INFORMATYKA EKONOMICZNA BUSINESS INFORMATICS 3(25) • 2012
ISSN 1507-3358

Jadwiga Sobieska-Karpińska, Marcin Hernes
Wrocław University of Economics, Poland
e-mail: {jadwiga.sobieska-karpinska, marcin.hernes}@ue.wroc.pl

KNOWLEDGE CONFLICTS RESOLVING IN THE MULTI-AGENT DECISION SUPPORT SYSTEM USING MULTI-STAGE CONSENSUS DETERMINING[1]

Abstract: The present paper describes using a multi-stage consensus algorithm to knowledge conflicts resolving in multi-agent decision support systems. The problem of knowledge conflict between agents, structure of decision, profile and criteria of consensus determining are presented in the first part of the article. Next, a two-stage algorithm of consensus determining is elaborated. This algorithm, among other things, allows shortening the period of time necessary to take a decision and limiting the risk associated with this process as well as it leads to increased effectiveness of decision-making since solutions generated by agents of an inadequate level of knowledge are not taken into account.

Keywords: multi-agent systems, decision support system, knowledge conflicts, consensus algorithm.

1. Introduction

Precise and swift decision-making is what makes companies competitive in the present economic climate. When considering a company's prospects and future, the industry forces management to make complex tactical, operational and strategic decisions. The people responsible for such decision-making operate in a climate of risk and uncertainty as they are unable to foresee the end results of their actions. This is what makes the decision-making process extremely complicated.

The decision-making process is complemented by various, multi-agent computer systems that consist of at least a few or multiple programs (agents) whose purpose is to present the user with a solution (decision) to the particular problem. An agent is an autonomous object with a defined purpose, capable of communicating with other agents; the one that acts on its own and reacts to the changes in its environment [Soto et al. 2009]. Multi-agent systems allow for fast data collection, information processing and presenting the person in charge of the decision-making process (the

[1] Selected parts of this article were published under non-exclusive copyright in *Proceedings of the Federated Conference on Computer Science and Information Systems FedCSIS 2012* (see [Sobieska-Karpińska, Hernes 2012]).

decision-maker) with proposed acceptable results (decisions). What one means by "acceptable results" is the decisions that fulfill the decision-maker's conditions, while "optimal decisions" are those that are the most favorable for the assessment criteria set out by the decision-maker. The final decision, however, is always made by the decision-maker, thereby making him or her solely responsible for the outcome. Multi-agent systems that support decision-making processes significantly reduce the time required for performing the task due to their ability to process information, make data selection, use stored information and adequately react in order to supply the decision-maker with a variety of solutions.

It is often the case that the multi-agent system generates various conflicts – knowledge conflicts in particular – as the user may be presented with a number of various solutions (decisions). Knowledge conflicts arise from the variety of methods used by agents in the decision-making process. If the system generates a data conflict, it is impossible for it to produce an adequate decision, thus leaving the decision-maker without assistance. The decision-maker is then forced to make the decision without the system's help. This may be time-consuming, labor-intensive, and carry the risk of making the decision with the use of outdated and incomplete information. Clearly, this may unfavorably impact on the functioning of the entire organization.

The key element in the functioning of multi-agent systems is the early detection and recognition of knowledge conflicts, as is their adequate resolution. The literature on the subject presents a variety of methods of conflict resolution, e.g. negotiation or counseling. None of these methods, however, are perfect due to their computing complexity or other factors.

Consensus methods are commonly used – they allow singling out, from the many decisions generated by all agents, the one presented to the user. In other words, the decisions made by all agents are taken into consideration.

None of the works on the subject, however, mention the problem of a better resolution of the knowledge conflict. It is, indeed, quite an important issue as, while the consensus methods are being set up and decisions of multiple agents with potentially faulty or incomplete input information are considered, the user ends up with the inferior-quality result (decision) generated by the system. The solution to this problem is the implementation of a multi-stage consensus that allows bypassing/eliminating the decisions generated by agents with incorrect input/information. This article presents an algorithm of a two-stage consensus setup allowing for the knowledge conflict resolution and, at the same time, including the aspect of the improvement of the agent's knowledge.

2. Conflicts of knowledge

It is a common occurrence that in a multi-agent decision support system, agents generate different versions of solutions, resulting in conflicts of agents' knowledge.

