tag{8}$$

which maps set of concepts into set of storage locations we have got powerful tool for defining and testing similarities and connections among concepts. The $c_k \in C$ represents an object from the problem domain. For $c_k$ and by means of function $\omega$ applied on $c_k$ i.e.

$$\omega : c_k \mapsto x', \tag{9}$$

the center of circle x' is obtained.

A circle O' with radius $r$ and center $x'$ is defined as a set of points $y'_j$, $j = 1, 2, \ldots$, from space $N'$ with the following properties:

$$d(x', y'_j) \leq r, \quad j = 1, 2 \ldots \tag{10}$$

and at the same time because $x'$ is also element of $N$:

$$O'(r,x') = N' \cap O(r,x') \tag{11}$$

and area of the circle $O'$ is:

$$|O'(r,x')| = \frac{N'-1}{2^n} * |O(r,x')|. \tag{12}$$

Thus the portion $p$ that the circle $O'$ shall occupy in $N'$ (or number of locations $y'_i$) is the function of the following arguments: $N'$, $n$ number of dimensions of the space $N$, and radius $r$. We can vary radius in such a way that a circle contains one, some or all of the locations of $N'$.

A concept $c_j = \omega^{-1}(y'_j)$, for $j = 1, 2, \ldots$, is similar to the concept $c_k$ according to some measure expressed by value $r$.

Concepts are connected and form a *group of concepts* during learning process, but qualitative criteria depend on the user's knowledge and intuition.

According to the fact that concepts are clustered into a group of concepts, the space $N'$ is divided into subsets of locations which form *location groups* and each location has a *group indicator*. A location group $G_i$, where $i = 1, 2, \ldots, g$ is a uniform random sample of $N'$ where

$$\bigcup_{i=1}^{g} G_i = N' \tag{13}$$

and

$$G_i \cap G_j = \emptyset \text{ for } i \neq j \tag{14}$$

Each location group has a number of locations approximately equal to the average number of storage locations $N'/g$, that are randomly distributed over the $N'$, where $g$ is a total number of groups.

The user has to specify, for each concept $c_k \in C$, the *name of the group*, possible similarity to another concept $c_j$, name of the group to which $c_j$ belongs, and *measure of similarity* is expressed by linguistic variable. Each of linguistic variables, from the set of linguistic variables $L$ is mapped, by means of bijective function $h$, in corresponding value of measure $m$ from the set of measures, $M = \{0, 1, \ldots, z-1\}$, where $z$ is the number of linguistic variables (Table1).

| $h : L \rightarrow M, z = 7$ | (15) |
|---|---|
| linguistic variable | measure of similarity, $m$ |
| extremely similar | 0 |
| very similar | 1 |
| considerably similar | 2 |
| moderately similar | 3 |
| more-or-less similar | 4 |
| minorly similar | 5 |
| minimally similar | 6 |

*Table 1*: Mapping of linguistic variable to measure of similarity.

Concepts are stored in locations of location groups on the basis of the following *concept storing algorithm*:

**Input:** Concept $c_k \in C$, *name of the group* $NGC_k$, (concept $c_j$, name of the group $NGC_j$, *measure of similarity* — optional).

**Step 1:** **IF** $NGC_k$ is the new name **THEN** assign $NGC_k$ to unused location group $G_k$.

**Step 2:** **IF** there is possible similarity $c_k$ of to $c_j$ **THEN** specify $NGC_j$ and measure of similarity.

**Step 3:** Determine a free location for storing the concept as follows:

**IF** Step 2 was performed **THEN** calculate

$$r' = f(z, m, r).$$

For example:

$$r' = \lceil k * r \rceil, \quad k = \frac{m}{z} + \frac{1}{z+1} \tag{16}$$

where $\lceil\,\rceil$ denotes the ceiling of expression.

Store the concept $c_k$ to the free location $x'$ that satisfies the condition:

$$\mathrm{Min}(\ \mathrm{ABS}(d(x', y'_j) - r')). \tag{17}$$

**ELSE**

Store the concept $c_k$ to some free location $x' \in NGC_k$ randomly chosen from $G_k$.

**Step 4:** Decrement number of free locations in the group $G_k$.

For example we want to store the concept "*sparrow*" that is somehow similar to the concept "*bread*". That means that input is:

$$c_k = \text{"sparrow"}, \quad NGC_k = \text{"BIRD"}$$
$$c_j = \text{"bread"}, \quad NGC_j = \text{"FOOD"}$$

and similarity between concepts is defined as *moderately similar*.

**Step 1.** is skipped if we suppose that any of the members of the group of concepts "BIRD" is already stored.

**Step 2.** According to an input information $c_j$ = "bread" – the name of the group is $NGC_j$ = "FOOD"
By means of the function $\omega$, the location $y'_j$ from $NGC_j$ is uniquely determined, $y'_j = \omega(c_j)$. Similarity between concepts is expressed as *moderately similar*.

