Hierarchical Distributed Reasoning System for Geometric Image Generation

Nicolae Țândăreanu, Mihaela Verona Ghindeanu, Sergiu Andrei Nicolescu

University of Craiova, Romania
Department of Mathematics and Computer Science
A.I. Cuza St, No. 13, 200585
E-mail: ntand@rdslink.ro, mghindeanu@yahoo.com

Abstract: The concept of hierarchical reasoning system was introduced in [5], where an intuitive method to build such systems based on their inputs is given. In this paper we formalize several concepts which open a possible research line concerning the use of these structures. A hierarchical reasoning system $H$ is a directed graph organized on several levels such that each node of the level $j$ is a hyper-schema of order $j$. As a mathematical structure, $H$ is an abstract one and a special kind of formal computation is introduced. As a result of this computation we obtain a set $\mathcal{F}(H)$ of formulas. We explain what we understand by an interpretation of $H$ and define its corresponding semantical computation. By means of an interpretation $\mathcal{I}(H)$ for $H$ and applying the rules of the semantical computation, each element of $w \in \mathcal{F}(H)$ becomes some object $\mathcal{I}(w)$ of a given space. We exemplify these concepts and we show that for two distinct interpretations $\mathcal{I}_1(H)$ and $\mathcal{I}_2(H)$ for the same system $H$, a given formula $w \in \mathcal{F}(H)$ is transformed into a sentence $\mathcal{I}_1(w)$ of a natural language whereas $\mathcal{I}_2(w)$ is a geometric image. A short description of a Java implementation of a hierarchical system generating images is also given in a separate section. By examples we show that the mechanism introduced in this paper allows us to model the distributed knowledge. Finally several open problems are specified.

Keywords: semantic schema, interpretation, hyper-schema, distributed reasoning system, geometrical image generation

1 Introduction

Various kinds of mechanisms for image synthesis were presented and implemented on computer. The panel of the mathematical models for this subject includes the rewriting systems and graph-based models. Picture-processing grammars ([2]), picture grammars ([3]), stochastic grammars ([14]) and L-systems are some of the rewriting systems used to process images. The L-systems are a class of string rewriting mechanism originally developed by a biologist, A. Lindenmayer, in 1968 ([7]). The original emphases were on plant topology - spatial relations between cells or larger plant modules. The L-systems are a practical tool for generating fractal forms. Today these models are applied in architecture, physiology ([11]) and music. In order to interpret the L-system as music, LMUSe system ([9]) maps any of the turtle’s 3D movement, orientation directions (forward, up, and left), its drawing line length, and thickness into musical pitches, note durations and volume.

A great number of research works and practical implementations have confirmed the interest of mathematicians and computer scientists in developing and applying the methods of graph theory. These methods were applied to obtain new knowledge representation models and to process images. A very productive notion with large applications in knowledge representation is that of conceptual graph, a
notion introduced in literature by J.F.Sowa ([8],[10]). We can find several applications of the graph-based methods in [6] (low-level processing of digital images, learning algorithms for high-level computer vision and pattern recognition).

The concept of semantic schema was introduced in [11] as an extension of semantic networks. This structure is obtained by means of a labeled graph and a Peano algebra built over the edge labels. Since then many applications of this structure were presented (new semantics in logic programming, knowledge representation for intelligent dialog systems etc).

In [12] we defined a new mechanism for generating images similar with the edge rewriting in the way that both approaches can be used to define complex images based on some simple other images. In the mentioned paper the concept of Hierarchical Distributed Reasoning System was introduced. Each leaf of the system is given by a semantic schema. The other nodes are hyper-schemas ([12]). We presented an intuitive method to obtain geometrical images. The leaves represent the input of the system in semantic schemas and, by appending proper interpretations, they obtain the graphical illustrations of the received inputs. In this manner the leaves obtains the initial images. Then, at the upper levels, these images are combined by hyper-schemas to obtain complex images. We obtained a bottom-up method to obtain images from initiators.

In this paper we obtain the following results:

• Starting with the concept of Hierarchical Distributed and Reasoning System (HGR system) introduced in [12] in Section 3 we define a formal computation in such a structure. As a result of this computation we obtain a set $\mathcal{F}(H)$ of formulas for an arbitrary HDR system $H$. This is the formal computation in an HDR system.

