We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Agent Oriented Engineering and Methodologies with Application to Production, Economical and Social Systems

Baltazár Frankovič, Than Tung Dang, Tomáš Kasanický,
Viktor Oravec and Ivana Budinská
Institute of Informatics SAS
Slovakia

1. Introduction

Multi-agent systems paradigm and technologies are known for couple of years. They are applied in real word in many applications, where characteristic features of multi-agent systems (MAS) are useful and can help to handle, e.g. control and coordination problems. Possibilities to use MAS in many different applications are huge. This chapter features some possible alternatives, where and how MAS can be applied and a problem of agents' coalitions creation. The application field of this approach may be a variety and very wide. It is applicable in resolving planning problems in manufacturing (Sandholm, 1999), (Odubyi, 1997). Another applicable field is in creating of virtual and network enterprises (Horvath, 2006) and in marketing for enterprises, where to maximize own profit each manager (considered as one agent) can try to cooperate with another ones (Cantamessa, 2001), (Wellman, 1993). Coalition in this case may have another effect to eliminate execution of agents not belonging to the same coalition. Another problem belongs to the coalition creation for large scale multi-agent systems that adress problem of shceduling and planing in huge manufacturing systems and network enterprises. MAS can be used for modelling and managing social systems and virtual societies, as it is described in this chapter. One of new approaches to the MAS research is inspired by similarity between artificial immune systems and multi-agent systems. A multi-agent control system inspired by immune systems is described by the end of the chapter.

2. Agents and multi-agent systems approach to the modelling, management, and control of production systems.

The problem of scheduling in production process has been discussed in many recent papers. The scheduling process can be static (the schedule is constructed before execution) or dynamic (sometime, a decision has to be made immediately or for a predetermined time), depending on the system requirements. It is clear that the optimal situation in when there is enough time to search an appropriate plan so that all possible plans can be explored. In practice, mostly time is limited and so it is necessary to search for any solution for a shor time. Therefore, all requirements might not be fulfilled and, in some cases, we must content with a plan that is far from the optimal one.

The creation of coalition and coalition structure of agents is one of possible technical way for formulation agents' cooperation to improve plan quality. A coalition of agents essentially may be considered in the framework of multi-agent-system (MAS) as a group of such agents, which are willing to cooperate with all another members in this group and their common cooperative activities aim to reach the optimum of the given criteria, but on other hand, methods for cooperation are collected by these members via negotiation among themselves and they depend on concrete situations. The optimum, which the members in coalition try to reach, does not have to be always a global optimal solution, but in most cases only is optimal from any point of view (criterion) or a Nash optimal solution (Wooldrige, 1996), (Sandholm, 1999).

There exists a lot of possible ways to search for and create an optimal coalition – mainly via negotiation, or sequentially search all possible variants - and several of potential methods that we shall use for finding the solution of such formulated problem are as for example: an application of automata theory, game theory, genetic algorithms or another.

Finding a solution for the above-introduced problem has been the subject of many publications, which tried to obtain a feasible way to resolve a general problem of creating coalition (Dang, 2002), (Wooldrige, 1996), (Sandholm, 1999).

2.1 Coalition formation based on dynamic programming approach

Let $\{A_1,...,A_n\}$ be a set of n agents, and I denote a set of index, I ={1,..,n}. For simplicities we shall use a note I and a mark $K \subseteq I$ denotes a subset created by the agents, whose index belongs to a set K. Let denote $\Re_+^s$ a s-dimensional set of nonnegative real values, which is used to assess an agent's execution. Each option in this set represents one parameter of agent's execution, e.g. cost, payoff, effectiveness, etc.

Definition 1: An agent's value is a function mapping from a set of all possible results that the agent can achieve (independently or via cooperation) to $\Re_+^s$.

Further notations are: for $\forall\ i \in I, K \subseteq I$,

- q_i^* denotes a value if the agent A_i works independently,

- q_i^K denotes a value if the agent A_i joins coalition K ,

- f_i denotes an expected value of the agent A_i .

f_i = s if the agent A_i works independently, or

= q_i^K if the agent A_i joins the coalition K .

- F_K denotes a value of a set of agents $K \subseteq I$ and is defined as:

$$F_K = \sum_{i \in K} c_i * f_i .$$

where $\forall i \in K$, c_i is a parameter, which is used to express the priority of the agent A_i in this society (since some agents might be more important than other ones depending on concrete applications, therefore these parameters might be various) and $\sum_{i=1}^{k} c_i = const$. Generally, all the agents have the same priority; therefore, the function F_K can be defined as:

$$F_K = \sum_{i \in K} f_i \tag{1}$$

In this case $\forall i \in [1, n]$, $c_i = 1$.

In practice, each agent can know what it will get after joining any coalition until the discussion with other ones (about what every member within this coalition has to do regarding their possibilities, aims and willingness, and consequently which results each member can achieve) within this coalition finishes. As a result, in order to know all the values q_i^K the agents need another phase to negotiate with other ones about each possible coalition. It is clear that this process requires very exhausted work. To simplify let assume that these values are defined in advance or taken from historic database. Certainly, the more these values are defined, the larger is a probability to achieve the optimal solution. If the agents' society is too large, and that results an agent cannot explore every possible coalition, in such case all the values $q_i^K |_{i \in K}$, where the coalition K has not been explored, are defined to be zero. In the next part we will show some important definitions and characteristics of the agents' coalition.

General definitions and properties of agents' coalitions

Initially, let consider a following important property of the optimal coalition: for each set $K \subseteq I$ is valid that the set K is an optimal coalition if every experiment trying to separate the set K to a set of smaller coalitions will decrease a value F_K (defined by an equation (1)).

This property is derived from the assumption, in which the agents are always willing to remain in the large coalition, if they know that breaking this one to smaller sets will bring the whole party worse results.

Definition 2: A set $K \subseteq I$ is an optimal coalition if

$$\sum_{i \in K} f_i \geq \sum_{K_j} \sum_{i \in K_j} f_i, \tag{2}$$

where $K = \bigcup_j K_j$.

Every agent, in fact, is an optimal coalition itself, because it is impossible to divide it to smaller subsets. From definition 1 the following theorem is deduced.

Theorem 1: Let be any structure of a set I for which the value F_I achieves maximum, then, such structure must be only composed from the optimal coalitions.

Proof: Let a set I be divided to *m* subsets like coalitions.

$$I = \bigcup_{j=1}^{m} K_j, \text{ and } \forall i \neq j \in [1, m] \text{ is valid } K_i \cap K_j = \{\emptyset\}.$$

If any subset $K_{i \in [1,m]}$ is not an optimal coalition, then, resulting from the definition 1: here must exist such a structure for organization a set K_i, $K_i = \bigcup_r K_r^i$, in which a value F_{K_i} is bigger than an actual value. Afterwards a structure composed as $s = \{\bigcup_{j=1, j \neq i}^{m} K_j\} \bigcup_r K_r^i$ will have a bigger value than the actual one. Confrontation.

Definition 3: Let S_i denote a set of coalitions, which can bring the agent A_i at least the same or better results than when it works independently.

$$S_i = \{K \subseteq I \mid q_i^K \geq q_i^*\} \tag{3}$$

Definition 4: Let S_K denote a set of coalitions, which can bring at least the same or better results for all agents belonging to a set K than when they work independently.

$$S_K = \{Ko \subseteq I \mid \forall i \in K, \ q_i^{Ko} \geq q_i^* \} \tag{4}$$

Example 2: In the previous example, if a set $K = \{1,2\}$, then, $S_{\{1,2\}} = \{1,2,3\}$, etc.
From these definitions the following results are derived:
Lemma 1:
1.   If $K_1 \subset K_2$, then, $S_{K_1} \supset S_{K_2}$.
2. $S_{K_1 \cup K_2} \subset S_{K_1} \cap S_{K_2}$.
3. $S_{K_1} \cap S_{K_2} \subset S_{K_1 \cap K_2}$.
Proof: see in (Dang, 2002)
Consequences 1:
1. $S_{K_1 \cup K_2} \subset S_{K_1 \cap K_2}$
2. $S_{K_1 \cup .. \cup K_m} \subset S_{K_1} \cap ... \cap S_{K_m}$,

where $K_1, ..., K_m \subseteq I$.

Certainly, we want to attempt to implement all possible features of the real agent's behavior into our model, however, a mathematical model cannot copy exactly and completely agent's behavior, but one property has to be considered and it is also very often happened in practice. It is that, every agent is willing to join coalition with other ones if and only if this coalition brings it at least the same or better results than when it works independently. Furthermore, a coalition might exist if all the agents creating this one are willing to join.
Assumption 1: the agent A_i joins coalition $K \Leftrightarrow (q_i^K \geq q_i^*)$.
Assumption 2: a set $K \subseteq I$ is an *acceptable coalition* $\Leftrightarrow \forall i \in K : (q_i^K \geq q_i^*)$
With these assumptions it is possible to deduce a following theorem.
Theorem 2: A set $K \subseteq I$ is an *acceptable coalition* (does not have to be optimal) if and only if

$$K \in \bigcap_{i \in K} S_i \tag{5}$$

Proof:
1.   If K is an *acceptable coalition*, it means $\forall i \in K$ it is valid: $q_i^K \geq q_i^*$. Afterwards, $K \in S_i$ for $\forall i \in K$. A result from this is $K \in \{\bigcap_{i \in K} S_i\}$.

2.   If $K \in \bigcap_{i \in K} S_i$, it means that the coalition K can bring at least the same or better results for all the agents belonging to this coalition $\Rightarrow$ a set K can be an *acceptable coalition* (does not have to be optimal).