**Step 3.** Perform a calculation: $r' = f(z, m, r)$
According to the Table 1, $z = 7$, and for the linguistic variable *moderately similar*, the measure of similarity is $m = 3$. Radius $r$ is function of $n$, $N'$ and $p$. For proposed values $N' = 100{,}000$, $n = 100$ and $p = 0.001$ (or circle $O'$ will contain approximately 100 locations) radius $r$ is: $r = 31$. For the given informations the coefficient is:

$$k = \frac{m}{z} + \frac{1}{z+1} = \frac{3}{7} + \frac{1}{7+1} = 0.553$$

and radius $r'$ is:

$$r' = \lceil k * r \rceil = \lceil 0.553 * 31 \rceil = 18.$$

For every location from the circle $O'(r, y'_j)$ calculate Hamming's distance from $y'_j$ and "remember" location $x'$ which Hamming's distance is the most equal to $r'$, i.e. location $x'$ that $y'_j$ and $x'$ are 18 bits distanced away. By means of function $\omega$ store the concept $c_k$ = "sparrow" in the location, $x' = \omega(c_k)$.

**Step 4.** Decrement number of free locations in group $G_k$.

*KRP* scheme is defined as 7-tuple: $KRP = (P, T, I, O, \mu, \alpha, \beta)$, where $P, T, I, O$ and $\mu$ are the components of a marked Petri net [10], [11]. General $P, T, I, O$ and $\mu$ are defined as follows:

- $P = \{p_1, p_2, \ldots, p_k\}$ is a finite set of places,
- $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions,
- $P \cap T = \emptyset$. (18)
- $I : T \rightarrow P^\infty$ is an input function, a mapping transitions to the bags of places, (19)
- $O : T \rightarrow P^\infty$ is an output function, a mapping transitions to the bags of places, (20)
- $\mu : P \rightarrow N$ is a marking, a mapping from places to non-negative integers $N$. (21)

The marking $\mu$ in the *KRP* scheme corresponds to the initial conditions, intermediate and final states of the knowledge base in procedures of inference (inheritance, recognition, and activity intersection search [12], [13], [14]).

$\alpha : P \rightarrow D$ (22) is bijective function, which associates a concept from set $D$ is bijective function, which associates a concept from set $D$ to every place $p_i \in P$. $D$ is a set of concepts used for representing objects and facts from the real world which are defined at higher levels of the HETHI. Based on hierarchical properties of the scheme, the elements in $D$ are the

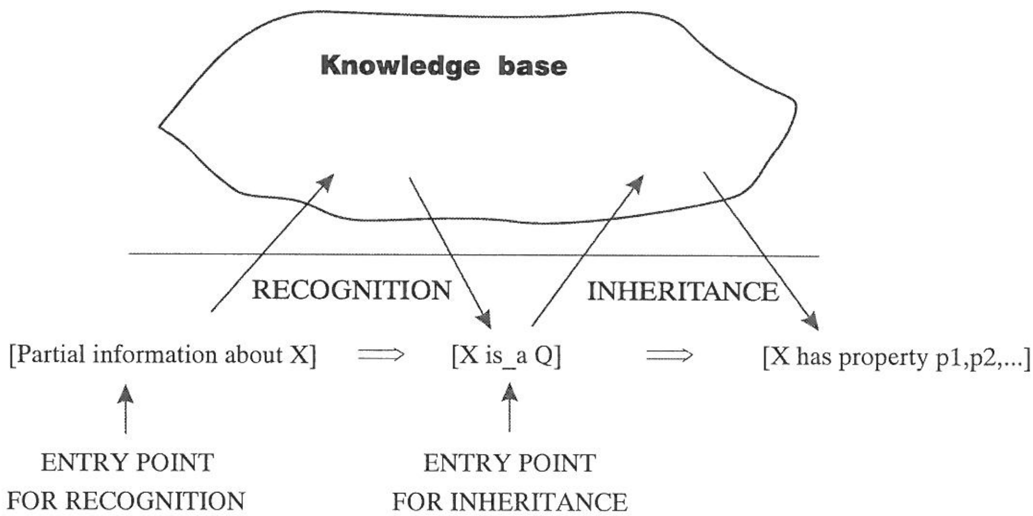



*Fig. 2.* Inheritance and recognition.

union $D = D_1 \cup D_2 \cup D_3$ (23) where: the subset $D_1$ corresponds to elements which represent characterics or individual exemplar or examples from the real world which is the subject of modelling; elements from subset $D_2$ denote classes or categories of exemplar and represent higher levels of abstraction, and, finally elements from subset $D_3$ represent intrinsic properties of concepts or values of these properties.

A surjective function $\beta : T \rightarrow \Sigma$ (24) associates elements from set $\Sigma$ to every transition $t_j \in T$. The elements from set $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$ (25) are also used for hierarchical structuring of knowledge: elements from $\Sigma_1$ correspond to relationship between concepts which are used for partial ordering of the set of concepts (e.g. "is_a", "is_part_of" etc.); elements from $\Sigma_2$ are used to specify types of properties to which values from $D_3$ are assigned, and, elements corresponding to relationship between concepts but not used for hierarchical structuring (subset $\Sigma_3$).