• An HDR system $H$ is an abstract structure. In Section 4 we introduce the concept of interpretation for $H$. By means of an interpretation $\mathcal{J}(H)$ for $H$ each element of $\mathcal{F}(H)$ becomes some object of a given space. This gives the semantical computation. Both the formal and semantical computations are exemplified. We show that for two distinct interpretations $\mathcal{J}_1(H)$ and $\mathcal{J}_2(H)$ for the same system $H$ we can generate sentences in a natural language giving the reasoning conclusions and geometrical images respectively.

• A short description of a Java implementation of an HDR system is also given in Section 5.

• By examples we show that the mechanism introduced in this paper allows us to model the distributed knowledge.

• The last section contains the conclusions and future works. Several open problems are specified in this section.

2 Basic concepts

Consider a symbol $\theta$ of arity 2. A $\theta$-semantic schema ([11]) or shortly, $\theta$-schema is a system $S = (X, A_0, A, R)$, where:

- $X$ is a finite non-empty set of symbols named object symbols;
- $A_0$ is a finite non-empty set of elements named label symbols and $A_0 \subseteq A \subseteq \overline{A}_0$, where $\overline{A}_0$ is the Peano $\theta$-algebra generated by $A_0$;
- $R \subseteq X \times A \times X$ is a non-empty set of relations which fulfills the following conditions:
  1. $(x, \theta(u, v), y) \in R \Rightarrow \exists z \in X : (x, u, z) \in R, (z, v, y) \in R$
  2. $\theta(u, v) \in A, (x, u, z) \in R, (z, v, y) \in R \Rightarrow (x, \theta(u, v), y) \in R$
3. \[ \{ \alpha \mid \exists(x, \alpha, y) \in R \} = A \]

An element from \( R \cap (X \times A_0 \times X) \) is a regular arc of \( S \).

We denote by \( \text{Ded}(S) \) the least set satisfying the following properties ([13]):

- If \( (x, a, y) \in R \) then \( (x, y, a) \in \text{Ded}(S) \).
- If \( ([x_1, \ldots, x_k], u) \in \text{Ded}(S) \) and \( ([x_1, \ldots, x_i], v) \in \text{Ded}(S), i < k < r \) and \( \theta(u, v) \in A \) then \( ([x_1, \ldots, x_i], \theta(u, v)) \in \text{Ded}(S) \).

An element of \( \text{Ded}(S) \) is a deductive path of \( S \).

Let us consider the schemas \( S_1 = (X_1, A_{01}, A_1, R_1) \) and \( S_2 = (X_2, A_{02}, A_2, R_2) \). In the remainder of this section we describe a new structure which relieves a special kind of cooperation between \( S_1 \) and \( S_2 \).

If \( d_1 = ([x, \ldots, y], u) \in \text{Ded}(S_1) \) and \( d_2 = ([y, \ldots, z], v) \in \text{Ded}(S_{3-i}) \), where \( i \in \{1, 2\} \), then we say that \( d_1 \) is connected to right by \( d_2 \) or connected to left by \( d_1 \). We say that \( d_1 \) is connected by \( d_2 \) if \( d_1 \) is connected to right or to left by \( d_2 \).

We consider the sets of deductive paths \( L_1 \subseteq \text{Ded}(S_1) \) and \( L_2 \subseteq \text{Ded}(S_2) \). We say that \( L_1 \cup L_2 \) is a pairwise connected set of deductive paths if every deductive path of \( L_i \) is connected by some deductive path of \( L_{3-i} \).

For each \( i \in \{1, 2\} \) we consider a set \( V_i \) of symbols such that \( V_i \cap (A_1 \cup A_2) = \emptyset \). We consider also a set \( E_i \) such that \( E_i \subseteq X_i \times V_i \times X_i \). We consider also a bijective mapping \( g_i : E_i \rightarrow E_1 \) such that \( g_i(d) = (x, e, y) \), where \( d = ([x, \ldots, y], \theta(u, v)) \in L_1 \). This mapping transforms each deductive path \( ([x, \ldots, y], \theta(u, v)) \) from \( L_1 \) into a regular arc \((x, e, y)\). Shortly, we say that the path \( ([x, \ldots, y], \theta(u, v)) \) is designated by \((x, e, y)\). We can define now a cooperating structure of hyper-schemas.