Consequence 2: If $\bigcap_{\alpha_i \in I, i=1,..,m} S_{\alpha_i} = \{\varnothing\}$, then these agents $A_{\alpha_1}, ..., A_{\alpha_m}$ cannot be in the same acceptable coalition.
The proof is derived directly from the theorem 2.
Example 3: In the previous example, since $S_1 \cap S_2 \cap S_3 = \{1,2,3\}$, therefore, the set $\{1,2,3\}$ is a unique possible coalition that all the agents are willing to join.

The main problem addressed here is: to find such a structure of a set I to maximize a function F_I defined by (1). Another way to say: we want to find such a structure $I = \bigcup_{i=1}^{m} K_i$, where $\forall$ i, j $\in [1,m]$ $K_i \cap K_j = \{\varnothing\}$ and $F_I = \sum_{i=1}^{n} f_i = \max$.

To resolve completely this task and to find a maximal value of a function F_I it is necessary to search the whole space of all possible structures of a set I. The complex searching of all possible structures of a set I can be executed in a variety ways, but a complexity of those approaches is the same for every method, because every variant have to be examined and for arbitrary n the problem is known to be NP-hard, thus, it is necessary to turn to heuristic methods to search, which are computationally efficient but which might guarantee only sub-optimal solutions from any point of view, for example: *tabu search, branch and bound, genetic algorithms*, etc.. Generally, a quality of those methods strongly depends on how long the process has repeated and a quality of an initial choice. On the other hand, each agent presents one independent object with own goals and intelligence, and consequently the agent's execution (or choice) will aim to the best results for the agent; therefore, naturally each agent might poll such coalitions, in which its benefit is the best from its point of view.

Furthermore, next property of such formulated problem that is easy to verify is that the mentioned problem might be resolved by a *recursive approach*, sequentially for cases with 1 to n agents, where each new agent will try to join any formulated coalition, afterwards a result of the criterion function F_I will be compared with the best one so far. This approach can guarantee the complex searching of the whole space and the final solution certainly will be the globally optimal one. But, as we have mentioned above, agents' choices might influence the searching process a lot, because on a basis of these choices lots of coalitions might not be necessary to be examined and searching space could be reduced.

A general dynamic programming scheme

For a set $K \subseteq I$ it is assumed: a set K can be decomposed to an amount of independent nonempty subsets. $K = \bigcup_{j=1,..,m} K_j$, where, $\forall$ i, j $\in [1,m]$, $K_i \cap K_j = \{\varnothing\}$. Certainly $m \leq |K|$.

Let us define a function Q_K as follows:

$$Q_K = \arg \max_{\{K_j\}} \{ \sum_{K_j} \sum_{i \in K_j} f_i \} \tag{6}$$

Or by other words, Q_K is a maximal sum of all agents' expected values, which are creating the set K, among all possible structures of the set K.

Furthermore, by comparison an equation (1) and (6) it is possible to derive that $Q_K = \max\{F_K\}$ for $\forall$ $K \subseteq I$. Therefore, the main task mentioned in the previous section might be transformed to the task to find a value Q_I and such a structure, which achieves this value.

Because each agent has only two alternatives: to work alone or to join any coalition. From this reason we shall present a general dynamic programming method for resolving problem Q_I. We consider n stages, where in each stage one agent is added and after n-th stage the whole problem will be resolved. For $j=1,..,n$-1 we denote $I_j = \{1,..,j\}$, where every set I_j means we will search an optimal solution (an optimal structure) for j agents $\{A_1,..., A_j\}$, which guarantees a maximal value $F_{I_j} (= Q_{I_j})$. Since each set I_j can be obtained by adding the agent A_j to the set I_{j-1}, we might have the following recursive equation:

$$Q_{I_j} = \arg\max_K \{(Q_{I_{j-1}} + q_j^*), (Q_{I_{j-1}\setminus K} + Q_{K\cup j})\} \tag{7}$$

Where $K \subseteq I_{j-1}$ and $j=\{1,..,n\text{-}1\}$, $I_0 = \{\varnothing\}$.

Further, for each $j=\{1,..,n\text{-}1\}$, if we denote

$$h_j = \arg\max_{K\subseteq I_{j-1}} \{Q_{I_{j-1}\setminus K} + Q_{K\cup j}\} \tag{8}$$

The equation (7) could be rewritten as:

$$Q_{I_j} = \max\{(Q_{I_{j-1}} + q_j^*), h_j\} \tag{9}$$

On the basis of these recursive equations we can, in principle, find an optimal solution for Q_I by the following backward scheme:

- Sub-problem in stage n: Search a value h_n in (8) by examining all possible coalitions that the agent A_n is willing to join with using theorem 2 and consequence 2. After that, replace them to (9) to obtain a value Q_I.

- Sub-problem in stage $n\text{-}1$: Analogously as above: it is necessary to examine those coalitions that the agent A_{n-1} is willing to join and do not involve the agent A_n, etc.

- Sub-problem in stage 1: $Q_{I_1} = q_1^*$.

Now we get:

Theorem 3: By solving sub-problem in n stages we will obtain the optimal structure for the given set I and a maximal value of F_I.

Proof: (outline) Since by adding an agent A_n to a set I_{n-1} we obtain the set I, and from these equations (7), (8) it is easy to verify that calculating of a value Q_I can be made from the known values in the previous stages from the first to (n-1), otherwise, these values have been calculated for optimal cases, in which a structure of each subset was optimal. Consequently, a value Q_I from an equation (9) has to be optimal one.

From an equation (8) it is possible to contend that the main complication in the general method presented above is that the process of calculating a function $h_{j\in[1,..,n-1]}$ is huge and complicated, essentially with large values of n. Because, theoretically, the new added agent is able to join every coalition (if there are $j_{j=1,..,n-1}$ agents in the previous set, then, the new $j+1$ agent could join one of 2^j possible coalitions), therefore dimensions of this problem will grow with an exponential speed when n is increased.

To improve the previous approach to computable dimensions, *the first step* that we need to do is to reduce an amount of coalitions that the new added agent can join. To do it, it is necessary to use agents' intelligence and their attributes. Because each agent can own different mechanisms to poll coalitions that it is willing to join. *The second step* is: applying theorem 2 and consequent 2 to remove coalitions that the new added agent cannot join and leave only the *acceptable coalitions,* which are accepted by all the agents creating these ones. Afterward, an equation (8) could be resolvable. Certainly, the results of the first step will hardly depend on each agent's choices even though, these choices might be executed by different ways and they also might be dependent on other agents' decisions too. In the next section we will discuss this problem.

Creating the optimal structure by sequential conceding.

As we have mentioned above, reducing an amount of potential coalitions that the new added agent could join is a necessary thing, which enables to overcome huge complexities of

an equation (8). We assume that, these expected values ($q_{i \in I}^{K \subseteq I}$), which each agent is able to achieve by joining coalition with another ones, are random and known for all the agents creating this coalition. Moreover, any continue function does not exist, which is able to approximate exactly these values (because if it exists, this problem could be solved by different ways as for example: searching global maximum or minimum of any multi-parameters' function). Furthermore, for simplicities we will omit a negotiation part to achieve these values $q_{i \in I}^{K \subseteq I}$ and assume that they are known.

The first step mentioned above: to reduce an amount of coalitions that the new added agent could join, might be made by the following principle:

Because for each agent, a coalition, in which its expected value is maximal, it will have the highest priority in its choice. On other hand, with such choice of each agent, they will with a very high probability never reach an optimal structure. Therefore, each agent will have to concede, however, a new question: which agent has to concede and how much is a further complicated problem, because the agent could prefer one coalition before another from its criteria or on the basis of its deliberations it might consider its choices as the best for its execution. We shall propose three methods, in which the agents are assumed to have the same aim to achieve the global optimal executions of the whole agents' society.

Before that, we have to define some important specifications:

Definition 5: for each agent $i \in I$, let $v_i^0 = \max_{K \subseteq I}\{q_i^K\}$ be a maximal expected value that the agent A_i can receive and for $\forall 0 < \alpha \leq v_i^0$ let $\Omega_\alpha^i = \{K \subseteq I \mid q_i^K \geq \alpha\}$ be a set of coalitions in which the expected value for the agent A_i is more or equal than α.

After these preparations we can now formulate our algorithms:

First, each agent states its aim equal a maximal value that it can achieve in all coalitions and collects only these coalitions in which its expected value is more or equal this value. After that, the agents try together to find a solution among these sets, if they cannot find any solution, all the agents sequentially have to concede by decreasing its aim. In an algorithm 1, all the agents sequentially decrease their aim but with the same ratio (ratio between a new aim and present one), in an algorithm 2, a difference between a new aim and the present one is the same for every agent. In an algorithm 3, we consider that in each turn only one agent decreases its aim and it always collects such coalitions, in which its expected value is maximal between remained ones.

An algorithm 1: (based on an *equal ratio* of conceding)

Step 1:

a. $\forall\ i \in I$ Choose coefficient $c=1$ and $\alpha_i = c * v_i^0$.

b. $\forall\ i \in I$ Search a set $\Omega_{\alpha_i}^i$.

Step 2:

a. Using a general dynamic programming scheme presented above to search a sub-optimal solution among sets $\Omega_{\alpha_i}^i$, $i \in I$.

b. If a solution is not found, decrease a coefficient c and repeat a step 1.

c. If any solution is found, this process is stopped.

To effectively resolve a step 1.b, at first each agent sorts out all possible coalitions accordingly a value q_i^K, $K \subseteq I$ (since each agent could join 2^{n-1} possible coalitions, then, this procedure has a complexity $\cong O(2^{n-1} * (n-1) * \log 2)$). Afterwards, in every time when

the coefficient c is decreased, each agent can choose immediately from its sorted coalitions feasible ones, which satisfy a condition for creating a set Ω^i_α, $\forall\ i \in I$. To solve a step 2.a we could apply the approach based on a dynamic programming scheme presented above.

