A Storage Algorithm for a Kanerva-like Memory Model

Slobodan Ribarić, Darijan Marčetić
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
{slobodan.ribaric, darijan.marcetic}@fer.hr

Abstract:
This paper presents a Kanerva-like sparse distributed memory (SDM) as the first level of a heterogeneous, hierarchical knowledge-base model. A concept-storing algorithm for storing concepts, based on the similarity between them, expressed by the value of a linguistic variable is described. The algorithm is demonstrated by using the concepts, with the associations among them obtained using psychological experiments.

Keywords: Associative level, Sparse Distributed Memory (SDM), Storage algorithm.

I. INTRODUCTION

Complex intelligent systems require a huge amount of knowledge, as well as some efficacious mechanisms for manipulating this knowledge, in order to perceive, organize and summarize observations and stimuli obtained from the real world. They also require a large amount of information from the problem domain to support a useful degree of problem-solving ability. In many cases the knowledge bases in intelligent systems are organized as structured organizations depicted as a succession of different levels of representations [1], [2], [3] mirroring the natural structure of knowledge from the real world (for example, the hierarchical knowledge organization of computer-vision systems [3], and the hierarchical organization of a knowledge base for conscious software agents [4]).

Our research goal is to develop a heterogeneous, hierarchical knowledge-base model consisting of multi-levels, starting from an associative level, through to a semantic, rule-based and description-generator level as the top level in the hierarchy. The main reason for such a heterogeneous, hierarchical knowledge-base model lies in some of the limitations and drawbacks of the inference processes of the “pure” knowledge-base models, such as semantic networks, rule-based systems, frame-based schemes and logic-based models. The heterogeneous, hierarchical knowledge-base model has to support efficient information storage and retrieval, as well as reasoning in complex information systems.

In this paper we describe only the first level of the knowledge-base model – an associative level, which is based on a modification of Kanerva’s Sparse Distributed Memory (SDM) [5]. For such a model the concept-storing algorithm is proposed.

The SDM has had many successful implementations as an associative memory for intelligent systems. In [6] a heterogeneous, hierarchical knowledge-base model called HETHI is described. It consists of one level of the Kanerva-like SDM that performs the associative retrieval information process and supports the initialization of the inheritance process at higher levels – the semantic and rule-based levels.

In [4], an enhanced version of the SDM, augmented with the use of genetic algorithms, as an associative memory in a ‘conscious’ software agent CMattie is described. CMattie, as an intelligent agent, interacts with seminar organizers via email in natural language and is responsible for emailing seminar announcements in an academic department. The SDM is a key ingredient in the complex agent architecture that implements global workspace theory, a psychological theory of consciousness and cognition. In this architecture, the SDM, as the primary memory for the agent, provides associations with incoming precepts.

In [7] the authors describe in detail the IDA (Intelligent Distribution Agent) architecture of autonomous software agents as a cognitive model of human cognition that employs the SDM as a working and associative memory. The IDA heterogeneous, hierarchical architecture is composed of a number of different levels, each devoted to a particular cognitive process. An example of a system based on the IDA, used for gathering logistical and medical information from a patient for later use by the triage nurse, is given in [2].

In [8] the SDM is used for multilevel cognitive tasks. The SDM memory is organized to link low-level information and high-level correlations. The authors conducted experiments that combined the pattern recognition of individual English characters followed by the assignment of ‘meaning’ to a string by giving it a Hebrew translation.

II. THE KANERVA-LIKE MEMORY MODEL

The associative level of the knowledge base is implemented with the use of some concepts related to the addressing mechanism of the Kanerva’s SDM [5]. The SDM model is defined in the space of $\{0, 1\}^n$, $n \in \mathbf{N}$, where **N** is a set of the natural numbers, elements of which are n-dimensional vectors with binary components. These vectors are represented as points in an n-dimensional space. The number of points in an n-dimensional space is $N = 2^n$. N is also used for naming the space itself, i.e., N-space. The main feature of N-space is its distribution, defined on the basis of the distances among the points. The distance $d(x,y)$ between two points x and y in N-space is defined as the number of corresponding vector components at which they differ, known as the Hamming distance. The distance is, by definition, an integer number in the range from 0 to n. The distance $d(x,y)$ can be used to express the similarity of the points x and y. Two points in N-space that are close to each other are more similar. The number of points from N-space at a distance of exactly d from an arbitrary point in N-space is equal to the binomial coefficient $\binom{n}{d}$. The distribution of N-space is thus the binomial distribution $N(d)$ with the parameters n (number of dimensions) and p = 0.5 (the probability of each binary component of the point being 0 or 1). The concept of the address region is used at the associative level. The address region of an arbitrary address location x in N-space is defined as a circle O with the radius r and the centre x. The address region of x contains a set of points that are at most r bits from x: $O(r,x)=\{y \mid d(x,y)\le r\}$.