Such situations may arise due to various reasons. Agents may use different sources of information, or they may employ different decision support methods. Knowledge conflicts concern situations in which different values are assigned by parties to a conflict with the same world objects and features [Nguyen 2002].

Knowledge conflicts in multi-agent systems stem from inconsistencies or contradictions among agents' data inputs. An inconsistency occurs when one agent acknowledges the presence of one attribute of the environment in a certain time frame, while another does not possess any relevant information, or is not willing to discuss the subject. A contradiction occurs when one agent acknowledges the presence of a given attribute of the environment in a given time frame, while another agent does not [Nguyen 2002].

Knowledge conflicts appear when both parties attribute different features to the same objects of the environment or different values to the same features. This operates on the assumption that there is a structure to the agents' information/data.

The following sources of knowledge conflicts were mentioned in the Nguyen 2002 paper:

a) The conflict regarding the rights to management of resources occurs when one of the parties assumes that the other side is not entitled to rights and information about the given resources, while the other challenges this position.

b) An ideological conflict occurs when both parties present conflicting positions on an issue/subject. These convictions may stem from the particular characteristics of the environment of a given system, or from its function.

c) When integration of various elements of the system is required, it automatically creates a conflict due to the varying structures of data/information and their presentation.

d) Conflicts stemming from the data/information management system occur when both parties assert their right to manage data/information resources stored in the system.

Clearly, the sources of knowledge conflicts are numerous and one can find a lot of research on the attempts to seek out and solve those problems. It is an important issue, especially when one considers the fact that when the system fails, it is up to the user to take on this task. The user, however, expects one uniform version of the system or, rather, a single decision. It is advisable, having analyzed all options, to present one solution that fulfills all user requirements. Simply put, one needs to resolve the knowledge conflict.

In various books on the subject, one may come across many methods of solving such conflicts. For example, Ferber [1999] suggests an arbitration method in which a conflict is solved by the system itself, without the participation of agents. De Long, Seemann [2000], however, suggest using the method of negotiation between agents. Agents communicate between one another in order to establish and agree on a single, uniform state of knowledge. Negotiations, however, are time-consuming and result in slowing down the system's operation, which obviously affects the effectiveness of decision-making.

Many papers, among others [Nguyen 2002; Sobieska-Karpińska 2011a], suggest using consensus methods to solve knowledge conflicts. Such methods enable selecting one solution (in this case, a decision) out of many alternatives. A decision selected by means of consensus methods does not have to be one of the decisions generated by the system. It may just be very similar to them. In the consensus methods, every party is taken into account, every party to a conflict "loses" as little as possible, every party contributes to the consensus, all parties accept the consensus and the consensus is representative of all the parties to a conflict [Sobieska-Karpińska, Hernes 2011b]. So far, determining a consensus has most often taken place within a single stage (after generating a decision by individual agents).

Korczak, Lipiński [2008] found that in order for a multi-agent system to function properly, agents have to continuously improve their knowledge. The knowledge may be improved in a variety of methods. Paper [Winikoff 2005] suggests a method which presupposes that the code of an agent's program contains monitoring procedures which verify whether an agent's knowledge is sufficient to its proper functioning. These procedures check the effectiveness of the agent's actions by monitoring results that it produces. However, such a method has some flaws. Firstly, an agent who goes to any lengths to achieve its aim may overestimate its knowledge. Secondly, in the case of malfunctioning agents, knowledge monitoring procedures may work inadequately which will result in an inadequate knowledge condition. That is why Parsons, Klain [2004] suggest a method which assumes that there are additional agents in the system which analyze the behavior of individual agents, and determine their current state of knowledge. What is among the numerous advantages of such an approach is the impartiality of knowledge assessment and resistance to failures of individual agent programs.

In a one-stage process of establishing a consensus, the decisions generated by agents with the correct data/information are equally considered with the ones of those with an incorrect input. This may cause the "deterioration" of the decisions presented to the users – in other words – a one-stage setup of a consensus does not allow for the improvement of the quality of information on the part of the agents.

In order to solve this type of problem, a multi-stage consensus determining method may be used. The method has been defined in [Nguyen 2002]; however, it has not yet been employed in practical solutions.

In the following part of the article the two-stage consensus determining algorithm will be elaborated on. This algorithm allows for resolving knowledge conflicts and agent's knowledge resolving and consequently makes the decision support process more effective.