Similarly, we shall propose further algorithm based on equal regressions of each agent in a searching process.

An algorithm 2: (based on an equal regression of conceding)

Step 1:

a. $\forall\ i \in I$ Choose coefficient c=0 and $\alpha_i = v^0_i - c$.

b. $\forall\ i \in I$ Search a set $\Omega^i_{\alpha_i}$.

Step 2:

a.   Search a sub-optimal solution among sets $\Omega^i_{\alpha_i}$, $i \in I$.

b.   If a solution is not found, increase a coefficient c and repeat a step 1.

c.   If any solution is found, this process is stopped.

Differences between these both algorithms are not very significant; the results obtained by applying them depend on values q^K_i, for example: when values q^K_i are significantly different (a value $\text{var}(q^K_i)_{K \subseteq I}$ is high) for every $i \in I$, then, the first algorithm could achieve better results than the second one, but in another case when these values are nearly similar, the results could be contrary. To improve these algorithms, we shall present the third one, which is modified from the previous methods and differs from them in such point that in each turn only one agent tries to decrease its aim and how much this agent decreases, it depends on its decision. Certainly, if each agent chooses a different way to decide how much it wants to concede, then, this process might be very interesting and immediately occur lots of problems related to agent's decision, negotiation, selecting agent to concede, etc.

In the next method, each agent sequentially will modify its aim in turn and we assume that each will choose immediately the best coalitions from a set of the remained ones. For the next algorithm we define some notations:

Let $\Psi_i = \{\phi^j_i\}^{\lambda_i}_{j=1}$ be a sorted set of values q^K_i for the agent A_i (from top to down). λ_i is a sum of possible coalitions that the agent A_i could join - $\max(\lambda_i) = 2^{n-1}$. Certainly it is valid: $\forall i \in I$, $\phi^1_i = v^0_i$. Instead of coefficient c we will use only α_i for simplicity.

After preparations we can present the following algorithm:

An algorithm 3: (modified from *the algorithm 1 and 2*)

Step 1:

a. $\forall\ i \in I$ Choose coefficient $\alpha_i = \phi^1_i = v^0_i$.

b. $\forall\ i \in I$ Search a set $\Omega^i_{\alpha_i}$.

Step 2:

   $i = 1$.

a.   Search a sub-optimal solution among sets $\Omega^j_{\alpha_j}$, for all $j \in I$.

b.   If a solution is not found,

    1.   decrease a coefficient α_i to: $\alpha_i = \max\{\phi^k_i \mid \phi^k_i < \alpha_i\}$,

    2.   update a set $\Omega^i_{\alpha_i}$,

3. repeat a step 2.a,

4. if $i < n$, then $i = i + 1$, else $i = 1$.

c. If any solution is found, this process is stopped.

Comparison between these algorithms is implemented in an illustrate example with 11 agents and shown in following graphs and a number of steps of the searching process to achieve the final solution is depicted in a table 1.

In these experiments we have bounded that each value q_i^K lies in an interval [0,100] and is randomly generated. From the obtained results it is possible to conclude that the algorithm presented above have achieved average very good results (average 95 percent of the optimal results, in some cases they even achieved the optimal results) and for a short time (see table 2). The difference between these algorithms is not so significant, in some examples one algorithm could achieve better results than another ones, but both algorithms 1 and 2 are more appropriate for parallel working, since after each round each agent could immediately calculate its value α_i and does not have to wait until its turn. On the other hand, for an arbitrary set of values $\{q_i^K\}$, the algorithm 3 generally produced less complicated sets $\Omega_{\alpha_i}^i \mid_{i=1,..,n}$ than both first ones, since after each turn only a small number of possible coalitions is added, not massively as in the algorithms 1 and 2. Moreover, the algorithm 3 could be extended and modified for the case when each agent should choose a different way to calculate its value α_i. In a graph (Fig. 1), the results of complex searching always have the maximal values. Another problem that may appear in this graph is that in several points is only one pattern; it means more results have the same value in this point.

In the Table 1 is shown how many steps the searching program has repeated until the final result is found. These values may be used also to point to a time-complexity of each method to achieve the final results. For a genetic algorithm we fixed that the program will finish after certain number of steps.

Order	Complex searching	GA	A1	A2	A3
1	859868	8500	134	173	238
2	900848	8500	112	113	189
3	1025352	8500	359	190	522
4	839476	8500	76	47	135
5	864455	8500	144	125	228
6	904389	8000	86	77	154
7	926803	8000	57	36	89
8	865845	8000	12	24	20
9	886832	8000	164	103	287
10	1112554	8000	152	168	260

Table 1. A case with 11 agents - complexity for searching max(F_I)

The experiment has been repeated with more agents and from the obtained results we can see that: with a small number of agents (up to 8 agents) the complex searching approach can be applied, because a complexity is acceptable and all another reducing algorithms give sufficiently large differences. For cases with more agents (9 agents and more), all reducing methods have an omissible low complexity in comparison with using the complex searching

approach or a genetic algorithm; nevertheless the results of these algorithms achieve just about the best results obtained by searching complex space. Better results we can reach by modifying the step 2.b.1 in the algorithm 3, instead of adding only coalitions, which have a value φ_i^k or q_i^K respectively immediately after a coefficient α_i , we can attach more ones at once. By this way a number of satisfied solutions will increase and a chance to reach the best solution will be bigger. Otherwise, for a case with 11 agents, a number of steps that the searching program repeated are about 1 million and this number will continuously increase with an exponential speed, therefore, from a practical point of view it is not possible to apply the complex searching method but more effectively is to use one of some presented reducing mechanisms, which give also average optimal results.
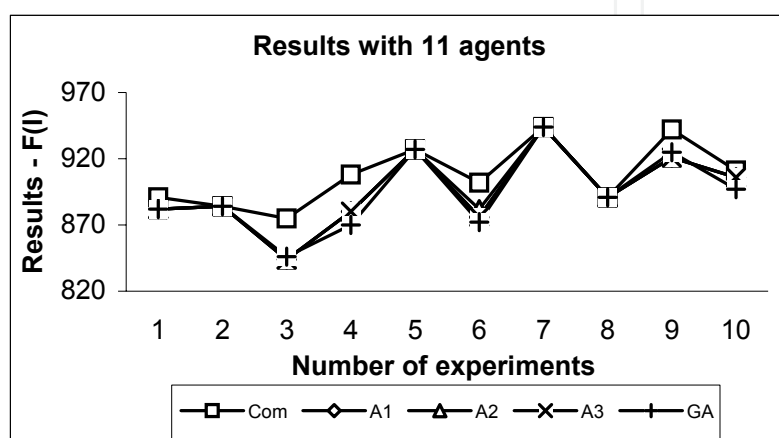

Fig. 1. Results of searching max(F_I) with 11 agents

Notation: in figure 1 and table 1: *com* = complex searching, *GA* = genetic algorithm, *A1* = algorithm 1, *A2* = algorithm 2, *A3* = algorithm 3.

2.2 Coalition formation for large scale multi/agent systems
Recently huge manufacturing systems and network enterprises have been created. There is lack of new theoretical and practical tools for solving upcoming problems such as enterprise integrity, distributed and heterogeneous environment. According to (Nof et al., 2005), previously mentioned problems in huge manufacturing systems can be handled by large-scale multi-agent systems (LSMAS). LSMAS are distributed systems with large number of elements. These systems can easier handle problems such as robustness, flexibility and scalability than centralized systems. Computations and methods used in systems can be distributed among independent parts of the system. These independent parts solve problems with smaller complexity and results are aggregated together. In MAS theory several distributive methods can be found, e.g. agreement reaching methods such as auction, bargaining and coalition formation (Sandholm, 1999), (Woodridge, 2002). Almost every previously mentioned approach has two type methods. The first type is known computational method, which uses more computations than negotiation. The second type is negotiation method, which prefers negotiation before computation.
Another very important topic in a new age manufacturing system is to utilize human knowledge and experience. There are numerous approaches how to store knowledge, for supporting users in manufacturing and production systems as well. Among recently most used approach belongs ontology utilization (Budinská, 2004), (Dang, 2003) and experience

management (Bergman, 2002). However, expressions such as near, slightly above or almost cannot be described by numbers and thus it is hard to store them. Fuzzy relations can handle this problem easily and complex. Users' inputs are fuzzy membership functions rather than single values.

Objective function

Huge amount of decisions are made on the basis of some criterions called objective functions. A LSMAS that fulfils enterprise integrity property is divided into dependent or independent subsystems. Decisions made in these subsystems may have different domains. And thus, each decision process should have different objective function. Principle of objective functions should be based on measuring distance between two compared entities, which can be represented by attribute-value pairs, an object or a tree structure and etc (Bergman, 2002). One of these entities is know as a set point and represents desired properties of right decision. An objective function can be Euclidian or hamming distance measure, fuzzy similarity measure and etc. Entities represented by vectors are the most used, hence lots of specialized metrics has been invented to solve this issue. Note that methods that utilize the vector representation of entities do not take into account their structure. For example, in coalition formation problem coalition properties are compared, but the structure of coalition is not considered. If the structure of compared entities is considered, then more sophisticated methods should be used, e.g. social hierarchical entropy (Balch, 2000). These metrics have very crucial assumption; vectors, which represent entities, must have the same dimensions, otherwise vectors could not be compared. This is very important fact, because in LSMAS should occur situations, where two vectors with different dimensions have to be compared. In such situations, standard distance measures, such as Manhattan metric, can be modified by dimension evaluation .

$$f(\overline{x}, \overline{y}) = \frac{1 - \alpha n}{m_y} \cdot \frac{1}{n} \sum_i |x_i - y_i|$$

$$\forall i : x_i \notin \{\} \wedge y_i \notin \{\}$$

(10)

where $\overline{x} \in X$ is a vector of attribute-values pairs with dimensions $(m_x, 1) : m_x \in N$ that describes properties of an entity that is compared with a set point entity represented by a vector $\overline{y} \in X$ with dimensions $(m_y, 1) : m_y \in N$, X is the space of all possible vector representations of entities, $n \in N$ is a number of attribute-value pairs, which can be found in both vectors $\overline{x}$ and $\overline{y}$, $\alpha \in (0,1>$ is a coefficient that shorten distances that have bigger n . This heterogeneous modification is based on preferring distances, where two entities with similar sets of attributes are compared. Distances with normalized input vectors are usually defined as functions $f : X \times X \rightarrow <0,1>$, where the shortest distance have the lowest value of distance measure f . The most of distance measures fulfil symmetry, reflexivity and triangular inequality properties; previously mentioned heterogeneous modification of Manhattan metric is one of such measures.