The set $U = D \cup C$ (26) represents a set of all concepts from the real world that are defined in the knowledge base. In general, the $D \cap C \neq \emptyset$, (27) where $C$ is a set of concepts from the associative level.

We can also define an inverse function $\alpha^{-1}$, and generalized inverse function $\beta^{-1}$ defined by mapping from every $\sigma \in \Sigma$ into a set $\tau \subseteq T$.

The *KRP* scheme is used for designing higher hierarchical levels which can be divided into two macro-levels: semantical level and rule-based level. For the rule-based level, the sets $D$ and $\Sigma$, and $\alpha$ and $\beta$ are appropriate modified: places correspond to conditions, and transitions correspond to events [8], [9], [11].

## 3. Inheritance in HETHI

Inheritance and recognition are two main inference processes in the network knowledge representation schemes [12]. Figure 2. shows entry points and outputs of both processes.

The entry point for an inheritance process is a known concept and the output are its properties. Inheritance can be defined as a form of reasoning which allows an agent to infer properties of a concept on the basis of the properties of its ancestors in a hierarchical structure. The properties are determined by looking up properties not only locally attached to the concept of interest, but also by looking up properties attached to all concepts that lie above in conceptual hierarchy [13],[14], [15].

The entry point for the recognition process (Figure 2.) is a partial information about an unknown concept $X$. By means of the recognition process, the output "$X$ is_a concept $Q$" is obtained, where $Q$ is the best matching among $X$ and concepts elements of set $U$. The recognition process can be considered as an inverse process according to inheritance.

Let us, for a moment, suppose that there is no associative level in the HETHI. Without going into detail description of the whole process of inheritance in the knowledge base designed by *KRP* (see [12], [13]) we can say that key points of the process are inverse function $\alpha^{-1}$ and inheritance tree. The inverse function $\alpha^{-1}$ maps concept of interest into corresponding place $p_i$ and determines initial marking (i.e. it puts token in the place $p_i$). The inheritance tree is similar to the reachability tree of Petri net [11] in which results of all sequences of firing of enabled transitions, starting with initial marking, are represented. In the *KRP* scheme the inheritance tree is obtained in such a way that after firing all enabled transitions which correspond to the elements in subsets $\Sigma_2$ and $\Sigma_3$, starting from the initial marking, corresponding tokens created in the output places are frozen [13] (Recall that only elements from subset $\Sigma_1$ are used for partial ordering of the set of concepts $D$).

If we are interested in properties of a concept $d_x$ and if the concept $d_x \notin D$, the process of inheritance will stop because the inverse function $\alpha^{-1}$ is not defined. The knowledge base, in this case, generates the answer : "The concept $d_x$ is unknown". The similar behaviour in the process of inheritance occurs for many other hierarchical knowledge representation schemes (e.g. NETL [16], semantic networks [17], FRL [15]). Even if $d_x \in D$, there is only one initial source of activity in the inheritance process.

In the HETHI knowledge base at the first level, i.e. associative level designed by Kanerva-like SDM, there are many concepts which are grouped according to some domain-dependent criteria and stored by means of concept storing algorithm. It is not necessary that every concept at associative level $c_i \in C$ is also an element of $D$ and vice versa. It is important to stress out that the storage capacity of the associative level is much larger than the capacity of *KRP* levels. The reasons for this are hierarchical concept-relationship structure of *KRP* and its hardware realization by means of p- and t- processors and interconnected switching network. By introducing an associative level in the knowledge base we have obtained more diverse and more efficient models of inheritance:

- a "pure" associative inference ("pure" API) or so-called recall procedure (Figure 3.) which is performed only by means of the first level of knowledge base, i.e. associative level. If the concept exists in the associative level of the HETHI, the concepts which lie in the circle $O''(r', x')$, defined by radius $r'$ (where $r' = \lceil k * r \rceil$ and $k$ is obtained by means of function $h$) with center corresponding to the concept, are recalled. In general $O'' \subset O'$. Also the names of corresponding groups are obtained. The above procedure is similar to human behaviour in the process of recalling the information. For example, the concept "sparrow" may associate somebody to "bird", "bread", "roof", "cat" and so on.

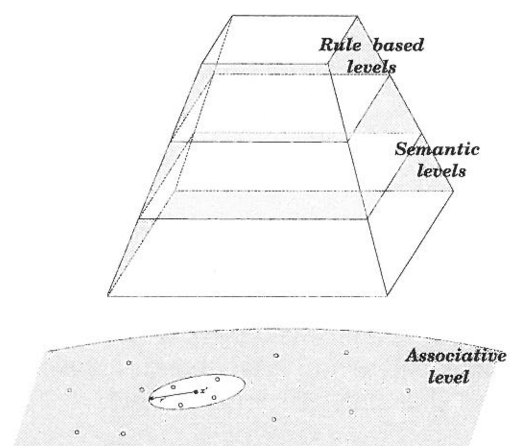



*Fig. 3.* A "pure" associative inference.