A hyper-schema of order zero is a semantic schema. Consider the hyper-schemas \( S_1 \) and \( S_2 \) of order zero. A hyper-schema of order one over \( S_1 \) and \( S_2 \) is obtained by means of \( L_1 \) and \( L_2 \) is a \( \theta \)-schema \( S \) which includes the regular arcs obtained from \( L_1 \) and \( L_2 \) ([12]). We denote by \( \text{Hyp}_1([S_1, S_2]) \) the set of all hyper-schemas of first order over \( S_1 \) and \( S_2 \). In general we write \( S \in \text{Hyp}_k([S_1, S_2]) \) and we name \( S \) a hyper-schema of order \( k \) if \( S_1 \) and \( S_2 \) are hyper-schemas of order \( j \leq k - 1 \) and at least one of them has the order \( k - 1 \).

An HDR system ([12]) is the tuple \( H = (Q_1, Q_2, \ldots, Q_k) \) where \( k \geq 2 \) and

- \( Q_1 = \{S_1, \ldots, S_{n_1}\}, n_1 > 1 \), constitutes the first level of the system. The entities \( \{S_1, \ldots, S_{n_1}\} \) are hyper-schemas of order zero. The set \( Q_1 \) gives the leaves of \( H \).

- \( Q_2 = \{S_{n_1+1}, \ldots, S_{n_2}\}, n_2 \geq n_1 + 1 \), gives the second level of the system and \( S_{n_1+1}, \ldots, S_{n_2} \) are hyper-schemas of order 1. More precisely, for every \( m \in \{n_1+1, \ldots, n_2\} \) there are \( m_1, m_2 \in \{1, \ldots, n_1\}, m_1 \neq m_2 \) such that \( S_m \in \text{Hyp}_1([S_{m_1}, S_{m_2}]) \).

- For \( j \in \{3, \ldots, k\}, Q_j = \{S_{n_{j-1}+1}, \ldots, S_{n_j}\} \) represents the \( j \)-th level of the system, where \( n_j \geq n_{j-1} + 1 \). For every \( m \in \{n_{j-1}+1, \ldots, n_j\} \) there is \( m_1 \in \{n_{j-2}, \ldots, n_{j-1}\} \) and there is \( m_2 \in \{1, \ldots, n_{j-1}\} \) such that \( S_m \in \text{Hyp}_{j-1}([S_{m_1}, S_{m_2}]) \).