Fuzzy voting criterion

Fuzzy voting criterion (FVC), which is depicted in simple scheme on Fig. 2, is based on fuzzy pre-processing of attribute values and on agents voting.

Fuzzy preprocessing includes a set of fuzzy set, which describes the set point entity and transforms attribute values of compared entities into interval $\langle 0,1 \rangle$. Each fuzzy set is defined by its membership function $\mu(x_{ji})$, shown in Fig. 3 where x_{ji} is the value of the j-

th property of i-th analyzed attribute vector. In this example, fuzzy transformation converts the value of one analyzed property, i.e. single variable transformation is considered. More complicated fuzzy transformations can be defined as well.
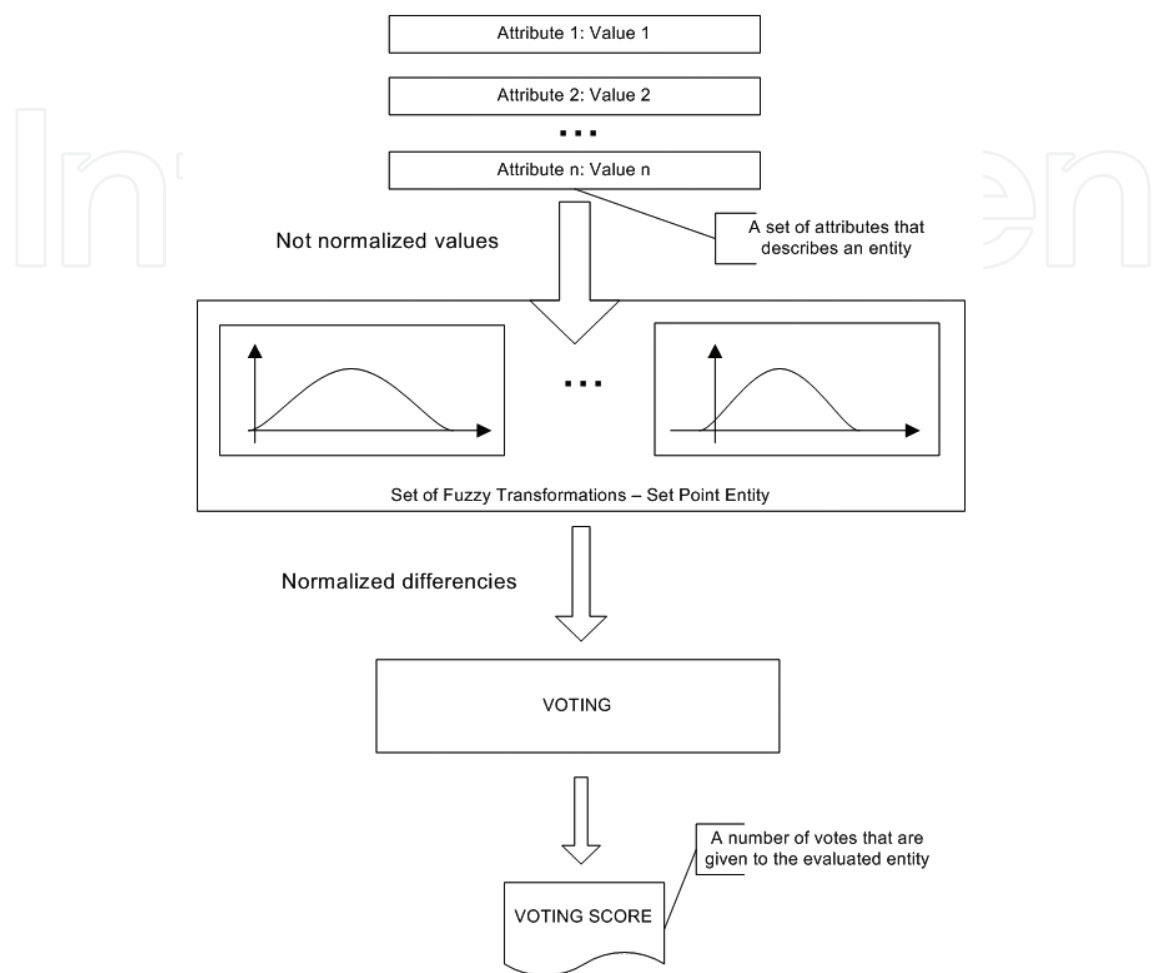

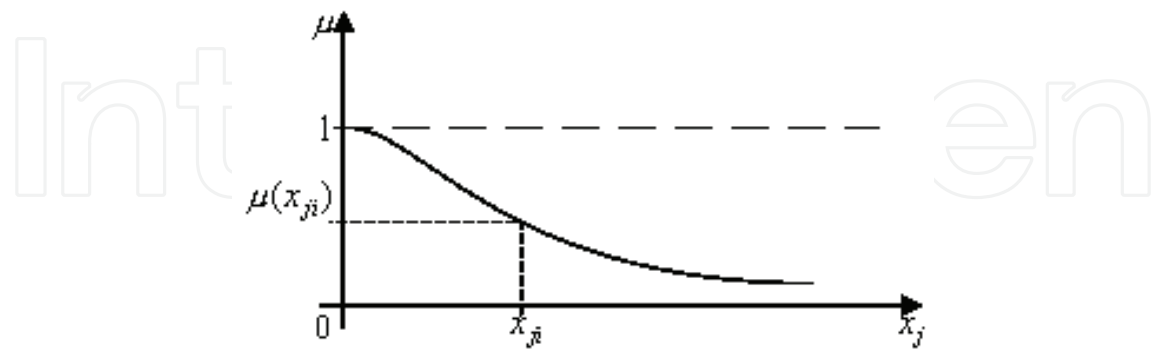Fig. 2. Illustrative scheme of fuzzy voting criterion.

Fig. 3. An example of fuzzy transformation of *j*-th coalition property.

Fuzzy transformation results are processed by voters that make votes in election. Voters assign votes to each transformed value. Assigning of votes is a simple straightforward algorithm and can be summarized as follows:
1. Each vector that includes analyzed properties is assigned to voter that supports these properties.

2. Each voter evaluates the membership function value of the defined fuzzy transformation for each vector. Note that voters transform only subset of values of the vector.

3. Each voter sorts the set of transformed values in descending order.

4. Each voter assigns particular amount of votes to a vector. The assigned amount of votes is based on the following rule: The first vector gets votes, where is the number of vectors assigned to the w-th voter. In general, can differ from the number of all vectors, because only vectors, whose properties are supported by the w-th voter, are assigned to that voter. The last vector gets one vote. If two or more vectors have the same values, then all of them get votes, where is the amount of votes assigned to the first vector that has higher transformed value.

5. Votes for each vector are summed across voters, i.e. final amount of votes for each vector is the sum of votes from all voters.

6. The best vector is that with the highest number of votes.

The sixth step declares that the best vector maximizes utility.

Mathematical representation of the fuzzy voting criterion

Definition of environment:

- Vector of coalitions is denoted by $C = (c_1, c_2, ..., c_l)^T$, where $l \in L$ is the number of coalitions that are compared in the coalition formation algorithm.

- Vector of voters is denoted by $V = (v_1, v_2, ..., v_w, ..., v_o)^T$.

- Vector of coalition properties is denoted by $X = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_l)^T$.

Definition of fuzzy and voting transformation.

Function $f_i : X \to Z_i$ is called fuzzy transformation function for the i-th voter, where vector $Z_i = (z_{i1}, z_{i2}, ..., z_{il})^T, z_{ij} \in \langle 0,1 \rangle$. Function $g_i : Z_i \to W_i$ is called voting transformation function for the i-th voter, where $W_i = (w_{i1}, w_{i2}, .., w_{il})^T, w_{ij} \in W$. Function $\overline{f} = (f_1 f_2 ... f_o)^T$ is called voting system fuzzy transformation function. Function $\overline{g} = (g_1 g_2 ... g_o)^T$ is called voting system voting transformation function.

Definition of fuzzy voting criterion.

Vector of fuzzy transformation criterion value for each coalition is $\overline{J} = (\overline{g}(\overline{f}(Y)) \cdot \overline{e})$, where $\overline{e}$ is vector $(1,1,...,1)^T$ of size $(l,1)$. $\overline{J}_i$ is value of fuzzy voting criterion for the i-th coalition.