3. Two-stage consensus determining algorithm

In the subject literature, a consensus is defined as a resolution, which is a compromise determined on the basis of existing resolutions. There are several phases of finding consensus. First, the exact structure of the set decision generated by agents must be specified. Next, the distance between these decisions is calculated. Finding consensus is choosing a decision for which the distance between this decision (consensus) and decisions generated by agents is minimal (according to different criteria).

To elaborate a two-stage consensus, it is necessary to elaborate the structure of decision first. The structure of decision representation is a set of decision elements which describe the real world, e.g. suppliers, customers, products, etc. These elements (attributes) are ordered in a sequence of proceeding during decision realisation. A formal definition of the structures is presented in [Sobieska-Karpińska, Hernes 2006]:

Definition 1.

The structure decision P of a finite set of decision elements $E = \{e_1, e_2, \ldots, e_Y\}$ is called a sequence:

$$P = \left\langle \{EW^+\}, \{EW^\pm\}, \{EW^-\}, Z, SP, DT \right\rangle,$$

where:

1) $EW^+ = \left\langle e_o, pe_o \right\rangle, \left\langle e_q, pe_q \right\rangle, \ldots, \left\langle e_p, pe_p \right\rangle;$

couple $\left\langle e_x, pe_x \right\rangle$, where: $e_x \in E$ and $pe_x \in [0,1]$, denote a decision element and this element's participation in set EW^+;

decision element $e_x \in EW^+$ will be denoted by e_x^+.

The set EW^+ is called a positive set; in other words, it is a set of decision elements about which the agent knows that these elements are in the environment.

2) $EW^\pm = \left\langle e_r, pe_r \right\rangle, \left\langle e_s, pe_s \right\rangle, \ldots, \left\langle e_t, pe_t \right\rangle;$

couple $\left\langle e_x, pe_x \right\rangle$, where: $e_x \in E$ and $pe_x \in [0,1]$ denote a decision element and this element's participation in set $EW^\pm$.

Decision elements $e_x \in EW^\pm$ will be denoted by $e_x^\pm$.

The set $EW^\pm$ is called a neutral set; in other words, it is a set of decision elements about which the agent does not know that these elements are in the environment.

3) $EW^- = \left\langle e_u, pe_u \right\rangle, \left\langle e_v, pe_v \right\rangle, \ldots, \left\langle e_w, pe_w \right\rangle.$

Couple $\left\langle e_x, pe_x \right\rangle$, where: $e_x \in E$ and $pe_x \in [0,1]$, denote a decision element and participation of this element in set EW^-.

Elementary objects $e_x \in EW^-$ will be denoted by e_x^-.

The set EW^- is called a negative set; in other words, it is a set of decision elements about which the agents knows that these elements are not in the environment.

4) $Z \in [0,1]$ – rate of return in percent.

5) $SP \in [0,1]$ – degree of secure of rate *Z*.

6) DT – date of knowledge.

The presented decision definition allows formulating system nodes conclusions in a uniform structure. It can happen that an agent does not "know" whether a given element of decision can be used or not (for example the agent has not received information about the given value paper). Therefore, set $EW^{\pm}$ is necessary. The presented structure is complex and multi-value; there are different data types and multi-attribute functional dependencies in this structure.

A situation in which the structures of a decision in the system differ, or the values of their attributes are different, is called an inconsistency in the state of knowledge of these agents. In this case consensus methods can by applied to coordinate the state of knowledge [Nguyen 2002].

It should be noted that one faces the choice of possible solutions in the decision-making process. When one acts under conditions of uncertainty or risk (when one cannot determine the consequences of one's decisions), wrong/incorrect decisions can be made. The use of consensus methods broadens one's choice by seeking out and adding new solutions, thus decreasing the risk in decision-making.

The outcome of the consensus method application is a good representation of the set of decisions, because it takes into consideration practically all the subsets of the set. We call a set of such decisions a profile and define it as follows [Sobieska-Karpińska, Hernes 2011a]:

Definition 2.

A set of decision elements $E = \{e_1, e_2, \ldots, e_Y\}$ is given.