### 3 Formal computations in HDR Systems

Suppose that \( H = (Q_1, Q_2, \ldots, Q_k) \) is an HDR system. The components of \( H \) are the hyper-schemas \( S_1, \ldots, S_{n_k} \). We can visualize \( H \) as a graph structure. In order to obtain this structure we represent each hyper-schema by a node and we draw two directed arcs from \( S_r \) to \( S_j \) and to \( S_m \) if \( S_r \in \text{Hyp}_p([S_j, S_m]) \) for some \( p \). The structure obtained in this manner is not a tree. This can be observed in Figure 1: there are two distinct paths from \( S_7 \) to \( S_2 \) and there is no root of this structure.
By the assumptions of the previous section we have
\[ R \]
\[ \text{denote} \]
\[ R \]
\[ \text{have the following property:} \]
\[ (\text{languages, the set} Z \text{and right parentheses, the square brackets [ and ]}, \text{the symbol} h \text{, obviously we have the following property:} \]
\[ \text{We denote} \]
\[ \text{the sets} \]
\[ \text{and} \]
\[ \text{the transformational mappings} g_{j,r} : L_{j,r} \rightarrow E_{j,r}, g_{m,r} : L_{m,r} \rightarrow E_{m,r}. \]
\[ \text{By the assumptions of the previous section we have} R_{0r} \supseteq E_{j,r} \cup E_{m,r}. \text{ We denote} N_{0r} = E_{l,r} \cup E_{m,r}. \]
\[ \text{Figure 2 and the hyper-schema of order 1 from Figure 3.} \]
\[ \text{Definition 1.} \]
\[ N_{0l} = \emptyset \text{ if and only if} S_l \text{ is a leaf of} H. \]
\[ \text{For a symbol} h \text{ of arity 1 we consider the set:} \]
\[ M = \bigcup_{i=1}^{n_k} \{ h([x, y], a) \mid (x, a, y) \in R_{0l} \setminus N_{0l} \} \]
\[ \text{where we used the notation} h([x, y], a) \text{ instead of} h([x, y], a). \]
\[ \text{We consider the symbols} \sigma_1, \ldots, \sigma_{n_k} \text{ of arity 2 and denote by} \mathcal{HR} \text{ the Peano} (\sigma_1, \ldots, \sigma_{n_k})\text{-algebra generated by} M. \]
\[ \text{We consider the alphabet} Z \text{ including the symbols} \sigma_i, \text{ the elements of} X_i, \text{ the elements of} A_i, \text{ the left and right parentheses, the square brackets [ and ]}, \text{ the symbol} h \text{ and comma. As in the theory of formal languages, the set} Z^* \text{ defines all the words over} Z. \text{ Because a hyper-schema is a semantic schema we have the following property:} \]
\[ \text{Proposition 1.} \]
\[ \text{Proposition 2. If} S_1 \text{ is a hyper-schema of} H \text{ and} (\{x_1, \ldots, x_{k+1}\}, \theta(u, v)) \in \text{Ded}(S_1) \text{ then there is} r \text{ uniquely determined such that} \]
\[ (\{x_1, \ldots, x_{r+1}\}, u) \in \text{Ded}(S_1) \text{ and} (\{x_{r+1}, \ldots, x_{k+1}\}, v) \in \text{Ded}(S_1). \]
\[ \text{Definition 1. Let} w_1, w_2 \in Z^*. \text{ We define the following binary relation on} Z^* \text{, denoted by} \Rightarrow_H : \]
\[ \text{For} i \in \{1, \ldots, n_k\}, \text{ if} (x, e, y) \in R_{0l} \setminus N_{0l} \text{ then} w_1([x, y], e)w_2 \Rightarrow_H w_1h([x, y], e)w_2; \]
\[ \text{For} i \in \{1, \ldots, n_k\}, \text{ if} (x, e, y) \in N_{0l} \text{ then} w_1([x, y], e)w_2 \Rightarrow_H w_1dbw_2, \text{ where} d \text{ is the deductive path designated by} (x, e, y); \]
\[ \text{Suppose that} \]
\[ (\{x_1, \ldots, x_{k+1}\}, \theta(u, v)) \in \text{Ded}(S_1), i \in \{1, \ldots, n(H)\}, (\{x_1, \ldots, x_{r+1}\}, u) \in \text{Ded}(S_1) \text{ and} (\{x_{r+1}, \ldots, x_{k+1}\}, v) \in \text{Ded}(S_1) \text{ then:} \]
\[ w_1([x_1, \ldots, x_{k+1}], \theta(u, v))w_2 \Rightarrow_H w_1\sigma_i([x_1, \ldots, x_{r+1}], u, ([x_{r+1}, \ldots, x_{k+1}], v))w_2 \]
\[ \text{The reflexive and transitive closure of} \Rightarrow_H \text{ is denoted by} \Rightarrow^*_H. \text{ We denote} \]
\[ \mathcal{F}(S_1) = \{ w \in \mathcal{H}(H) \mid \exists d \in \text{Ded}(S_1) : d \Rightarrow^*_H w \} \text{ and} \]
\[ \mathcal{F}(H) = \bigcup_{i=1}^{n_k} \mathcal{F}(S_i). \]
\[ \text{Let us exemplify this computation. We consider the hyper-schemas} S_1 \text{ and} S_2 \text{ of order zero from Figure 2 and the hyper-schema of order 1 from Figure 3.} \]
\[ \text{If we take} \]
Definition 3.

Let \( g \) be an argument such that if \( x \) is in some space. In this section we describe this transformational process.

\[
\begin{array}{c}
\theta(a, b) \\
\quad x_1 \quad x_2 \quad b \\
\quad x_3 \quad x_4 \\
\end{array}
\quad \begin{array}{c}
\theta(b, c) \\
\quad x_3 \quad b \quad x_6 \quad c \\
\quad y_1 \\
\end{array}
\]

\( S_1 \quad S_2 \)

Figure 2: Semantic schemas \( S_1 \) and \( S_2 \) of order zero

\[
\begin{array}{c}
\theta(e_1, e_2) \\
\quad x_1 \quad e_1 \quad x_3 \quad e_2 \\
\quad x_5 \quad a \quad z_1 \\
\end{array}
\]

\[ \theta(\theta(e_1, e_2), a) \]