Vector of fuzzy transformation criterion can be rewritten in a more complex way:

$$\overline{J} = (\overline{g}(\overline{f}(Y)) \cdot \overline{e}) = $$
$$= g_1(f_1(Y)) + ... + g_l(f_l(Y)) \tag{11}$$

Note that the above definition looks like linear criterion. In some applications, voters may have different influence on results of the criterion. The definition of FVC can be modified as follows

$$\overline{J} = (\overline{g}(\overline{f}(Y)) \cdot \overline{a}), \tag{12}$$

where $\overline{a} \in R^n$ is voters' influence vector. Note that $\overline{a}$ has the same role in FVC as diagonal items in matrix A of the quadratic criterion.

Comparison of fuzzy voting and heterogeneous Manhattan metric

An advantage of FVC is that decisions are made in distributed manner and no preprocessing of properties values is needed. Normalization of input data is performed by the fuzzy transformation.
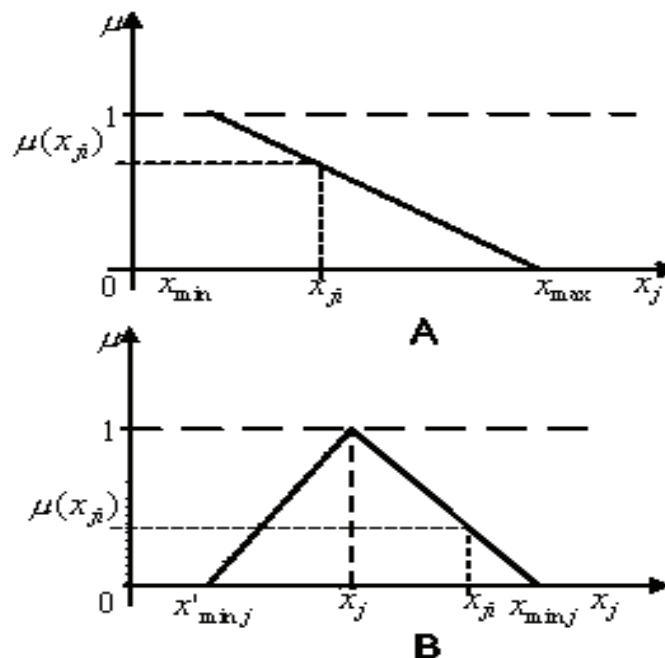

Fig. 4. Examples of normalization performed by fuzzy transformation

In Fig 4 examples of such normalization are given. Note that each entity property is transformed into the interval $\langle 0,1 \rangle$; thus all properties are normalized. In these examples, the normalization is linear (Figure 4A). The linear normalization depends on minimal x_{min} and maximal x_{max} values of the set of considered values. In general however, normalization can be influenced by experts' intentions, i.e. some values of the transformed interval can be preferred.

In Figure 4 B, special type of fuzzy transformation is shown. The maximal normalized value of the j-th analyzed property is $\tilde{x}_j$. This value represents the desired value of the j-th analyzed property. This type of transformation has two minimal values – $x'_{min\,j}$ and $x_{min\,j}$.

Figure 4 proposes that in general entities have two types of properties, the first (Figure 4A) being a type of properties that is minimized during a evaluation, and the second (Figure 4B) minimizes the difference between the value of property and the desired value of this property $\tilde{x}_j$. In case of heterogeneous Manhattan metric differences have to be evaluated; in case of FVC these differences are evaluated by fuzzy transformation function.

An illustrative example:

FVC is used in a polynomial coalition formation algorithm in a super-additive environment, based on an enhanced distributed generation of a coalition configuration proposed by Shehory and Kraus (1999), where the coalition configuration is a set of coalitions of agents in MAS and This generation algorithm has the following advantages:

- the set of coalitions are polynomial
- the load of computation is distributed among agents

- agents generates only unique coalitions
- the search space (coalition configuration) of the algorithm can be controlled by probability distribution of random choices made by agents.

A multi agent system consists of one hundred agents, which generates 5098 unique coalitions in the coalition configuration. Each agent represents one power plant in an electric grid that is described by three properties, namely cost per time unit (c), resource (r), loading time (lt) and loading cost (lc). Note that engineer units are considered in all measurements. Several agents with properties are presented in the table 2.

power plant	r	lt	c	lc
A31	21	9	1	0.25
A32	10		0.125	
A33	24	2	1.25	1.7
A34	36	2	1	0.57
A35	10	6	1.7	1.7
A36	30		0.8	3.5
A37	20	1	0.5	0.5
A38	8	3	0.5	
A39		5	0.75	
A40	19	5	1	4

Table 2. Values of several analyzed agents' properties. Note that some values are not filled in tables, i.e. attribute values of particular agents are not known or not defined. This is typical situation in LSMAS, which result in heterogeneous environment.

Experiment in a heterogeneous environment

In this subsection comparison of FVC and heterogeneous Manhattan metric is performed, which is made by the experiment based on searching for the best coalition that minimizes the cost per time unit the cost per preparation time, preparation time and to minimize the difference between the value of the coalition's resource property and the desired value of the resource property stored in set point. The force of each property is equal, i.e. each property has the same influence on the criterion value (objective function).

The heterogeneous Manhattan metric is defined as in (Balch, 2000). Definition of the FVC is more complicated. FVC consists of four voters, where each voter is assigned to one property. Each voter has a fuzzy transformation. Each fuzzy transformation is defined as follows

$$f_i(x_{ij}, c_i, \sigma_i) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \tag{13}$$

where $x_{ij} \in \bar{x}_j$ is the *i*-th value of the *j*-th property which is assigned to the *i*-th voter, c_i is the desired value of the *i*-th property and finally σ_i is *i*-th coefficient that describes width of Gaussian curve. In this experiment the values of c_i and σ_i for each voter are defined in the following table 3.

Voter	i	c_i	σ_i
r voter	1	30	10
lt voter	2	0	10
c voter	3	0	10
Lc voter	4	0	10

Table 3. Parameters of voters' fuzzy transformations

Note that the value of the parameter c_i is the desired value of a particular property.

In the following two tables 4 and 5, results of the experiment are presented. In the first table, the results of top ten coalitions evaluated by the FVC are given. In the second one, there are results evaluated by the heterogeneous Manhattan metric.

Position	lt	lc	r	c	FVC
1	1,00	0,50	20,00	0,50	20356
2	1,92	0,50	33,00	0,50	20348
3	1,33	0,50	38,00	1,00	20325
4	3,00	1,00	24,00	0,67	20313
5	3,00	1,00	45,00	0,67	20313
6	6,00	0,88	39,00	0,10	20311
7	8,00	0,44	14,00	0,17	20303
8	6,00	0,75	36,00	0,67	20301
9	6,75	1,00	20,00	0,50	20292
10	8,00	0,44	47,00	0,79	20280

Table 4. Results of fuzzy voting criterion in heterogeneous environment

Pos.	lt	lc	r	c	HMM
1	1,00	0,50	20,00	0,50	0,001834
2	1,92	0,50	33,00	0,50	0,002188
3	1,33	0,50	38,00	1,00	0,002718
4	0,33		2	0,8	0,002744
5	3,00	1,00	45,00	0,67	0,003551
6	6,00	0,88	39,00	0,10	0,003681
7	8,00	0,44	14,00	0,17	0,003958
8	6,00	0,75	36,00	0,67	0,004363
9	6,75	1,00	20,00	0,50	0,004748
10	8,00	0,44	47,00	0,79	0,004901

Table 5. Results of heterogeneous Manhattan metric in heterogeneous environment

Note that in the tables, only top ten coalitions are given; the remaining coalitions are omitted. The top ten coalitions are very similar in both cases. In general the best coalition can differ in each criterion, because the best coalition is suboptimal, not optimal. Better

results might be encountered in case of weigthened modification is considered. However, the proof of this subsection is obvious: the FVC is applicable criterion for coalition formation problem.

In the case of multivariable fuzzy transformation, a voter works with more than one property of a coalition. This can be very useful in the case when some subset of properties are dependent or are meaningful as whole only. In situations where several values of some properties are preferred, multi-variable preference by fuzzy transformation can be used, which can be constructed from several fuzzy transformations constructed for each preferred value. This results in performing decision on fuzzy transformation level. One of the most significant issues in economical applications of MAS is cheating. If agents can predict decisions then they can change their properties. Thus, they knowingly lie. Agents, who have same interests or common goal, can interchange knowledge about others to reveal cheating. Afterwards, cheaters should be excluded from next considerations or are penalized.

The main aim of the future work is to implement FVC into management decision support system of national electric grid that utilizes experience management in process of coalition formation.

3. Modelling of social environment in multi/agent systems

MAS are appropriate to model and control huge systems in many different applications. Besides manufacturing, social environment and virtual societies can be modelled via MAS.

Suppose that social environment contains, for example, the cultural, educational, business, traffic etc. parts. Each of them may be characterized by certain phenomena, behaviours, properties, etc., and may be considered as an autonomous complex with its own norms, relations, rules, aims, conflicts, communications, working way, etc.

From this aspect their definitions have different meaning in various areas, such as social science, psychology, legal theory, biology, medicine, ecology, education, production etc. To solve their performance requires special tools, which are able to handle the appeared problems. As one of the possible tools multi-agent system (MAS) can be used, because:

- multi-agent systems are evolved as a methodical solution to large complex distributed problems
- an agent is a hardware or more usually software entity that enjoys the properties such as autonomy, social ability, reactivity and pro-activeness,
- each agent in MAS is considered rational and autonomous in making decision for improving its individual benefit.

According to the above considerations the paper will be focused to the application of MAS to handle some special problems of society performance. Each society is characterized by the norms (N) in a population (P), which is defined as a function of beliefs and preference of the P members, if the following conditions hold :

- almost every member of P prefers to conform to N on the condition that almost everyone else also conforms,
- almost every member of P believes that every other member of P conforms to N.