The basic characteristics of the above-described SDM model are: the *similarity* and *sparseness* of the memory. *Similarity*, as mentioned above, is based on the distance between points. *Sparseness* is derived from the fact that the actual number of storage locations used is very few compared with 2^n, n >>1. The storage locations are distributed randomly in N-space and a unique address is assigned to each storage location. Even for a relatively small dimensionality of N-space (for example, n = 100) an exorbitant number of possible locations (N=2^{100}) exists. Let us suppose that only a fraction of the possible address space (for example, N' = 1,000,000) is available and points are randomly distributed over the entire address space. Such a type of space is called a sparse memory. Correspondingly, addressing N-space is now reduced to addressing the locations of the subspace N', N' ⊂ N called N'-space. The address location in N'-space is represented by an n-dimensional address vector in the same way as in N-space.

III. RELATION OF A CONCEPT TO A POINT IN N'-SPACE AND A CONCEPT GROUP ORGANIZATION

Let C be a set of abstract objects or concepts from the real world, which is the subject of the modelling. An element from C can be represented with one or more points in N'-space. A point in N'-space is called a location when N'-space is represented as the SDM model. The concepts are clustered into groups of concepts based on a user's knowledge and/or intuition. A concept can belong to one or more groups of concepts. Groups of concepts are used because it is natural for humans to organize similar concepts based on their characteristics.

Let us denote a group of points in N'-space by G_i, i =1, …, g (indicated by the name of the group NGC_i) where g is the total number of groups. Each group of points is represented by a uniform, random sample of N'-space. Note that $G_i \cap G_j = \varnothing$ for $i \neq j$, and that each point in N'-space is indicated by its unique address. The number of points in every group is equal to N'/g.

The basic assumption for the proposed model is that the number of points in N'-space is much larger than the number of concepts. For mapping a concept $c_i \in C$ into a set of points in N'-space, the concept-address list is used.

A concept-address list for a concept $c_i \in C$, which is stored in the group with the name NGC_j, is represented by the 3-tuple: $\left(c_i, NGC_j, \left(a_1^{i,j},..., a_s^{i,j}\right)\right)$, where the s-tuple $\left(a_1^{i,j},..., a_s^{i,j}\right)$ is an ordered list that contains the addresses of points in N'-space; the first super index of a denotes the corresponding concept, the second super index denotes the group G_j and the s denotes the total number of copies of the concept c_i in the group G_j. When the same concept c_i belongs to different groups NGC_k, NGC_l, …, then for each group there is a corresponding concept-address list: $\left(c_i, NGC_k, \left(a_1^{i,k},..., a_t^{i,k}\right)\right)$, $\left(c_i, NGC_l, \left(a_1^{i,l},..., a_v^{i,l}\right)\right)$, … . Note that the concept-address lists are a rudimentary implementation of Kanerva's basic idea of *the word to focus transformation* defined by mapping of the physical signals to sensory features [5].

A. Similarities between concepts

A linguistic variable L is used to express the similarity among the concepts. The values of the linguistic variable L are from the following set {*minimally, minorly, more-or-less, moderately, considerably, very, extremely*}. The values of the linguistic variable L can be transformed to the Hamming distance between two points in N'-space. For example, in terms of the description of similarity with a value of the linguistic variable L, two concepts described as "*very*" similar will have a smaller distance between the corresponding points in N'-space than the concepts described as "*minorly*" similar.

IV. THE CONCEPT-STORING ALGORITHM

The process of storing a concept in the Kanerva-like memory model can be described as follows. The user specifies a concept $c_i \in C$, where C is a set of concepts, with the name of its group NGC_k, where the concept will be stored. Optionally, the user can specify a measure of similarity, expressed by a value of the linguistic variable L, to another already-stored concept $c_j \in C$, and the name of its group NGC_l, where $l \neq k$ or $l = k$.

In the first case, when $c_i \in C$ and the name of its group NGC_k are given, the concept will be stored in the randomly chosen free location from its group NGC_k.

In the second case, a new concept c_i will be stored in a location or locations from the group with the name NGC_k, based on the measure of similarity between c_i and c_j.