Figure 3: Hyper-schema \( S_3 \in \text{Hyp}_1([S_1, S_2]) \)

- \( L_{1,3} = \{ ([x_1, x_2, x_3], \theta(a, b)) \}, L_{2,3} = \{ ([x_3, x_6, x_5], \theta(b, c)) \} \)
- \( E_{1,3} = \{ (x_1, e_1, x_3) \}, E_{2,3} = \{ (x_3, e_2, x_5) \} \)
- \( g_{1,3}([x_1, x_2, x_3], \theta(a, b)) = (x_1, e_1, x_3), g_{2,3}([x_3, x_6, x_5], \theta(b, c)) = (x_3, e_2, x_5) \)

then we obtain the following computations:

- \( ([x_1, x_3, x_5], \theta(e_1, e_2)) \Rightarrow_H^1 \sigma_3([x_1, x_3], e_1), ([x_3, x_5], e_2) \)
- \( ([x_1, x_3], e_1) \Rightarrow_H ([x_1, x_2, x_3], \theta(a, b)) \Rightarrow_H \sigma_1([x_1, x_2], a), ([x_2, x_3], b) \Rightarrow_H^* \sigma_1(h([x_1, x_2], a), h([x_2, x_3], b)) \in \mathcal{F}(S_1) \)
- \( ([x_3, x_5], e_2) \Rightarrow_H ([x_3, x_6, x_5], \theta(b, c)) \Rightarrow_H \sigma_2([x_3, x_6], b), ([x_6, x_5], c) \Rightarrow_H^* \sigma_2(h([x_3, x_6], b), h([x_6, x_5], c)) \in \mathcal{F}(S_2) \)

In conclusion,

\( ([x_1, x_3, x_5], \theta(e_1, e_2)) \Rightarrow_H^1 \sigma_3(\sigma_1(h([x_1, x_2], a), h([x_2, x_3], b)), \sigma_2(h([x_3, x_6], b), h([x_6, x_5], c))) \)

and the last formula is an element of \( \mathcal{F}(H) \), where \( H = (Q_1, Q_2), Q_1 = [S_1, S_2] \) and \( Q_2 = [S_3] \).

4 Semantical computations in HDR systems

The semantical computation in an HDR system \( H \) transforms every formula of \( \mathcal{F}(H) \) into an object of some space. In this section we describe this transformational process.

Let us consider the HDR system \( H = (Q_1, Q_2, \ldots , Q_k) \) and an element \( w \in \mathcal{F}(H) \). If \( d = [x_1, \ldots , x_k], \theta(u, v) \in \text{Ded}(S_1) \) and \( d \Rightarrow_H^* w \) then we write \( \text{sort}(w) = \theta(u, v) \).

Definition 2. An interpretation for \( H \) is a system \( J = (\text{Ob}, \text{ob}, \text{ALG}) \):

- \( \text{Ob} \) is a set of objects;
- \( \text{ob} : X \rightarrow \text{Ob} \), where \( X = \bigcup_{i=1}^{n_k} X_i \), is a mapping that “interprets” each node as an object;
- \( \text{ALG} = \bigcup_{i=1}^{n_k} [\text{Alg}^i_{\theta(u,v)}]_{u \in A_i} \), where \( \text{Alg}^i_{\theta(u,v)} \) is an algorithm with two input arguments and one output argument such that if \( g_{i,k}([x, \ldots , y], \theta(u, v)) = (x, e, y) \) then \( \text{Alg}^i_{\theta(u,v)} = \text{Alg}^i_{\theta(u,v)} \).