The above conditions may be translated to MAS as follows:

- norms as obligations,
- norms as requirements,
- norms as goals or objectives (deduced from obligations and requirements),
- norms as constraints on behaviour.

Depending on the membership in agent coalition which corresponds to any society in P and its individual requirements and intentions, an agent is confronted with a set of norms, based on the situation of the society, to which the agent belongs. This agent has to strictly abide by norms, and for other norms it has to consider an adoption strategy to decide whether to comply.

In production systems and intelligent engineering research, virtual space can be taken as analogy to the physical intelligent space .

In social research, norms are considered as responsible for regulating the social behaviour of individuals. They prescribe how to interact and resolve issues in conflicting circumstances. In MAS this case is solved by the rules of negotiation. The negotiation form is an important part of social agent: agents need to cooperate and organize their activities to achieve their individual as well as collective goals.

On the basis of this consideration the challenges addressed by this paper can be formulated as follows:

- building the knowledge in the multi-agent system's environment
- incorporating knowledge system into multi-agent system environment
- management of virtual societies
- experience retrieval from the experience base.
- conclusion and future work.

One objective of multi-agent system's environmental engineering is to implement knowledge into environment in order to describe

- world of interest,
- environmental knowledge,
- agent system.

This implementation should have to follow the environmental reference model , which divides environment into several active and passive elements supporting communication between agent system and world of interest. Furthermore, it includes the module for processing internal processes over data repository known as State concluding state of the environment.

In general, the knowledge system model consists of three layers, namely:

- conceptual,
- contextual,
- product layer.

These three layers include the following six models: organizational, task, agent, knowledge, communication and design model. Organizational model describes setting where the knowledge system is deployed. The model is spread over five subsystems of the environment: synchronization, observation, interaction, perception (cover concepts parts of the world of interest, which are taken into account by the knowledge system) and communication.

Task model represents all assignments which have to be done by executive parts of the knowledge system.

Agents defined in agent model are called environmental processes. Each environmental process has one or more tasks assigned, which is/are defined in the task model.

Knowledge represented in knowledge system of environment is described within the knowledge model which partially results in a part of a law model describing principles of the world of interest.

Communication model describes interactions among environmental processes working with knowledge and fulfilling the tasks defined by the task model.

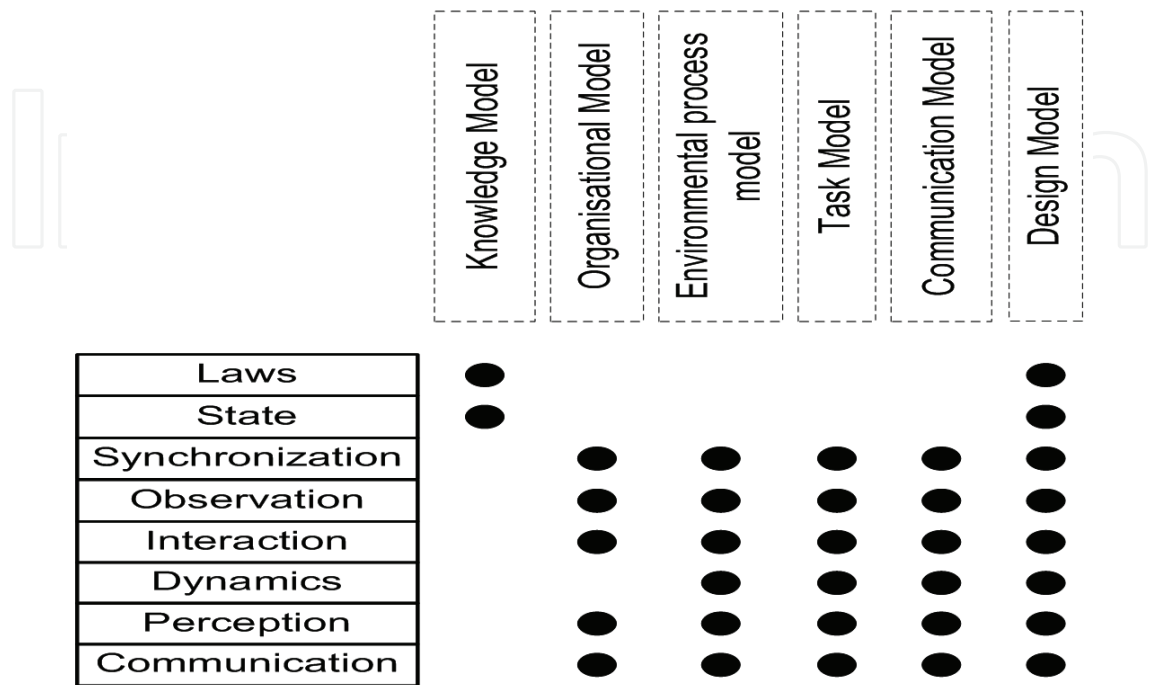The design model represents implementation of the knowledge system into the settings.

	Knowledge Model	Organisational Model	Environmental process model	Task Model	Communication Model	Design Model
Laws	●					●
State	●					●
Synchronization		●	●	●	●	●
Observation		●	●	●	●	●
Interaction		●	●	●	●	●
Dynamics			●	●	●	●
Perception		●	●	●	●	●
Communication		●	●	●	●	●

Fig. 5. The relation matrix between reference model of the environment and models of the knowledge system

The knowledge model is distributed over whole MAS environment which is shown in the following figure utilizing interaction matrix. Figure 5 presents two types of boxes, namely solid and dashed ones. Solid boxes represent parts of the environment's reference model. On the other hand, dashed boxes encompass knowledge system models.

The set of executed processes depends on application. Valckenaers et al. in (Valckenaers, 2007) present three types of applications:

- emulation; in emulation application, environment is off line emulator of the world of interest
- interacting information system communicates with the world of interest in real-time
- adaptive information system is interacting information system dynamically adapting upon changes in the world of interest.

From this aspect the main tasks are as follows:

- mapping knowledge into MAS environment through experience management of virtual societies in a multi-agent system,
- addressing three experience management processes, namely:
- experience,
- item representation,
- storing and retrieval.

Management of virtual societies

Agents in multi-agent system form structured virtual societies with different organization and purposes.
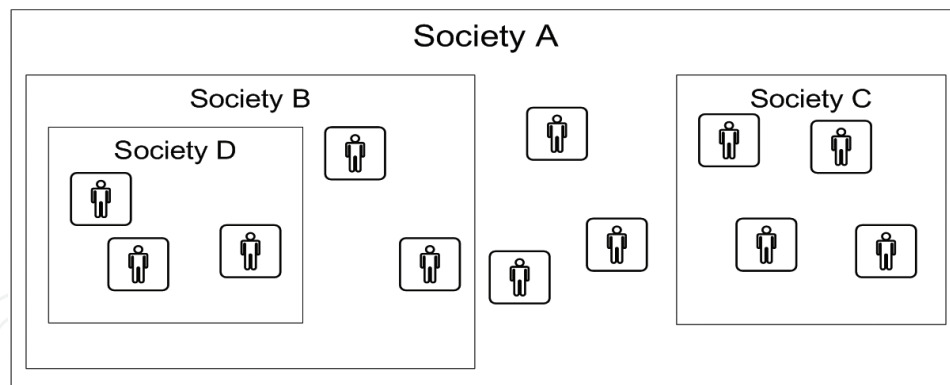
Fig. 6. Conceptual description of structured agent society is presented. Society A is top-level society in this example which encompasses three agents and two societies, namely Society B and Society C. The important result of this description is hierarchical distribution of agents over nested societies.
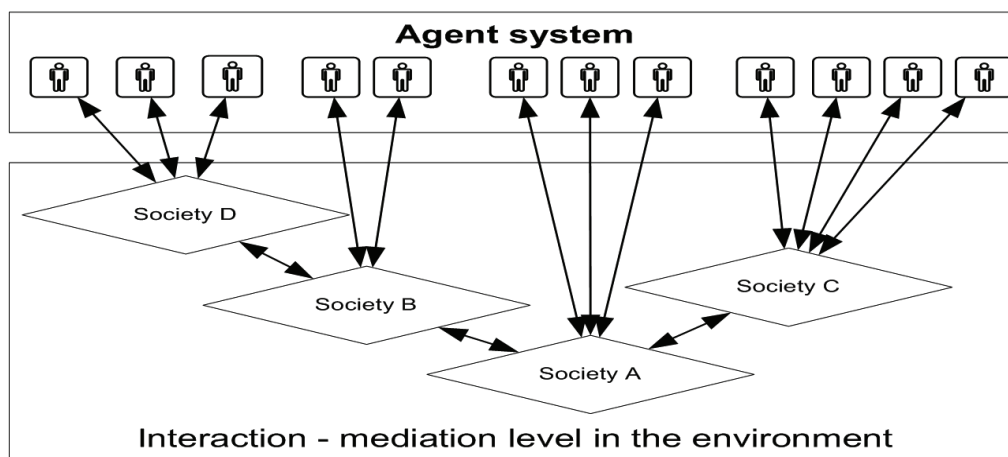

Fig. 7. Vertical section of the multi-agent system of nested agents societies.

The interaction and mediation levels are taken into account in following way:

• virtual societies are distributed over the agent system and its environment has impact on the expansion of functionalities of the environment,

• vertical cut over virtual society shows that it is distributed over agent system and its environment.