A profile $A = \{A^{(1)}, A^{(2)}, \ldots, A^{(M)}\}$ is called a set of *M* decisions of a finite set of decision elements *E*, such that*:*

$$A^{(1)} = \left\langle \{EW^+\}^{(1)}, \{EW^{\pm}\}^{(1)}, \{EW^-\}^{(1)}, Z^{(1)}, SP^{(1)}, DT^{(1)} \right\rangle;$$

$$A^{(2)} = \left\langle \{EW^+\}^{(2)}, \{EW^{\pm}\}^{(2)}, \{EW^-\}^{(2)}, Z^{(2)}, SP^{(2)}, DT^{(2)} \right\rangle;$$

$$A^{(M)} = \left\langle \{EW^+\}^{(M)}, \{EW^{\pm}\}^{(M)}, \{EW^-\}^{(M)}, Z^{(M)}, SP^{(M)}, DT^{(M)} \right\rangle.$$

In various papers (see [Hernes, Nguyen 2007; Kisielnicki 2008; Sobieska-Karpińska, Hernes 2011b], different kinds of criteria of determining consensus are presented. For example, a minimal sum of distance between consensus and profile was taken into consideration (called a consensus according to criterion C_1). Such a consensus is very similar to one of the profile elements. A minimal sum of the square of distance between consensus and profile is also used as a distance measure (consensus according to criterion C_2). Such a consensus is more even as it is to an equal extent near to all profile elements. In consequence, the decision determined by consensus to an equal extent is taking all parts of the conflicts into consideration.

In the following part of the article the two-stage consensus determining algorithm will be elaborated on. In such proceedings it is possible to use criteria C_1 and C_2 at the same time depending on the system users' preferences.

Two-stage consensus determining allows eliminating decisions generated by agent programs whose knowledge state is incorrect; it is very possible that their decisions are also incorrect. By these means it eliminates the influence of an incorrect decision on the final decision, which is determined by the use of consensus methods and the presented user.

In the concept of a two-stage consensus determining [Sobieska-Karpińska, Hernes 2012], it assumes determining a consensus in the first stage (for example according to criterion C_1) on the basis of decisions generated by each agent, which is working in the system, that is, the initial set of agents (decisions profile). Next, the evaluation of the decisions of all the agents is achieved. This evaluation can be achieved for example by an evaluation agent (thus the method described in [Parsons, Klain 2004] is used) in this way; decisions most remote from consensus (by means of distance) receive the worst evaluations, and decisions nearest consensus receive the best evaluation. Next, agents whose decisions got the worst evaluations are eliminated and a set of agents (decisions profile) after evaluation is created. On the basis of these agents decisions in the second stage consensus is determined (for example according to criterion C_2) and the decision determined in this stage is presented to the user.

One can use a theorem defined in [Nguyen 2002; Sobieska-Karpińska 2011b] to determine a two-stage consensus in the multi-agent decision support system. In this algorithm, for each decision element in set E, it is checked how many times this element appeared in set EW^+, $EW^\pm$ and EW^-. If the element appeared in one of the sets more times than half of the number of all agents, it belongs to this set in theconsensus. If it appeared in the given set as often as half of the number of all agents or less often than half of the number of all agents, it does not belong to the consensus.

If a given decision element belongs to a given set in the consensus, we move to another element of set *E*. Next, we determine an ascending order of values *Z*, *SP*, *DT* for the whole profile and estimate where, between the values in these orders, the value that represents consensus has to be. The algorithm is done when all the elementary objects have been checked and the consensus for a DT value has been found.

In the next stage, an evaluation of the decisions generated by agents and elimination agents with an incorrect knowledge state is achieved. The set of agents after evaluation is created (in this set there are agents with a correct knowledge state). The decisions of these agents produce a profile, and on the basis of this profile a second-stage consensus is determined (according to criterion C_2).

To determine such a consensus first, on the basis of profile after evaluation, a consensus according to criterion C_1 is determined, and the square of distance between consensus and profile is calculated and this consensus is taken as minimal.

Next, for each element of set E, it is checked if it is to appear in a given set in the consensus. If it is, then it is eliminated from this set and the square of distance is calculated. If this square of distance is greater than the previous then go to the next set, if it is smaller, then it is taken as consensus and the distance between this consensus and profile as minimal.

If the element does not appear in a given set in the consensus, then it is calculated how many times it appears in this set in decisions of profile. If it does not at any time, then go to the next set; if it appears once or more , then it is additional to this set (if necessary it is eliminated with the other set) and it is checked if the distance between the new consensus and profile is lesser than the previous. If it is not, then the previous consensus is taken as the best, if it is, then the new consensus is taken as the best and the distance between this consensus and profile as minimal. After checking each of the sets it goes to the next element of set E.