- associative recall based on transfer activity through the intersected circles (TAPI) (Figure 4.) The concept of interest $c_k \in C$ is mapped by means of function $\omega$, $\omega : c_k \mapsto x'$ into center of circle $O''$ with radius $r'$. The circle $O''$ is defined as a set of points $y_j'$, $j = 1, 2, \ldots$, from space $N'$.

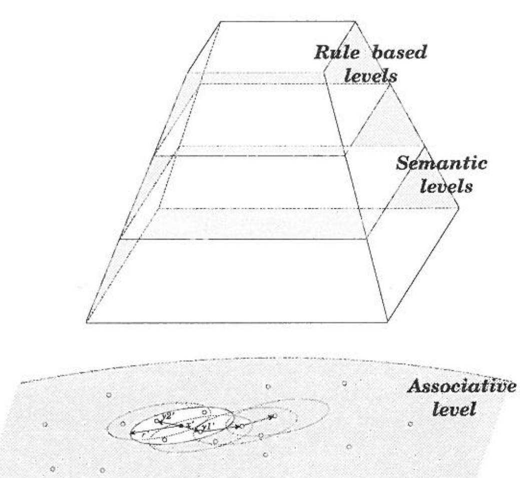



*Fig. 4.* Associative recall procedure based on transfer activity.

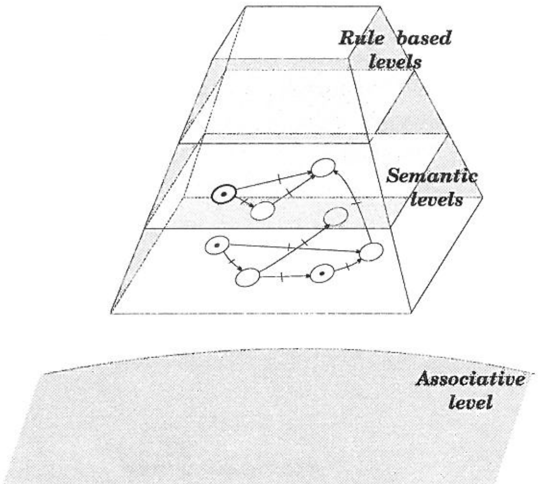



*Fig. 5.* A "pure" inheritance procedure.

The associated concepts are obtained by means of function $\omega^{-1}$ — every point $y_j'$ in this circle becomes new center of the circle with the same radius $r'$ and it recalls only new "non-retrieved" concepts and so on. The above process is performed also at the associative level.

• "pure" inheritance procedure ("pure" IP) defined at semantical level which is implemented by the *KRP* scheme (Figure 5.). If concept $d_x \in D$, where $D$ is a set of concepts defined in the *KRP* scheme, then by using inverse function $\alpha^{-1}$, the corresponding place in the knowledge base is found. The place $p_i$ is defined as a place with token and defines initial marking for a construction inheritance tree. The "pure" inheritance procedure is described in detail in [12]. If $d_x \notin D$, then inverse function is not defined and process of inheritance is stopped.

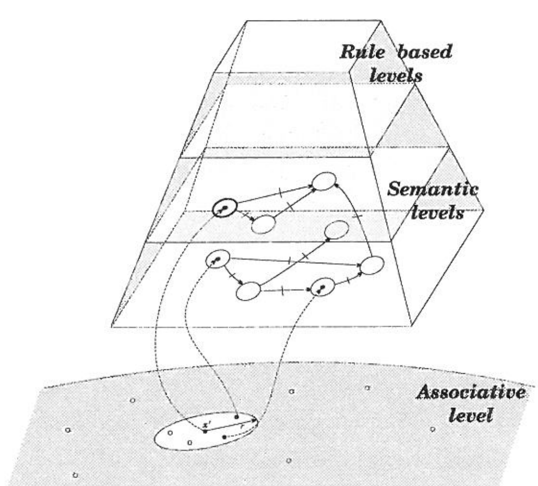



*Fig. 6.* Mixed inheritance procedure ($d_x \in D$).

• "mixed" inheritance procedure ("mixed" IP) which is based on the co-operation between the Kanerva-like SDM (associative level) and the *KRP* levels of the knowledge base. There are three main cases:

*i*) The concept of interest (Figure 6.). Using measure of similarity, expressed by $r'$, function $h$ and function $\omega$ the centre of circle $O''$ with radius $r'$ is defined. The inverse function $\alpha^{-1}$ can be defined for some concepts obtained by $\omega^{-1} : y_j' \mapsto c_j, j = 1, 2, \ldots$, that means $c_j$ is an element of $D$. All such concepts and the concept of interest define initial markings in the semantical level of the HETHI. These initial markings define many sources of activity which are used in inheritance procedure at the *KRP* level.

*ii*) The concept of interest $u_x \in U$, but $u_x \notin D$ and $u_x \in C$. In this case, the function $\alpha^{-1}$ is not defined for the concept of interest $u_x$. By using the function $\omega : u_x \mapsto x'$ the centre of the circle $O''(r', x')$ is defined at the associative level. The inverse function may be defined for some concepts which are obtained by applying the inverse function $\omega^{-1} : y_j' \mapsto c_j$ for some $j$-s. Also, the inverse function $\alpha^{-1}$ can be defined for some names of location groups (Figure 7).