This is an important fact, because structure of a society is developed in the environment, not in the agent system, which is just a container encompassing all agents on one place. Developed societies can be registered to the environment; afterwards the environment can manage them. In this point, two approaches to manage societies - free and controlled - can be recognized: Free management of agent society just manages existence of the society and does not control agent behaviour in the agent system. On the other hand, controlled agents' society management can control behaviour of agent through communication among agents and virtual societies. Societies are built according to a precisely defined problem which is the goal of the directive or negotiation formation process. Agents in a society take actions which can be described by a plan, (in negotiation processes actions are usually generated ad-hock) The plan of actions and the plan of outcomes' distribution are properties of the society. Thus, society can be described by a concept with the following properties:

• an activity defining the aim of the society;

- a plan describing sequence of the activities taken by entities in the society, or distribution of the outcomes;
- a set of entities located in the society;
- a relation to upper level society.

The concept entity encompasses both agents and societies. The term activity includes one of the following concepts:

- an objective addressed by societies,
- a task solved by agents,
- an action executed by agents.

For mathematical description of society with action plan, propositional logic can be used. Since societies are oriented towards actions, it is proper to use concurrent dynamic epistemic logic.

For our purposes, the sequential, parallel and conditional execution and testing of proposition are the most important.

According to logic agents, such societies can be described by the following sorted set of four elements:

$$S = \left(\Sigma, F, F^0, \Sigma_S^0 \right)$$

Where:

- Σ denotes all agents and societies in the MAS,
- F represents a set of all formulas in particular propositional logic,
- F^0 represents the set of all formulas in the particular propositional logic including $\varnothing$, this means that the society has not defined a plan of actions,
- Σ_S^0 defines the set of al societies existing in the MAS and symbol $\varnothing$ denotes that defined society S is the top-level society and has no parent society.

The second and third elements represent activities of society S. However, the former element defines the aim of the society, and the latter introduces an optimal pal, if there s any.

Example of the society description is in the following formula:

$$\left(\{S_2, A_2, A_3\}, [? p_1] p_2, [? p_1; \alpha_1; \alpha_2] p_2, S_0 \right)$$

where society including two agents A_2, A_3 and society S_2 is defined. This society has an objective to satisfy proposition p_2 when p_1 is satisfied. Optimal plan of actions to satisfy the objective of the society is also defined. The parent society is defined as well, using the fourth element of society definition.

The above presented knowledge system distributed over the MAS environment is used to manage virtual societies of agents. The knowledge system includes ontology of the virtual society derived from mathematical representation that presents idea of defining society structure in the environment rather than in agent system. Similar MAS approach may be effectively applied in a framework of extended social field, for example, in business areas because it deals with continuous input of varying information and the maintenance of distributed databases over the intra and inter nets.

4. Multi agents´ control system inspired by immune systems

4.1 Artificial immune systems

Study of biological principal as a new solution in computer science become very popular in last two decades. There are many popular approaches with biological background applied in

computer science. One of the most often used algorithms is neural network, which is inspired by neuro-science result. Also genetic algorithm and cellular engineering are used with great success in computer science. Another, in last years very popular area of research, is artificial immune systems (AIS) based on principals of mammal immunology. Artificial immune system imitates the natural mammal immune system that has sophisticated methodologies and capabilities to build computational algorithm that solves engineering problems efficiently. The main goal of the mammal immune system is to protect the internal components of the mammal body by fighting against the foreign elements such as the fungi, virus and bacteria. Mammal immune system can be dividing in to the two basic immune responses:

- Innate immune system
- Adaptive immune system

Innate immune system consists of several barriers that protect organism from infection. Typical barriers are mechanical, chemical and biological barriers. These barriers are the first line of defence against infection.

Artificial immune systems are based on metaphor to adaptive immune system. Adaptive immune systems are composed of a great variety of molecules, cells, and organs spread all over the body. There is no central organ coordinating functions of the immune system, and there are several elements performing complementary roles. The main task of the immune system is to survey the organism in the search for malfunctioning cells from their own body (e.g., cancer and tumour cells), and foreign disease causing elements (e.g., viruses and bacteria) (de Castro & Timmis, 2002).

4.2 Adaptive immune systems and their metaphors

Elements that can be recognized by immune system are called antigens. On the other side, cells that originally belong to our body are harmless to its functioning are termed self antigens and disease causing elements are named non-self antigens. One of the main functions of the immune system is to be capable to distinguish between what is self and what is non-self antigens. There are many types of immune cells but two of them are necessary to know, to understand basic principals B-cells and T-cells. These two types of cells are rather similar, but differ with relation to how they recognise antigens and by their functional roles. B-cells are capable to recognise antigens free in solution (e.g., in the blood stream), while T-cells require antigens to be presented by other accessory cells (de Castro & Timmis, 2002). Four basic metaphors have been taken from the mammal immune systems and applied to the artificial immune systems:

- *Positive and negative selection*
- *Clonal selection*
- *Jerne's idiotropic network*
- *Danger theory*

Positive and negative selection.

The mammal Thymus is pleased where the T-cell maturated. Process of maturating consists of removing T-cells which attack self antigens. If they escape this elimination, they may subsequently react against self antigens, and cause Autoimmune disease. Negative selection was introduced in 1994 (Forrest et al., 1994), and is now often called negative detection. The basic principle of negative selection is to find sets of individual which does not react with any self individuals. So, when in future will be present non-self antigen, some of the individual form set of T-cell will be reacting (eliminate) with it. Positive selection is opposite process. Immune cell will be exposed to the set of non-self antigen and tries to learn their characteristics.

Danger theory.

There appeared, in few experimental observations, cases, which can not be explained by original self-non-self theory. For example, how can be explained the fact that bacteria in our gut, which is clearly non-self pathogen, does not start immune reaction. Matzinger has developed an interesting alterative theory (Matzinger P. (1994).) to explain that phenomen. Immune cell can die in two ways. The first one is normal death that has been requested by the body's internal signaling system, and the second is via unexpected death caused by something going wrong with the cell, which often causes an inflammatory response. Matzinger suggested that the immune system is particularly activated by cell necrosis. It appears that danger theory might help intrusion detection systems by focusing attention only on those internal or external events that are likely to be harmful in general a smaller subset of events than the non-self subset.

Clonal selection

Clonal section focuses on how immune cell can adapt to the new match and how can eliminate invaders. Clonal selection is responsible for the ability of mammal immune systems to adapt its B-cell to new types of non-self antigens. There is more variants of clonal selection in area of artificial immune systems. One of the main algorithm that represents this group of approaches is COLNALG(de Castro and Von Zuben, 2002).The CLONALG algorithm is as follows:

1. Randomly initialise a population of individuals (M);
2. For each pattern of P, present it to the population M and determine its affinity (match) with each element of the population M;
3. Select n1 of the best highest affinity elements of M and generate copies of these individuals proportionally to their affinity with the antigen. The higher the affinity, the higher the number of copies, and vice-versa;
4. Mutate all these copies with a rate proportional to their affinity with the input
5. Add these mutated individuals to the population M and re-select n2 of these maturated (optimised) individuals to be kept as memories of the system;
6. Repeat Steps 2 to 5 until a certain criterion is met, such as a minimum pattern recognition or classification error.

Jerne's idiotropic network

Jerne's idiotropic network deals with the interaction of antibodies. Jerne's network is a network of B cells that communicate the shape of the antigenic epitope amongst them through idiotopes and paratopes. There is few models (Farmer et al., 1986)(Fukuda et al., 1998)(Timmis et al., 2000) of idiotropic network used in artificial immune system.

4.3 Relation between AIS and MAS, case study

There are two basic points of view on interaction between AIS and MAS. The main difference between these two concepts is question which technology could be used as a tool. The basic concept is to use MAS as a powerful modelling tool for AIS simulation. Agents simulate cell of an immune system and interaction between cells are reflected in to the interaction between agents. This concept is used in process of understating some disease. Typical application is simulation of reaction of the immune system on viruses (Bernaschi & Castiglione, (2002)). The second concept is, when metaphors that form mammal immune system are used to solve problem of agent system. Mammal immune system is robust, distributed, decentralized, self-organized and adaptive complex. All these features are very important in multi-agent funkcionality. This is reason, why artificial immune system is more and more applied on multi-agent paradigms.

Security task is one of the common application areas of the artificial immune systems in multi-agent system. The security problem is very similar to the specific immune response of

mammals. There is the same task in the network security and the immune response, to protect system against intruders (pathogen). All types of immune responses principals are tried to be used in the area of security. With the development of the network security technology, there are more an more equipments to protect network systems. Especially for large network system, the network architecture becomes more and more complex and dynamic. In the dynamic network environment, agent is suitable for collecting data information and constructing the complex fusion model. In work (Zhou et al., 2007) authors try to solve problem of the large network security thru the application of the danger theory (Matzinger, 1994). In the paper they show, the Danger Theory can be successfully applied to the field of network security.

Interesting work in this field is Nishyama's research (2001). Authors try to build a system to detect and reject unauthorized intrusion using security technology similar to the flexible structure of an immune system. They applied the recognition mechanism of self and non-self of an immune system and cooperation among immune cells. In such a way they were able to realize a security system that automatically detects and eliminates unauthorized intrusion from inside or outside. Authors extend the multi-agent language MRL (Nishiyama, 1998), which design is based on a parallel-logic programming language and apply it to implementing an immune security system. System is based on B-cells and T-cells functionality and on thir cooperation. When connection is acquired system generate B-cell agent for this connection, so B-cells response to outside connection. The role of B-cell agent is to watch at the input of the connection. When activity comes from inside, some process starts or some file is created and system generates T-cell agent which watch at these processes. When network connection invocateds some process or file creation, then connection between B-cell and T-cell will happen.

Another interesting point of view of artificial immune systems usage in multi agent system is reported in paper (Bakhouya & Gaber, 2007). Authors try to use agent, based on AIS algorithm to create a network for computer-based human interactions. Basic principle of AIS, the affinity is used for representation of relationship between agents. The learning mechanism is used to adjust the affinity relationships between agents representing the services to cope with the user context and provoke or produce emergent services. In this approach, the mechanism of establishing affinity relationships is based on keywords similarities.