If each of the elements of set E will be checked then it is taken that the consensus of set $EW^{+}, EW^{\pm}, EW^{-}$ is determined, and a consensus of Z, SP and DT will next be determined. After determining this consensus the algorithm is finished and the received consensus is a double-stage consensus. This algorithm is defined as follows:

Data: Profile $A = \{A^{(1)}, A^{(2)}, ... A^{(M)}\}$ consist of M knowledge structure of agent.
Result: Consensus $CON = \langle CON_{+}, CON_{\pm}, CON_{-}, CON_{DT} \rangle$ consideration for A.
BEGIN
Step 1: $CON_{+} = CON_{\pm} = CON_{-} = \varnothing, CON_{Z} = 0, CON_{SP} = 0, CON_{DT} = 0.$
Step 2: $j := 1$.
Step 3: $i := +$.
Step 4: If $t_i(j) > M/2$, then $CON_i := CON_i \cup \{e_j\}$. Go to: Step 6.
Step 5: If $i = +$, then $i := \pm$. If $i = \pm$, then $i := -$. If $i = -$, then go to: Step 6.
Go to: Step 4.
Step 6: If $j < Y$, then $j := j + 1$. Go to: Step 3.
If $j \geq Y$, then go to: Step 7.
Step 7: $i := Z$.
Step 8: Calculate $pr(i)$.
Step 9: $k_i^1 = (M+1)/2$, $k_i^2 = (M+2)/2$.
Step 10: $k_i^1 \leq CON_i \leq k_i^2$.
Step 7: $i := SP$.

Step 8: Calculate $pr(i)$.

Step 9: $k_i^1 = (M+1)/2$, $k_i^2 = (M+2)/2$.

Step 10: $k_i^1 \leq CON_i \leq k_i^2$.

Step 7: $i: = DT$.

Step 8: Calculate $pr(i)$.

Step 9: $k_i^1 = (M+1)/2$, $k_i^2 = (M+2)/2$.

Step 10: $k_i^1 \leq CON_i \leq k_i^2$.

Step 11: Evaluation of the decision and elimination of agents which have incorrect knowledge state**.**

Step 12: Calculate a profile
$B = \{B^{(1)}, B^{(2)}, ... B^{(N)}\}$ consist of N decisions.

Step 13: Let CON is a consensus according to criterion C_1.

Step 14: $CON_Z = \frac{1}{N}\sum_{i=1}^{N} Z^i$.

Step 15: $CON_{SP} = \frac{1}{N}\sum_{i=1}^{N} SP^i$.

Step 16: $CON_{DT} = \frac{1}{N}\sum_{i=1}^{N} DT^i$ let

$d := \sum_{i=1}^{N}\left[\Psi\left(CON, A^{(i)}\right)\right]^2$ and $j: = 1$.

Step 17: If $e_j \in CON_+$, then CON': = $\left\langle CON_+ \backslash \{e_j\}, CON_{\pm}, CON_-, CON_Z, CON_{SP}, CON_{DT}\right\rangle$.

Go to: Step 20.

If $e_j \notin CON_+$, then go to: Step 18.

Step 18: If $t_+(j) = 0$, then go to: Step 21.

If $t_+(j) > 0$, then go to: Step 19.

Step 19: If $e_j \cap CON \neq \varnothing$ and $e_j \in CON_{\pm}$ or $e_j \in CON_{\pm}$, then

CON': = $\left\langle CON_+ \cup \{e_j\}, CON_{\pm} \backslash \{e_j\}, CON_- \backslash \{e_j\}, CON_Z, CON_{SP}, CON_{DT}\right\rangle$.

If $e_j \cap CON = \varnothing$, then

CON': = $\left\langle CON_+ \cup \{e_j\}, CON_{\pm}, CON_-, CON_Z, CON_{SP}, CON_{DT}\right\rangle$.

Go to: Step 20.

Step 20: If $\sum_{i=1}^{N}\left[\Psi\left(CON', A^{(i)}\right)\right]^2 < d$, then d: = $\sum_{i=1}^{N}\left[\Psi\left(CON', A^{(i)}\right)\right]^2$ and CON: = CON'.

Go to: Step 21.

Step 21: If $e_j \in CON_{\pm}$, then CON': = $\left\langle CON_{+}, CON_{\pm} \setminus \{e_j\}, CON_{-}, CON_{Z}, CON_{SP}, CON_{DT} \right\rangle$.

Go to: Step 24.