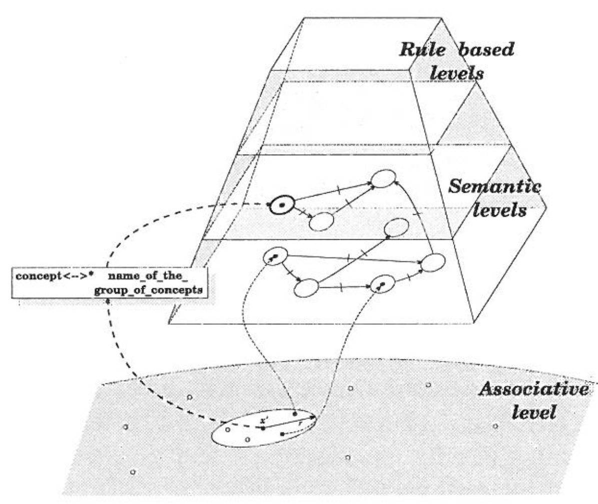



*Fig. 7.* Mixed inheritance procedure ($u_x \in U$, $u_x \notin D$).

*iii*) The combination of associative inference procedures "pure" API and TAPI is used to define initial sources of activity at semantical level of the HETHI (Figure 8) when inheritance procedure is activated.

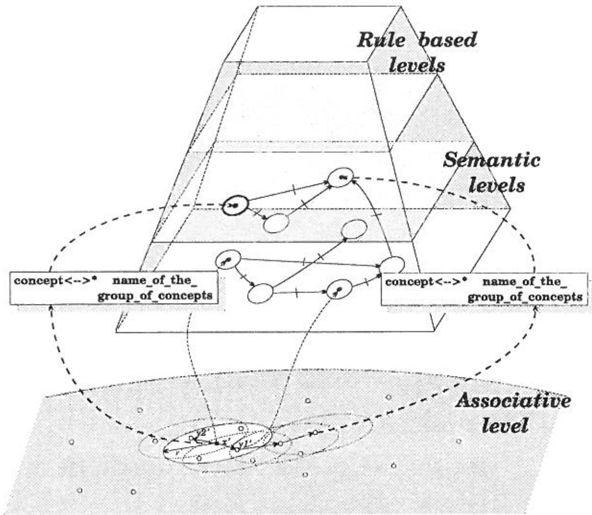



*Fig. 8.* The combination of associative inference procedures.

The following inheritance algorithm describes the inheritance procedure in detail:

*Inheritance algorithm*

**Input:** Concept of interest $d_x$, measure of similarity expressed by a linguistic vapriable, number of iterations of transferring activity through the intersected circles NoIteration, depth of inheritance $l$ expressed by levels of the inheritance tree.

**Step 1:** Set the counters:

LocCounter0, LocCounter1 $= 0$,
$w = 0$ , counter of obtained places $p_i$ by using $\alpha^{-1}$;
StepCounter $= 1$.

Compute value of radius $r'$ using function $h$ and the following formula:

$$r' = \lceil k * r \rceil, \quad k = \frac{m}{z} + \frac{1}{z+1}$$

**Step 2:** **IF** $d_x \in D$ **THEN**

Apply inverse function $\alpha^{-1}$ for the concept $d_x \in D$ :
$\alpha^{-1} : d_x \mapsto p_x$. For the place $p_x \in P$, set the corresponding component of the marking vector $\mu_{00}$ to 1.
**IF** $d_x \in C$ **THEN GOTO** Step 8.

**ELSE**

**IF** $d_x \notin C$ **THEN**

Generate the answer: "The concept $d_x$ is an unknown concept!" and stop the procedure.

**Step 3:** Find the corresponding location at the associative level:
$\omega : d_x \mapsto x'$, and by means of the group indicator, the name of location group $NGC_k$ is determined.

**Step 4:** Count all occupied and "non-retrieved" locations $y'_j$, $j = 1, 2, \ldots$ from the circle $O''(r', x')$.

**IF** StepCounter$=1$ **THEN**
LocCounter0$=$max $j$
**ELSE**
LocCounter1$=$LocCounter1$+$max $j$
Apply the inverse function $\omega^{-1}$ for each location: $\omega^{-1} : y'_j \mapsto c_j$, $j = 1, 2, \ldots$. By means of the group indicators, the names of location groups $NGC_j$ are determined.