One of the parameters which is hard to set in large scale multi-agent systems is population size. It is difficult to estimate the appropriate number of agents allowed to be spawned in the network, when mobile agents operate in a dynamic and distributed environment. Trying to increase robustness and efficient of the MAS in heterogeneous environment through the rising population of agent may cause, on the other hand, increasing resource demands of the network. In work (Bakhouya & Gaber, 2006) authors try to solve population size problem through artificial immune system approach. The main idea is to use life cycles of B-cell in immune network to dynamically set size of population. In this adaptive approach, agent retrieves local information from its environment and makes appropriate decision to either kill, or move without cloning or clone itself and move to another node of the network.

Very interesting work in the area of AIS application in MAS is described in (Sathyanath & Sahin, 2002). There are AIS and MAS applied in interesting field of mine detection problem. All the self-agents work in an agent network similar to Jerne's network. The process of information transfer and communication between the agents is an analogy of the agent network to the immune network. The nature of the agent network is application dependent. Also in typical application domain of MAS technology are artificial immune system applied. In paper (Lee & Suzuki, 2006) authors present how negative and positive selection in process of improving performance and workload balancing can be used. Authors call the

network of agent in this work, which simulate cell of immune system, iNet. Each agent continuously senses a set of current environment conditions as an antigen and examine whether it is self or non-self. A self antigen indicates that the agent adapts to the current environment conditions well, and a non-self antigen indicates it does not. When environment evaluation facility detects a non-self antigen, it activates behaviour selection. Behaviour selection allows each agent to choose behaviour as an antibody that specifically matches with the detected non-self antigen. Antibodies are situated in idiotope Jerene's network. Authors in result describe that iNet artificial immune system, allows each grid service to autonomously sense its surrounding environment conditions to evaluate, whether it adapts well to the conditions, and if it does not, adaptively perform a behavior suitable for the conditions. Also the simulations show that iNet allows agents to autonomously adapt to dynamic environmental changes for improving their performance and balancing workload. Presented case studies show great potential of AIS application in field of MAS. In future AIS can be helpful in a solving of the new challenges in area of multi-agent system. Especially in tasks, which require fault tolerance, self- organizing and self-adapting.

5. References

Balch T. (2000) Hierarchic Social Entropy and Behavioral Difference: New Measures of Robot Group Diversity, NIST Workshop on Metrics for Intelligent Systems, Gaithersburg, July, 2000

Bakhouya, M. and Gaber, J. (2006). Adaptive Approach for the Regulation of a Mobile Agent Population in a Distributed Network. In *Proceedings of the Proceedings of the Fifth international Symposium on Parallel and Distributed Computing* (July 06 - 09, 2006). ISPDC. IEEE Computer Society, Washington, DC, 360-366.

Bakhouya, M.; Gaber, J.,(2007). A Propitient Multi-agent System for Spontaneous Service Emergence in Pervasive Computing environments, *Pervasive Services*, IEEE International Conference on , vol., no., pp.409-414, 15-20 July 2007

Bernaschi, M., Castiglione,(2002). F., Selection of Escape Mutants from Immune Recognition During HIV Infection, *Immunology and Cell Biology Journal*, 80, february 2002, 307–313

Bergman (2002) Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Springer

Budinská I., Dang T.T. (2004) A Case Based Reasoning in a Multi Agents Support System, Proceedings. of the 6th International Scientific – Technical Conference, Process Control 2004, Kouty nad Desnou, Czech Republic, June 8-11, 2004, Eds.: Stanislav Krejčí et.al, Editorial University of Pardubice, pp. 57, ISBN 80-7194-662-1

Cantamessa, M. and Villa, A.: Negotiation Models and Production Planning for Virtual Enterprises. In Proceeding of The IFAC Workshop on Manufacturing, Modeling, Management and Control – MIM 2001, Prague, Czech Republic, 1-5, 2001.

Dang, T-Tung, Frankovic, B. and Budinska, I.: Agents Coalition in Coordination Process. In XV IFAC World Congress b'02, Barcelona, 2002.

Dang T.Tung, Frankovič, B., Budinská, I. (2003) Optimal creation of agent coalitions for manufacturing and control. In Computing and informatics. ISSN 0232-0274, Vol. 22, No.1

de Castro, L. N. and Von Zuben, F. J. (2000). The clonal selection algorithm with engineering applications. *In Proceedings of GECCO'00*, Workshop on Artificial Immune Systems and Their Applications, pages 36–37.

de Castro, L.N & Timmis, J. (2002). Artificial Immune Systems: A Novel Approach to Pattern Recognition, In: *Artificial Neural Networks in Pattern Recognition Corchado, L Alonso and C Fyfe*, 67-84, University of Paisley, ISBN 84-95721-22-8

Farmer, J. D., Packard, N. H., and Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Phys.* D 2, 1-3 (Oct. 1986), 187-204.

Fukuda, T., Mori, K., and Tsukiyama, M. (1993). Immune networks using genetic algorithm for adaptive production scheduling. *In 15th IFAC World Congress*, volume 3, pages 57–60.

Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994). Self-Nonself Discrimination in a Computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, PP.132-143, May 16 - 18, 1994, SP. IEEE Computer Society, Washington, DC.

Horváth László, Rudas Imre, J.: Virtual Space for Intelligent Engineering in Mechatronics, In Proc. Of IEE Conference ICM 2006, Budapest Hungary, pp. 248-253

Lee C. and Suzuki, J. (2006) Biologically Inspired Design of Autonomous and Adaptive Grid Services In *Proc. of the 2nd IEEE International Conference on Autonomic and Autonomous Systems (ICAS)*, Santa Clara, CA, July 2006.

Matzinger P. (1994).Tolerance "Danger and the Extended Family", Annual Review of Immunology. 1994, pp. 911-1045

Nishiyama, H., Ohwada, H. and Mizoguchi, F. (1998). A Multiaget Robot Language for Communication and Concurrency Control, International Conference on Multiagent Systems, pp. 206–213, 1998.

Nishiyama, H. and Mizoguchi, F. (2001). Design of Security System Based on Immune System. In *Proceedings of the 10th IEEE international Workshops on Enabling Technologies: infrastructure For Collaborative Enterprises* (June 20 - 22, 2001). 138-143,WETICE. IEEE Computer Society, Washington, DC.

Nof, S.Y., G. Morel, L., Monostori, A. Molina, F. Filip, (2005) "Milestone report for CC5 Manufacturing & Logistics Systems: From plant and logistics control to multi-enterprise collaboration," Proceedings of IFAC Congress , Prague , CZ.

Odubiyi, J. B., Kocur, D. J., and Weinstein, S. M.: A Scalable Agent-Based Information Retrieval Engine. In Proceedings of the First Annual Conference on Autonomous Agents. California USA, ACM Press / ACM SIGART, 292-299, 1997.

Sandholm, T. (1999) Distributed Rational Decision Making In the textbook Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence, Weiß, G., ed., MIT Press. Pp. 201-258

Sathyanath, S., Sahin, F. (2002) "AISIMAM – An AIS based intelligent multi agent model ands its pplication to a mine detection problem", in *Proceedings of the ICARIS 2002 1st international Conference on Artificial Immune Systems*, Canterbury, UK, September 9 – 11, 2002

Shehory, O., Kraus, S. (1999) Feasible formation of coalitions among autonomous agents in non-super-aditive environments, in Computational Intelligens, Vol. 15 (3), August 1999, pp 218-251

Timmis, J., Neal, M., and Hunt, J. (2000). An artificial immune system for data analysis. *Biosystems*,55(1/3):143–150.

Valckenaers P., Sauter J., Sierra C., Rodriguez-Aguilar J. A. (2007) Application and environments for multi-agent systems, Journal of Autonoumous Agents and Multi-Agent Systems, Vol 14, pp. 61-85, DOI 10.1007/s10458-006-9002-5, Springer

Wellman, M. P.: A MarketOriented Programming Environment and its Application to Distributed Multicommodity Flow Problems. In Journal of Artificial Intelligence Research 1, 123, 1993.

Wooldridge, M., Bussmann, S. and Klosterberg, M.: Production Sequencing as Negotiation. In Proceedings of the First International Conference on the Practical Application of Agents and Multi-Agent Technology (PAAM--97), 709-726, 1996.

Wooldridge, M. (2002) An introduction to Multiagent systems, John Willey & Sons, 2002.

Multiagent Systems

Edited by Salman Ahmed and Mohd Noh Karsiti

ISBN 978-3-902613-51-6

Hard cover, 426 pages

Publisher I-Tech Education and Publishing

Published online 01, January, 2009

Published in print edition January, 2009

Multi agent systems involve a team of agents working together socially to accomplish a task. An agent can be social in many ways. One is when an agent helps others in solving complex problems. The field of multi agent systems investigates the process underlying distributed problem solving and designs some protocols and mechanisms involved in this process. This book presents an overview of some of the research issues in the field of multi agents. It is a presentation of a combination of different research issues which are pursued by researchers in the domain of multi agent systems as they are one of the best ways to understand and model human societies and behaviours. In fact, such systems are the systems of the future.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Baltazár Frankovič, Than Tung Dang, Tomáš Kasanický, Viktor Oravec and and Ivana Budinská (2009). Agent Oriented Engineering and Methodologies with Application to Production, Economical and Social Systems, Multiagent Systems, Salman Ahmed and Mohd Noh Karsiti (Ed.), ISBN: 978-3-902613-51-6, InTech, Available from:

http://www.intechopen.com/books/multiagent_systems/agent_oriented_engineering_and_methodologies_with_application_to_production__economical_and_social_s

INTECH
open science | open minds