If $e_j \notin CON_{\pm}$, then go to: Step 22.

Step 22: If $t_{\pm}(j) = 0$, then go to: Step 25.

If $t_{\pm}(j) > 0$, then go to: Step 23.

Step 23: If $e_j \cap CON \neq \varnothing$ and $e_j \in CON_{+}$ or $e_j \in CON_{-}$,then

CON': = $\left\langle CON_{+} \setminus \{e_j\}, CON_{\pm} \cup \{e_j\}, CON_{-} \setminus \{e_j\}, CON_{Z}, CON_{SP}, CON_{DT} \right\rangle$.

If $e_j \cap CON = \varnothing$, then

CON': = $\left\langle CON_{+}, CON_{\pm} \cup \{e_j\}, CON_{-}, CON_{Z}, CON_{SP}, CON_{DT} \right\rangle$.

Go to: Step 24.

Step 24: If $\sum_{i=1}^{N} \left[\Psi\left(CON', A^{(i)}\right)\right]^2 < d$, then d: = $\sum_{i=1}^{N} \left[\Psi\left(CON', A^{(i)}\right)\right]^2$ and CON: = CON' .

Go to: Step 25.

Step 25: If $e_j \in CON_{-}$, then CON': = $\left\langle CON_{+}, CON_{\pm}, CON_{-} \setminus \setminus \{e_j\}, CON_{Z}, CON_{SP}, CON_{DT} \right\rangle$.

Go to: Step 28.

If $e_j \notin CON_{-}$, then go to: Step 14.

Step 26: If $t_{-}(j) = 0$, then go to: Step 29.

If $t_{-}(j) > 0$, then go to: Step 27.

Step 27: If $e_j \cap CON \neq \varnothing$ and $e_j \in CON_{+}$ or $e_j \in CON_{\pm}$, then

CON': = $\left\langle CON_{+} \setminus \{e_j\}, CON_{\pm} \setminus \{e_j\}, CON_{-} \cup \{e_j\}, CON_{Z}, CON_{SP}, CON_{DT} \right\rangle$.

If $e_j \cap CON = \varnothing$, then

CON': = $\left\langle CON_{+}, CON_{\pm}, CON_{-} \cup \{e_j\}, CON_{Z}, CON_{SP}, CON_{DT} \right\rangle$.

Go to: Step 28.

Step 28: If $\sum_{i=1}^{N} \left[\Psi\left(CON', A^{(i)}\right)\right]^2 < d$, then d: = $\sum_{i=1}^{N} \left[\Psi\left(CON', A^{(i)}\right)\right]^2$ and CON: = CON'.

Go to: Step 29.

Step 29: If $j < Y$, then j: $= j + 1$. Go to: Step 14.

If $j \geq Y$, then END.

END.

The calculation complexity of this algorithm is $O(N^2M) + O(3NM)$. Let us notice that in the second stage, the determining consensus according to criterion C_2 is a NP-complete problem; therefore, the above algorithm is a heuristic algorithm.

The presented algorithm of two-stage consensus determining allows coordinating a decision presented by the system to the user, taking into consideration the improving knowledge of agents which function in the system. The elaborated algorithm can be implemented in any multi-agent decision support system under the condition of representation of the agents knowledge by the structure described in this article. The algorithm is run automatically after generating a proposal of decisions by agents.

Using this algorithm in a decision support system allows a more effective supporting decision taking process, because several results are taken into consideration. In multi-agent decision support systems, these results (decisions) are generated by different agents. Different kinds of decision support methods are implemented in these agents. Of course, the decision maker can choose himself or herself a decision presented by agents, but it is a time-consuming process. Thus, using consensus methods considerably shortens the time of taking decisions and, in consequence, contributes to a better organization functioning.

4. Conclusion

Improving agents' knowledge is an essential element of multi-agent decision support systems' operation. One of the aspects of improving knowledge is solving knowledge conflicts within almost every system of that type. Designers of multi-agent systems should bear in mind that methods of recognition, classification and solving conflicts should already be taken into account in the design phase. It may be extremely difficult to include them once the system has been implemented, due to changes in the code of agent programs. A proper solution of the conflicts is essential because it guarantees that the system will suggest proper decisions. If the mentioned aspects are disregarded in the system, then the user (decision-makers) may have problems with taking right and quick decisions because the system may suggest wrong decisions, or several decisions, which may result in the decision-maker's wondering which one to chose.