The above process is specified by the following *concept-storing algorithm*:

Input: A concept $c_i \in C$, the name of its group NGC_k, and (optionally) the already-stored concept $c_j \in C$, the name of its group NGC_l, where k and l can be equal, and the measure of similarity between the concepts c_i and c_j expressed by the value of the linguistic variable L.

IF NGC_k is a new group name **THEN** assign NGC_k to the unused group of locations G_k .

Case 1: Only the concept c_i is given

1.1) **IF** the concept c_i is new in the group with the name NGC_k **THEN** store the concept c_i to the randomly chosen free location x_1 defined by the address $a_1^{i,k}$ in the group with the name NGC_k. Add the concept-address list $\left(c_i, NGC_k, \left(a_1^{i,k}\right)\right)$ to the set of concept-address lists. **END**.

1.2) **IF** the same concept c_i has already been stored in the group with the name NGC_k **THEN** store the "new" concept c_i to the randomly chosen free location x_{s+1} defined by the address $a_{s+1}^{i,k}$ in the group with the name NGC_k. Update the concept-address list $\left(c_i, NGC_k, \left(a_1^{i,k}, \ldots, a_s^{i,k}\right)\right)$ for the concept c_i from the group G_k with the name NGC_k to $\left(c_i, NGC_k, \left(a_1^{i,k}, \ldots, a_s^{i,k}, a_{s+1}^{i,k}\right)\right)$. **END**.

Case 2: The **s**imilarity between the concepts c_i and the already stored concept c_j is defined with a value of the linguistic variable *L*, determine the free location(s) for storing the concept c_i as follows:

2.1) Calculate $r' = \lceil r_p \cdot K \rceil$, where $\lceil\ \rceil$ denotes the ceiling of the expression and

- radius r_p, called the basic radius, defines the x'-centred circle $O(r_p, x')$; $x' \in$ N', which contains on average *p* proportion of points y' in N'-space that satisfy the relation $O(r_p, x') = \{ y' \mid d(x', y') \le r_p \}$. The radius r_p is obtained from TABLE I [5] based on the number of dimensions *n* and the proportion *p* of N'-space within r_p bits of point *x*'. The proportion *p* is determined experimentally and its value is between 10^{-3} to 10^{-6},
- radius multiplicator $K = f(m, z, \alpha, \beta, r_p, n)$; based on the experiments *K* was selected as: $K = \alpha \cdot \frac{m}{z-1} + \beta$, where β; $1 \le \beta \le \frac{n}{2 \cdot r_p}$ defines the minimum value of *K* (when the concepts are extremely similar), while the parameter α ; $0 < \alpha < \frac{n}{2 \cdot r_p} - \beta$, determines the incremental step of the radius according to the dissimilarity measure *m* obtained by the bijective function $h: L \rightarrow M$ (TABLE II). The value of *m* is a discrete value from a set $M = \{0, 1, \ldots, z\text{-}1\}$, where *z* is the total number of values of the linguistic variable *L* (TABLE II). Recall that *n* is the number of dimensions of N'-space.

2.2) Retrieve the concept-address list $\left(c_j, NGC_l, \left(a_1^{j,l}, \ldots, a_u^{j,l}\right)\right)$ from a set of stored concept-address lists.

2.3) For every address $a_e^{j,l}$, $e = 1, \ldots, u$, from $\left(c_j, NGC_l, \left(a_1^{j,l}, \ldots, a_u^{j,l}\right)\right)$, store the concept c_i to the free location in the group G_k with the address $a_b^{i,k}$, b =1, …, *u*, in such a way that $a_b^{i,k}$, satisfies that $\left| d\left(a_b^{i,k}, a_e^{j,l}\right) - r' \right|$ is close to zero, as much as possible, where | | denotes the absolute value.

2.4) **IF** the concept c_i is new in the group with the name NGC_k **THEN** add the concept-address list $\left(c_i, NGC_k, \left(a_1^{i,k}, \ldots, a_u^{i,k}\right)\right)$ to the set of concept-address lists. **END**.

2.5) **IF** the concept c_i was already stored in the group NGC_k **THEN** update the concept-address list $\left(c_i, NGC_k, \left(a_1^{i,k}, \ldots, a_s^{i,k}\right)\right)$ for the concept c_i from the group with the name NGC_k to $\left(c_i, NGC_k, \left(a_1^{i,k}, \ldots, a_s^{i,k}, a_{s+1}^{i,k}, \ldots, a_{s+u}^{i,k}\right)\right)$, where $a_{s+1}^{i,k}, \ldots, a_{s+u}^{i,k}$ correspond to the addresses $a_b^{i,k}$, *b* = *1*, …, *u*, of the locations where the concept c_i is stored. **END**.