**Step 5:** Apply the inverse function $\alpha^{-1}$ for every concept $c_j$, $j = 1, 2, \ldots$, and for $NGC_k$ and all $NGC_j$ for which $\alpha^{-1}$ is defined. Define all initial markings:

$\mu^s_{0t} = (a^s_1, a^s_2, \ldots, a^s_k)$, where $k = |P|$, $|\,|$ denotes cardinality of the set $P$ is a set of places, $h$ is the number of obtained places $p_i$ by using $\alpha^{-1}$ for the circle $O''(r', x')$, $s$ is an index which has value: $0 \le s \le h$, $w$ is the total number of obtained places in all steps of transferring activity, $t$ is an index which has the value: $w + 1 \le t \le w + h$. An initial marking vector $\mu^s_{0t}$, has only one nonzero component which corresponds to the place $p_i$:

$$\alpha'_j = \begin{cases} 1 & \text{for } j = 1 \\ 0 & \text{for} j \neq 1 \end{cases},$$

and the $j$-th component of the initial marking vector can be denoted as: $a^s_j = \mu^s_{0t}(j)$.

Increment counter: $w = w + h$

**Step 6:** Increment counter: StepCounter $=$ StepCounter$+1$

**Step 7:** **WHILE** StepCounter <= NoIteration **DO**

**WHILE** LocCounter0 > 0 **DO**

Set each of LocCounter0 locations $y'_j$ as the new center $x'$ of the new circle $O''$ and perform steps 4 and 5. Decrement counter: LocCounter0=LocCounter0–1

**END WHILE**

Set counters:

LocCounter0 = LocCounter1,
LocCounter1 = 0,
StepCounter = StepCounter + 1.

**END WHILE**

**Step 8:** For each initial marking $\mu^s_{0t}$ and for the initial marking $\mu_{00}$ construct $l$ levels of inheritance trees.

**Step 9:** Form inheritance paths and interpret them as the conjuction of inheritance assertion.

## 4. Example

By the next example we would like to show, though only partially, how the combination of inheritance procedures works at associative and semantic levels. We want to get an answer from the HETHI knowledge base model about the concept 'sparrow'. The main fact is that the concept 'sparrow' is not in $D$ but it is stored at the associative level.

The *KRP* levels of HETHI are defined as follows:

$$\begin{aligned} D &= D1 \cup D2 \cup D3 \\ &= \{\text{Mouse, Owl, Plumage, Egg,Nest,} \\ &\quad \text{Straw, Scarecrow, Blackbird,} \\ &\quad \text{Crow, Worm, Corn, Bread}\} \\ &\quad \cup \{\text{MAMMAL, BIRD}\} \cup \{\text{Black}\}. \\ \Sigma &= \Sigma1 \cup \Sigma2 \cup \Sigma3 \\ &= \{\text{is\_a}\} \cup \{\text{has\_colour}\} \\ &\quad \cup \{\text{eat, can\_fly, procreate, live\_in\_a,} \\ &\quad \text{scare, made\_of\_a, protected\_of\_a}\}. \end{aligned}$$

$$P = \{p_1, p_2, \ldots, p_{15}\},\ T = \{t_1, t_2, \ldots, t_{18}\}$$

$$\begin{aligned} I(t_1) &= \{p_2\} \quad O(t_1) = \{p_1\} \\ &\quad \alpha : p_1 \mapsto \text{MAMMAL} \quad \beta : t_1 \mapsto \text{is\_a} \\ I(t_2) &= \{p_3\} \quad O(t_2) = \{p_2\} \\ &\quad \alpha : p_2 \mapsto \text{Mouse} \quad \beta : t_2 \mapsto \text{eat} \\ I(t_3) &= \{p_3\} \quad O(t_3) = \{p_4\} \\ &\quad \alpha : p_3 \mapsto \text{Owl} \quad \beta : t_3 \mapsto \text{is\_a} \\ &\quad \vdots \\ I(t_{16}) &= \{p_{14}\} \quad O(t_{16}) = \{p_{12}\} \\ &\quad \alpha : p_{13} \mapsto \text{Straw}\ \beta : t_{16} \mapsto \text{protected\_by\_a} \\ I(t_{17}) &= \{p_{15}\} \quad O(t_{17}) = \{p_{14}\} \\ &\quad \alpha : p_{14} \mapsto \text{Corn}\ \beta : t_{17} \mapsto \text{made\_of\_a} \\ I(t_{18}) &= \{p_{11}\} \quad O(t_{18}) = \{p_{15}\} \\ &\quad \alpha : p_{15} \mapsto \text{Bread}\ \beta : t_{18} \mapsto \text{eat} \end{aligned}$$

The part of HETHI knowledge base is shown in Figure 9. The associative level of HETHI is depicted at the bottom of Figure 9.

The inheritance algorithm described in Section 3 is performed as follows :

**Input:** *sparrow*, linguistic variable = minorly similar , NoIteration = 1, $l = 2$

**Step 1:** Set the counters: LocCounter0, LocCounter1=0, $w = 0$, StepCounter=1 According to the expressed linguistic variable and Table 1. the value of

the measure of similarity is $m = 5$;
coefficient is

$$k = \frac{5}{7} + \frac{1}{7+1} = 0.839;$$

radius $r'$ is $r' = \lceil 0.839 * 31 \rceil = 26$.