Definition 3. The valuation mapping \( \text{Val}_H \) of the HDR system \( H \) is defined as follows:

- If \( w = h([x, y], a) \in \mathcal{F}(H) \) then \( \text{Val}_H(w) = \bigcup_{i=1}^{n_k} [\text{Alg}^i_{\theta(u,v)}(\text{ob}(x), \text{ob}(y))] \).
If \( w = \sigma \alpha (w_1, w_2) \in \mathcal{F}(H) \), \( w_1 \in \mathcal{F}(H) \), \( w_2 \in \mathcal{F}(H) \) and \( \text{sort}(w) = \alpha \) then

\[
\text{Val}_H(w) = \{ \text{Alg}_\alpha^1(o_1, o_2) \mid o_k \in \text{Val}_H(w_k), k = 1, 2 \}
\]

In order to exemplify the computations we consider again the HDR system \( H \) from Section 3. We define an interpretation of \( H \) by means of some *sentential forms*. Such a structure is a sentence containing two variables. If we substitute each variable by an object then a sentential form becomes a sentence in a natural language. We shall consider the following sentential forms:

- \( p_1(x, y) = \"x\) is the father of \( y\);\)
- \( p_2(x, y) = \"x\) is the mother of \( y\);\)
- \( p_3(x, y) = \"x\) is the brother of \( y\);\)
- \( p_4(x, y) = \"x\) likes to eat \( y\);\)
- \( q_1(x, y) = \"x\) is the grandmother of \( y\);\)
- \( q_2(x, y) = \"a\) brother of \( x\) likes to eat \( y\);
- \( r(x, y) = \"A\) nephew of \( x\) likes to eat \( y\);

We consider the following algorithms:

- **Algorithm Alg_\alpha^1(o_1, o_2)** { return \( p_1(o_1, o_2) \); }
- **Algorithm Alg_\alpha^2(o_1, o_2)** { return \( p_2(o_1, o_2) \); }
- **Algorithm Alg_\theta(a, b, o_1, o_2)** { return \( p_3(o_1, o_2) \); }
- **Algorithm Alg_\theta(b, c, o_1, o_2)** { return \( p_4(o_1, o_2) \); }
- **Algorithm Alg_\theta(a, e, c, o_1, o_2)** { return \( q_1(o_1, o_2) \); }
- **Algorithm Alg_\theta(b, c, e, o_1, o_2)** { return \( q_2(o_1, o_2) \); }
- **Algorithm Alg_\theta(a, e, c, o_1, o_2)** { return \( q_3(o_1, o_2) \); }

Consider the interpretation \( \mathcal{I} = (O_1, \text{OB}_1, \text{ALG}_1) \) of the system \( H \), where we specify only the useful entities allowing to exemplify the computation:

- \( \text{OB}_1 = \{ \text{Peter, Helen, John, Sorin, pizza} \} \)
- \( \text{OB}_1(x_1) = \text{Peter, OB}_1(x_2) = \text{Helen, OB}_1(x_3) = \text{John, OB}_1(x_6) = \text{Sorin, OB}_1(x_5) = \text{pizza} \)
- \( \text{ALG}_1 = \{ \text{Alg}_\alpha^1, \text{Alg}_\alpha^2, \text{Alg}_\theta^2(a, b, c), \text{Alg}_\theta^2(b, c, a), \text{Alg}_e^2, \text{Alg}_\theta^3(a, b, c), \text{Alg}_e^3, \text{Alg}_\theta^3(a, e, c) \} \)

where \( \text{Alg}_\theta^3(a, b, c) = \text{Alg}_\theta^3(b, c, a) \).

It is not difficult to observe that for the formula

\[
w = \sigma_3(\sigma_1(h(x_2, x_3, b), h(x_3, x_5, b), h(x_3, x_6, c))) = \sigma_3(\alpha, \beta)
\]

from the last part of the previous section we obtain the following computations:

- \( \text{Val}_H(\alpha) = \{ \text{Alg}_\alpha^3(o_3, o_4) \mid o_3 \in \text{Val}_H(h(x_1, x_2, a)), o_4 \in \text{Val}_H(h(x_2, x_3), b) \} \)
- \( \text{Val}_H(h(x_1, x_2, a)) = \{ \text{Alg}_\alpha^3(\text{Peter, Helen}) = \{ p_1(\text{Peter, Helen}) \} \)
- \( \text{Val}_H(h(x_2, x_3), b) = \{ \text{Alg}_\alpha^3(\text{Helen, John}), \text{Alg}_\alpha^3(\text{Helen, John}) = \{ p_2(\text{Helen, John}), p_3(\text{Helen, John}) \}
\)

therefore \( \text{Val}_H(\alpha) = \{ \text{Alg}_\alpha^3(p_1(\text{Peter, Helen}), p_2(\text{Helen, John}), \text{Alg}_\alpha^1(\text{p_1(\text{Peter, Helen}), p_3(\text{Helen, John})) = \{ q_1(\text{Peter, Helen}), q_2(\text{Helen, John}) \}
\)