An assumption for the *concept-storing algorithm* is that the number of free locations in every group is much higher than the number of stored concepts.

TABLE I
RADII r_p OF *x*-CENTRED CIRCLES THAT ENCLOSE A SPECIFIED PROPORTION *p* OF N'-SPACE [5].

Proportion *p* of the N'-space within r_p, bits of point *x*	Number of dimensions *n*		
	n=100	*n*=1,000	*n*=1,000,000
	r_p	r_p	r_p
0.000000001	20	405	4,700
0.00000001	33	411	4,720
0.0000001	24	418	4,740
0.000001	26	425	4,760
0.00001	29	433	4,790
0.0001	31	441	4,810
0.0005	34	448	4,840
0.001	35	451	4,850
0.01	38	463	4,880
0.01	44	480	4,940
0.25	47	489	4,970
0.5	50	500	5,000

TABLE II
MAPPING OF VALUES OF THE LINGUISTIC VARIABLE *L* TO THE MEASURE OF DISSIMILARITY *m*.

$h: L \rightarrow M, z=7$	
Values of the linguistic variable *L*	Measure of the dissimilarity, *m*
extremely	0
very	1
considerably	2
moderately	3
more-or-less	4
minorly	5
minimally	6

A. Example

Let us suppose that N'-space is created, where N'-space is a subset of N-space; $N=2^n$; n = 100. Let the number of locations of the SDM be N'=1,000,000, the number of groups g = 20, p = 0.0001, α is 0.5 and β is 1, and the number of values of the linguistic variable L is z = 7 (TABLE II).

The number of storage locations in every group is equal to N'/g = 50,000. Based on selected values of p and n, the radius r_p= 31 is determined (see TABLE I).

To demonstrate the *concept-storing algorithm* we used concepts with associations among them obtained with the psychological experiment described in [9]. The goal of this psychological experiment was to determine the associative connections between concepts based on the frequency of the answers of 1000 examinees.

From this experiment two concepts were selected, c_1 = "*Needle*" and c_2 = "*Thread*", with the similarity among them defined as "*very*" similar. Note that the similarity is estimated by the authors based on the frequency of the answers given in [9]. First, let us suppose that the concept c_1 = "*Needle*" from the group with the name NGC_1 = "*Tools*" is already stored in the three locations in N'-space with the addresses (100 bit vectors, represented as the hex number):

$a_1^{1,1}$ = 208257A1CD7121EA3DBBB6842,

$a_2^{1,1}$ = E7E17CEBF0E1E2C9A2153024B,

$a_3^{1,1}$ = D091BAE3E76B918D623049E5C.

Note that the addresses are generated by the program simulator described in Section V.

Storing of a new concept c_2 = "*Thread*", belonging to a new group with name NGC_2 = "*Fabric*", with similarity to the concept c_1 = "*Needle*", expressed with the value of the linguistic variable L equal to "*very*" similar (m = 1; see TABLE II), is processed as follows:

NGC_2 = "*Fabric*" is a new group name, so assign NGC_2 to the unused group G_2.

Case 2: The similarity between the concepts c_2 = *"Thread"* and c_1 = *"Needle"* is defined as *"very"* similar, so free locations in the group with the name NGC_2 = "*Fabric*" for the storage the concept c_2 are determined as follows:

2.1) $K = \alpha \cdot \frac{m}{z-1} + \beta = 0.5 \cdot \frac{1}{7-1} + 1 = 1.083$, r_p = 31 and $r' = \lceil K \cdot r_p \rceil$ = 34.

2.2) The concept-address list $\left(Needle, NGC_1, \left(a_1^{1,1}, a_2^{1,1}, a_3^{1,1}\right)\right)$ is retrieved from the set of concept-address lists.

2.3) The concept c_2 = *"Thread"* is stored in three free locations in the group G_2 with the addresses:

$a_1^{2,2}$ = 39E315911D11116E1915F5410,

$a_2^{2,2}$ = B752BFEBE0414098242F22C8B,

$a_3^{2,2}$ = F8185BC0A400B185DB316961B

so that the values of the terms $\left|d\left(a_1^{2,2}, a_1^{1,1}\right) - r'\right|$, $\left|d\left(a_2^{2,2}, a_2^{1,1}\right) - r'\right|$ and $\left|d\left(a_3^{2,2}, a_3^{1,1}\right) - r'\right|$ are as close to zero as possible.