**Step 2:** *sparrow* $\in C$ and *sparrow* $\notin D$

**Step 3:** $\omega(sparrow) = x'$, where $x'$ is the location at the associative level of HETHI. $NGC_k$ = BIRD.

**Step 4:** By counting locations from the circle $O''(r', x')$

LocCounter0=4 (StepCounter=1),

and retrieving their contents by applying the inverse function $\omega^{-1}$, the

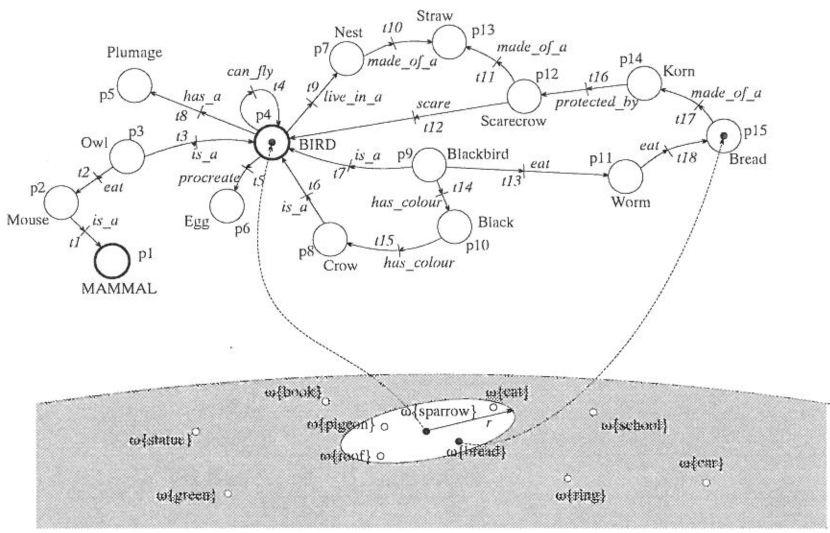



*Fig. 9.* The model of the HETHI knowledge base.

following concepts and $NGC_j$, are obtained (see Figure 9).

| concept | $NGC_j$ |
|---|---|
| bread | FOOD |
| pigeon | BIRD |
| cat | DOMESTIC_ANIMAL |
| roof | PART_OF_BUILDING |

**Step 5:** The inverse function $\alpha^{-1}$ is defined for the concept "bread" and for the name of location group $NGC_k$ = BIRD.

$$\alpha^{-1} : \text{bread} \mapsto p_4$$
$$\alpha^{-1} : \text{BIRD} \mapsto p_{15}.$$

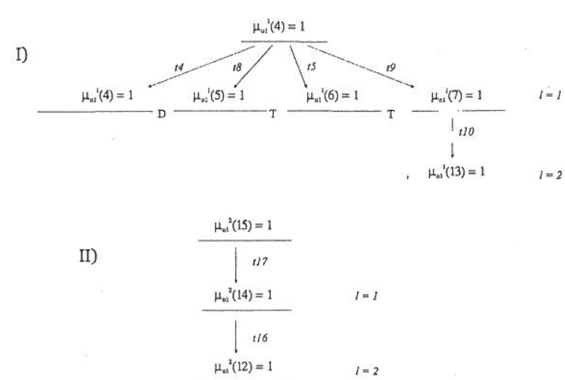
The inverse function $a^{-1}$ is not defined for concepts "pigeon", "cat", "roof" and for the following names of the location groups FOOD, DOMESTIC_ANIMAL and PART_OF_BUILDING.

Initial markings are obtained:

$\mu_{01}^1 = (a_1^1, a_2^1, \ldots, a_{15}^1)$, $\mu_{01}^1(4) = 1$,
$\mu_{01}^1(i) = 0$ for all $i = 1, 2, \ldots, 15$, $i \neq 4$
$\mu_{01}^2 = (a_1^2, a_2^2, \ldots, a_{15}^2)$, $\mu_{01}^2(15) = 1$,
$\mu_{01}^2(i) = 0$ for all $i = 1, 2, \ldots, 14$

**Step 6:** StepCounter = StepCounter + 1 = 2
StepCounter > NoIteration so Step 7 is skipped.

**Step 8:** Two 2-level inheritance trees are constructed for initial markings $\mu_{01}^1$ and $\mu_{01}^2$:




**Step 9:** Inheritance paths are formed for both inheritance trees:

I) BIRD live_in_a Nest.
BIRD can_fly .
BIRD procreate Egg.
BIRD has_a Plumage.
Nest made_of_a Straw.

II) Bread made_of_a Corn.
Corn protected_by Scarecrow.

| | | | |
|---|---|---|---|
| *** ARP *** | | | |
| sparrow | is a member of | BIRD | |
| sparrow | associate to | pigeon (member of BIRD) | |
| | | bread (member of FOOD) | |
| | | roof (member of PART_OF_BUILDING) | |
| | | cat (member of DOMESTIC_ANIMAL) | |
| ** IP *** | | | |
| sparrow | <–>* BIRD | live_in_a | Nest. |
| | | can_fly | . |
| | | procreate | Egg. |
| | | has_a | Plumage. |
| Nest | | made_of_a | Straw. |
| Bread | | made_of_a | Corn. |
| Corn | | protected_by_a | Scarecrow. |

So, finally, the HETHI response to the concept sparrow is given above.