- \( \text{Val}_H(\beta) = \{ \text{Alg}_\theta^3(o_5, o_6) \mid o_5 \in \text{Val}_H(h(x_3, x_6, b)), o_6 \in \text{Val}_H(h(x_6, x_5, c)) \}
\)
- \( \text{Val}_H(h(x_3, x_6, b)) = \{ \text{Alg}_\alpha^3(\text{John, Sorin}), \text{Alg}_\alpha^3(\text{John, Sorin}) = \{ p_2(\text{John, Sorin}), p_3(\text{John, Sorin}) \}
\)

- \( \text{Val}_H(h(x_6, x_5, c)) = \text{Alg}_\theta^3(\text{Sorin, pizza}) = \{ p_4(\text{Sorin, pizza}) \}
\)

therefore \( \text{Val}_H(\beta) = \{ \text{Alg}_\theta^3(p_2(\text{John, Sorin}, p_4(\text{Sorin, pizza}), \text{Alg}_\theta^3(p_3(\text{John, Sorin}), p_4(\text{Sorin, pizza}) = \{ q_2(\text{John, pizza}) \}
\)

Finally, from \( \text{Val}_H(\alpha) \) and \( \text{Val}_H(\beta) \) we deduce

\[
\text{Val}_H(w) = \{ \text{Alg}_\theta^3(e_1, e_2) \mid q_1(\text{Peter, John}), q_2(\text{John, pizza}) \} = \{ \text{A nephew of Peter likes to eat pizza} \}
\]

We observe that the conclusion obtained by \( H \) can not be obtained neither by \( S_1 \), neither by \( S_2 \). This explains why \( H \) is named a *distributed system*. 

---
We give now a short description of another interpretation \( I_2 \) for the same system \( H \). As a result we obtain geometrical images.

- \( \text{Ob}_{b_2} = \{1, (3, 3), (3, 1.5)\} \)
- \( \text{ob}_{b_2}(x_1) = 1, \text{ob}_{b_2}(x_2) = \text{ob}_{b_2}(x_6) = (3, 3), \text{ob}_{b_2}(x_3) = 1, \text{ob}_{b_2}(x_5) = (3, 1.5) \)
- \( \text{Alg}_{1}(p, q) | \text{Return the interior of the circle with radius } p \text{ and center } q \)
- \( \text{Alg}_{2}(p, q) | \text{Return the exterior of circle with radius } p \text{ and center } q \)
- \( \text{Alg}_{3}(a, b)(\alpha, \beta) | \text{If } \alpha = \text{Alg}_{1}(p, q) \text{ and } \beta = \text{Alg}_{1}(q, r) \text{ then return } \beta \setminus \alpha \)
- \( \text{Alg}_{4}(b, c)(\alpha, \beta) | \text{If } \alpha = \text{Alg}_{2}(p, q) \text{ and } \beta = \text{Alg}_{2}(q, r) \text{ then return } \beta \cap \alpha \)

For the same formula \( w \in \mathcal{F}(H) \) as in the previous computation, the object \( \text{Val}_H(w) \) given by \( J_2 \) is shown in Figure 4.

5 A Java implementation

If we note by \( \mathcal{A} \) the set consisting of some geometrical objects names then each system’s input is an word \( \nu = a_1 \ldots a_k \) over the alphabet \( V = \mathcal{A} \cup \{+,-\} \) having the following properties:

- \( a_i = +/− \) means a left/right rotation with a specific angle, denoted by \( \delta \) and to draw a line on the current direction
- \( a_i = O_j \) means to draw the graphical illustration of the object \( O_j \) such that its entry direction is on the current direction. In our implementation, each geometrical object used in the generation method is an instance of the a class named \textit{Object}. Graphically, it is a representation of a figure inside a square. Every instance of this class can have one of the following types: \textit{circle}, \textit{triangle}, \textit{star} and \textit{square} corresponding to the figure it consists of. Other members of this class are the \textit{entry direction} and the \textit{exit direction} related to some corner of the object. The corner corresponding to the entry direction becomes the \textit{entry point} of the object. Similar for the \textit{exit point}. The main routine of the Algorithm is createHDRS (Algorithm 2). The construction of the system starts by defining the schemas of the agents (steps 1 ÷ 4). The hyper-schemas of order one corresponding to the managers of the second level are constructed using the steps 7 ÷ 14. The condition for existing a hyper-schema over two schemas is that their maximal paths are connected deductive paths. This property is verified using the routine connectedPaths (Algorithm 3). If the second level of the system was successfully defined (If condition of step 15) then the process of creating new levels in HDRS continues using the While loop of step 17. The hyper-schemas of orders greater than 2 are created using the routine createHypSchs (Algorithm 4).