2.4) The concept c_2 = "*Thread*" is new in the group with the name NGC_2 ="*Fabric*", so add the concept-address list $\left(Thread, NGC_2, \left(a_1^{2,2}, a_2^{2,2}, a_3^{2,2}\right)\right)$ to the set of concept-address lists.

2.5) No action.

We can see that the Hamming distances between the addresses of the locations in N'-space where the concepts "*Needle*" and *"Thread"* lie are: $d(a_1^{2,2}, a_1^{1,1})$ = (208257A1CD7121EA3DBBB6842, 39E315911D11116-E1915F5410) = 34, $d(a_2^{2,2}, a_2^{1,1})$ = (E7E17CEBF0E1-E2C9A2153024B, B752BFEBE0414098242F22C8B) = 34 and $d(a_3^{2,2}, a_3^{1,1})$ = (D091BAE3E76B918D623049E5C, F8185BC0A400B185DB316961B) = 34, and $\left|d(a_u^{2,2}, a_u^{1,1}) - r'\right|$, u = *1*, *2*, *3*, is equal to 0 for every corresponding address-pair.

V. PROGRAM SIMULATOR

The program simulator developed in Matlab supports the following main functionalities (see Fig. 1):

(i) the Kanerva-like memory model (SDM),
(ii) the concept-storage,
(iii) the concept-similarity estimation,
(iv) the concept-address list,
(v) the GUI (graphical user interface).

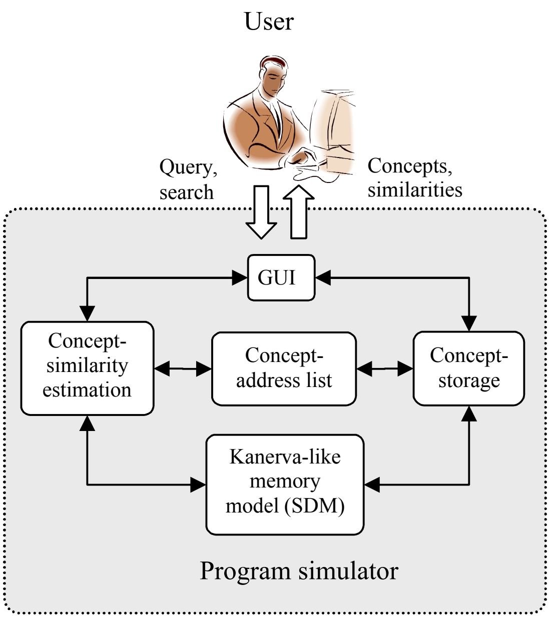

Fig. 1. Main modules of the developed program simulator.

The Kanerva-like memory model (SDM) module is designed according to the functionalities described in Sections II and III. The concept-storing module is used to store the concepts in N'-space by means of the *concept-storing algorithm*. The concept-similarity estimation module is used to estimate the similarity between the stored concepts based on the distance in N'-space (which is not the subject of this paper). The concept-address module is used as a rudimentary implementation of *the word to focus transformation* and it updates the concept-address lists. A graphical user interface (GUI) is employed for a user-friendly interaction with the system. It is important that the GUI is both intuitive and simple to use. The GUI is also used to display the results and it enables the manual control of every step in the algorithms employed. The visual appearance of the GUI used in our system is shown in Fig. 2.