## 5. Conclusion

We have proposed a model of heterogeneous hierarchical knowledge base HETHI. It consists of one level Kanerva-like sparse distributed memory (SDM) and multilevel knowledge base designed by the KRP knowledge representation scheme based on the Petri net theory. The lowest level of the HETHI, implemented by means of Kanerva-like SDM, performs associative retrieval information process and supports inheritance procedure at HETHI's higher levels. The enhanced inheritance procedures based on cooperation of SDM and KRP are developed and tested for this model of knowledge base. A model of heterogeneous hierarchical knowledge base can be used as a building block of the knowledge base at a description-generator level of the complex computer vision systems.

## References

[1] RAO A. R. and JAIN R., Knowledge Representation and Control in Computer Vision Systems, IEEE Expert, Vol. 3, No. 1, 1988, pp. 64–79.

[2] FIKES R. and KEHLER T., The Role of Frame-Based Representation in Reasoning, Comm of the ACM, Vol. 28, No. 9, September 1985, pp. 904–920.

[3] H. G. BARROW and J. M. TENENBAUM, Computational Vision, Proceed. of the IEEE, Vol. 69, No. 5, May 1981, pp. 572-579.

[4] RIESEMAN E. and HANSON A. R., A Methodology for the Development of General Knowledge- Based Vision Systems, in Vision, Brain and Cooperative Computation (eds. Arbib M. A., Hanson A. R.) , The MIT Press, Cambridge, 1987, pp. 285–328.

[5] D. MARR, Vision, Freeman And Co., San Francisco, 1982.

[6] V. D. HUNT, Smart Robots, Chapman and Hall, 1985.

[7] KANERVA P., Sparse Distributed Memory, The MIT Press, Cambridge, 1988.

[8] RIBARIĆ S., Knowledge Representation Scheme Based on Petri Net Theory, Int. Journal of Pattern Recognition and Artificial Intelligence, Vol. 2, No. 4, 1988, pp. 691–700.

[9] RIBARIĆ S. and PAVEŠIĆ N., Declarative and Procedural Knowledge Interaction in Robot Vision System, Image Processing and Stereo Analysis, Proceed. of the Slovenian-German Workshop, Erlangen, Dec. 3, 1992, pp. 3–71.

[10] MURATA and PETRI NETS, Properties, Analysis and Applications, Proceed of the IEEE, Vol. 77, No. 4, April 1989, pp. 541–580.

[11] PETERSON J. L., Petri Net Theory and Modelling of Systems , Prentice-Hall Inc. 1981.

[12] RIBARIĆ S., Inheritance in Knowledge Representation Scheme Based on Petri's Net Theory, Pattern Recognition and Image Analysis, Vol. 3, No. 3, July — September 1993, pp. 300–304.

[13] RIBARIĆ S., An Inheritance Algorithm in the Knowledge Representation Scheme, Proceedings of the 15th Int. Conference of Information Technology Interfaces ITI'93, June 1993, pp. 237–244.

[14] SHASTRI L., A Connectionist Approach to Knowledge Representation and Limited Inference, Cognitive science 12, 1988, pp. 331–392.

[15] TOURETZKY D. S., The Mathematics of Inheritance Systems, Pitman, London, 1986.

[16] FAHLMAN S. E., NETL A System for Representing and Using Real-World Knowledge, The MIT Press, Cambridge, 1979.

[17] MYLOPOULOS J. and LEVESQUE H. J., An Overview of Knowledge Representation in On Conceptual Modelling (eds. Brodie et al.), Springer–Verlag, 1984, pp. 3–17.

*Contact address:*
Slobodan Ribarić and Berislav Lastrić
Faculty of Electrical Engineering
University of Zagreb, Unska 3, Croatia
e-mail: slobodan.ribaric@etf.hr

SLOBODAN RIBARIĆ received the Dipl.-Ing., M. Sc. and Doctorate degrees all in Electrical Engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia in 1974, 1976, and 1982, respectively. He is Full Professor of Computer Science and Engineering at the Faculty of Electrical Engineering, University of Zagreb, Croatia. His interests include Computer Architecture, Image Processing, Computer Vision and Pattern Recognation. Prof. Ribarić is author of more than seventy papers. He is also the author of three books and co-author of one. Ribarić is a member of the IEEE, and the Croatian Association of Electrical Engineers.

BERISLAV LASTRIĆ is postgraduate student at Faculty of Electrical Engineering, University of Zagreb, the specialization Core of computer sciences. He works, as a system engineer, on projects of organization of computer systems, networks and POS systems in a commercial trade. His interests include artificial intelligence, an architecture of OO RDBMS, a computer graphic and a multimedia. B Lastrić received the B.E., the branch Computer sciences, in 1993. for a work titled "Usage of the idea of the P. Kanerva's Sparse Distributed Memory model".