The geometrical objects that are used for the image generation process are introduced using the first
window of the application. For each object the user must specify the type, the entry and exit points (the corners are numbered starting from the down-left) and related to them the entry and the exit direction. Also, using the controls of the first window, the input descriptions can be edited (see Figure 5). The second window of the application gives the outputs provided by the system’s reasoning components (see Figure 6). It consists of three buttons and a panel. The application can draw maximum 1000 images with maximum 50 geometrical objects per image.

6 Conclusions and future works

In this paper we formalized the syntactical and semantical computations in an HDR system. We exemplified these computations and for some HDR system H we gave two interpretations: one interpretation generates phrases and the other generates geometrical images. This examples give an idea concerning the generative power of our mechanism. We relieved also by these examples the fact that the distributed reasoning can be modeled by an HDR system. A short description of a Java implementation for an HDR system generating images is also given. We intend to develop the applications of an HDR system. First, we intend to use the mobile agents to process such systems ([4]). Second, we intend to use the HDR systems in e-learning. The basic idea comes from the fact that a link in an HTML document gives a reference to another document of the similar structure.
(a) The initiators and some images obtained by the managers of $Q_2$ and $Q_3$ levels

(b) Images obtained at the 4th level in the system

Figure 6: Second window of the application
Algorithm 2 Procedure createHDRS

Procedure createHDRS
1. For $i \leftarrow 1$, noCmd
2. call create_schema(commands[$i$], schema[$i$], agent[$i$])
3. maximalPath[$i$] $\leftarrow$ schema[$i$].getMaximalPath()
4. EndFor
5. noAg $\leftarrow$ noComd
6. noKM $\leftarrow$ noAg + 1
7. For $i, j \leftarrow 1$, noAg; $j \neq i$
8. If connectedPaths(maximalPath[$i$], maximalPath[$j$])
9. call create_hyperSch(hypSch[noKM], schema[$i$], schema[$j$])
10. hypSch[noKM].order $\leftarrow$ 1
11. maximalPath[noKM] $\leftarrow$ hypSch[noKM].getMaximalPath()
12. noKM $\leftarrow$ noKM + 1
13. EndIf
14. EndFor
15. If noKM > noAg + 1
16. order $\leftarrow$ 2
17. While createHypSchs(order)
18. order $\leftarrow$ order + 1
19. EndWhile
20. EndIf
EndProcedure

Algorithm 3 Function connectedPaths

Function connectedPathsPath1, Path2
1. If Path1.lastNode = Path2.firstNode
2. return true
3. EndIf
4. If Path1.firstNode = Path2.lastNode
5. return true
6. EndIf
7. return false
EndFunction

Algorithm 4 Function createHypSchs

Function createHypSchs(order
1. newHypSch $\leftarrow$ false
2. For $i \leftarrow$ noKM - 1, noAg
3. If hypSch[$i$].order $\neq$ order - 1
4. continue
5. EndIf
6. For $j \leftarrow 1$, noKM - 1; $j \neq i$
7. If connectedPaths(maximalPath[$i$], maximalPath[$j$])
8. newHypSch $\leftarrow$ true
9. If $j \leq$ noAg
10. call create_hyperSch(hypSch[noKM], hypSch[$i$], schema[$j$])
11. Else
12. call create_hyperSch(hypSch[noKM], hypSch[$i$], hypSch[$j$])
13. EndIf
14. hypSch[noKM].order $\leftarrow$ order
15. maximalPath[noKM] $\leftarrow$ hypSch[noKM].getMaximalPath()
16. noKM $\leftarrow$ noKM + 1
17. EndIf
18. EndFor
19. EndFor
20. return newHypSch
EndFunction
Bibliography