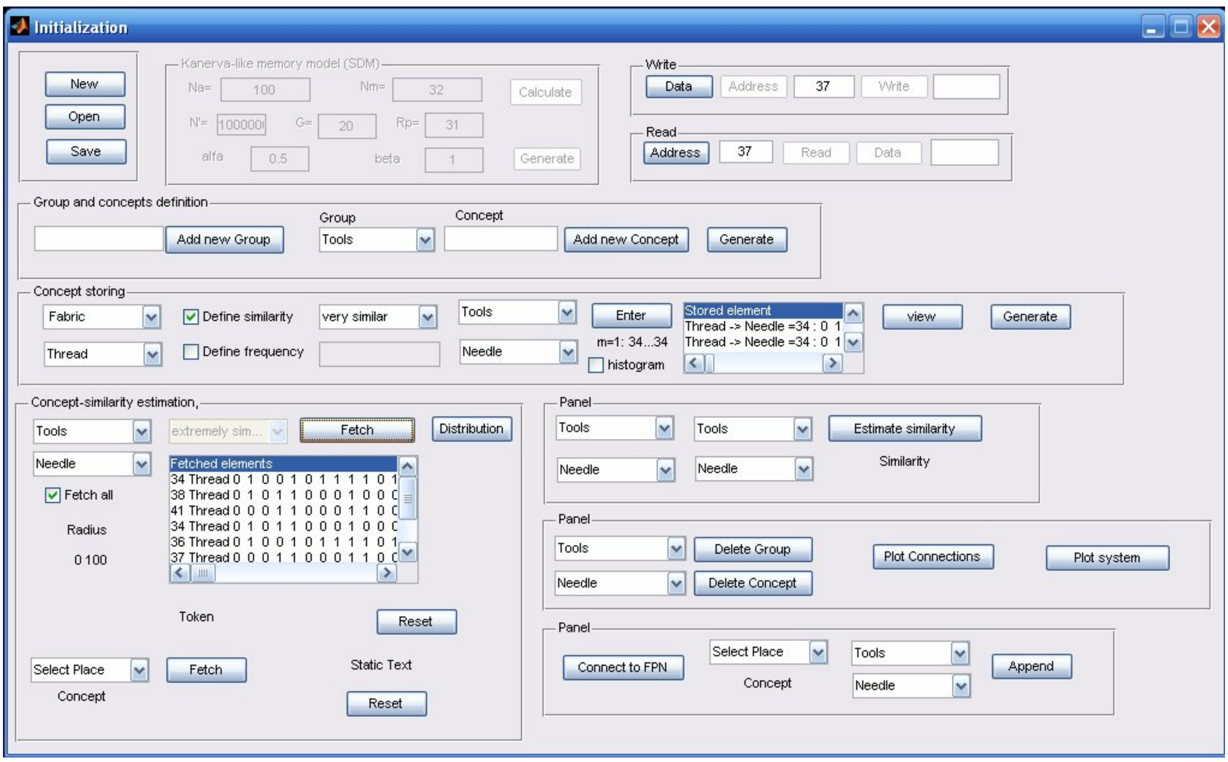

Fig. 2. Visual appearance of graphical user interface (GUI).

VI. CONCLUSION

This paper presents a formal model of the first level of the heterogeneous, hierarchical knowledge-base model – an associative level implemented with Kanerva-like SDM. The *concept-storing algorithm* for storing concepts in N'-space based on the similarity between them expressed by a linguistic variable is proposed. Future work will involve the implementation of the heterogeneous, hierarchical knowledge-base model consisting of multi-levels, starting from an associative level, through to the semantic, rule-based and description-generator level as the top level in the hierarchy for the document and the World Wide Web search.

ACKNOWLEDGEMENTS

This work has been supported within the project "Theory, modeling and application of autonomy-oriented computing structures" (no. 036-0361935-1954); Ministry of Science and Technology, Croatia.

REFERENCES

[1] R. Sun, "Robust reasoning: integrating rule-based and similarity reasoning", *Artificial Intelligence*, vol. 75, p. 241-295, 1995.

[2] S. Franklin and D. Jones, "A Triage Information Agent (TIA) based on the IDA Technology", *AAAI Fall Symposium on Dialogue Systems for Health Communication*, Washington, DC, 2004.

[3] E. Rieseman and A. R. Hanson, *A Methodology for the Development of General Knowledge-Based Vision Systems, in Vision, Brain and Cooperative Computations*, The MIT Press, Cambridge, p 285-328, 1987.

[4] A. Ashraf and S. Franklin, "Sparse Distributed Memory for 'Conscious' Software Agents", *Cognitive System Research*, vol. 4, p 339-354, 2003.

[5] P. Kanerva, *Sparse Distributed Memory*, The MIT Press, Cambridge, 1988.

[6] S. Ribarić and B. Lastrić, "HETHI- Heterogeneous Hierarchical Knowledge Base", *Journal of Computing and Information Technology (CIT)*, vol. 2, p 167-179, 1994.

[7] S. Franklin and A. Graesser, "Modeling Cognition with Software Agents", *Proceedings of the 23rd Annual Conference of the Cognitive Science Society NA*, p. 301-306, 2001.

[8] L. M. Manevitz and Y. Zemach, "Assigning Meaning to Data: Using Sparse Distributed Memory for Multilevel Cognitive Tasks", *Neurocomputing*, vol. 14, p. 15-39, 1997.

[9] R. S. Woodworth, *Experimental Psychology*, Henry Holt & Co., New York, 1938.