Solving knowledge conflicts is possible also thanks to the use of the consensus methods and the multi-stage consensus determining process, including the two-stage algorithm described in the article, enables more effective improvement of knowledge. Such an approach enables providing users with a decision which has been selected on the basis of decisions generated by agents possessing adequate knowledge. Agents whose level of knowledge is inappropriate at a given stage are not taken into account in the final consensus; however, they do have a chance to improve their knowledge, and consequently may be taken into consideration while determining a consensus in selecting further decisions. Thanks to such an approach, the risk that a decision presented to a user is faulty is decreased, and decision-makers gain greater confidence that a decision suggested by a system using proper knowledge of agents is effective. Sobieska-Karpińska, Hernes [2011a] points out that the use of consensus methods to solve knowledge conflicts, among other things, allows shortening the period of time necessary to take a decision and limit the risk associated with this process. The use of a two-stage algorithm for determining consensus additionally leads to the increased effectiveness of decision taking since solutions generated by agents of an inadequate level of knowledge are not taken into account.

It is not certain that, when faced with a choice of agent-generated decisions, the user would choose a satisfactory solution. The use of consensus methods assures consideration of all possible solutions, thereby increasing the probability of making the right decision.

Thanks to the multi-stage consensus setup, it is possible to "raise the bar" when it comes to increasing the usefulness of the process. As a consequence, various organizations may benefit financially or function and compete better on the market.

Further research purpose relating to improving agents knowledge with the use of multi-stage consensus determining shall depend on, for example, taking into consideration the greatest number of stages consensus determining and researching its influence on effective decisions presented to the user, elaborating the mechanism of self-evaluation by the agent of its knowledge state or appointing a different (than distance) criteria of evaluation of an agent's knowledge state. It is necessary also to verify this algorithm in practice.

References

Badica C., Ganzha M., Gawinecki M., Kobzdej P., Paprzycki M., Towards trust management in an agent-based e-commerce system – initial consideration, [in:] A. Zgrzywa (Ed.), *Multimedia and Network Information Systems*, MISSI 2006, Oficyna Wydawnicza PWr., Wrocław 2006.

De Long D., Seemann P., Confronting conceptual confusion and conflict in knowledge management, *Organizational Dynamics* 2000, Vol. 29(1), pp. 33–44.

Dyk P., Lenar M., Applying negotiation methods to resolve conflicts in multi-agent environments, [in:] A. Zgrzywa (Ed.), *Multimedia and Network Information Systems*, MISSI 2006, Oficyna Wydawnicza PWr., Wrocław 2006.

Ferber J., *Multi-Agent Systems*, Addison-Wesley Longman, 1999.

Hernes M., Nguyen N.T., Deriving consensus for hierarchical incomplete ordered partitions and coverings, *Journal of Universal Computer Science* 2007, Vol. 13(2), pp. 317–328.

Kisielnicki J., *MIS systemy informatyczne zarządzania*, Wydawnictwo Placet, Warszawa 2008.

Korczak J., Lipiński P., Agents systems in capital market decision support, [in:] S. Stanek, H. Sroka, M. Paprzycki, M. Ganzha (Eds.), *Evolution Multiagent Information Systems in Social-Economic Environment*, Placet Press, Warszawa 2008.

Nguyen N.T., *Metody wyboru consensusu i ich zastosowanie w rozwiązywaniu konfliktów w systemach rozproszonych*, Oficyna Wydawnicza PWr., Wrocław 2002.

Nissen M.E., Kamel M. N., Sengupta K.C., *Toward Integrating Knowledge Management, Processes and Systems*, A Position Paper, Proceedings of the AAAI Symposium on Bringing Knowledge to Business Processes, Stanford 2008.

Pastuszak Z., *Implementacja zaawansowanych rozwiązań biznesu elektronicznego w przedsiębiorstwie*, Wydawnictwo Placet, Warszawa 2007.
Parsons S., Klain M., *Towards Robust Multi-Agent System: Handling Communication Exceptions in Double Auction*, The 2004 Conference on Autonomous Agents and Multi-Agent Systems, Valencia 2004.
Rahman S.A., Yadav D., Agerwal P., Bisth P.S., Multi-agent knowledge management architecture, *Journal of Software Engineering and Application* 2012, Vol. 5.
Sobieska-Karpińska J., Hernes M., Metody reprezentacji wiedzy agentów w multiagenckich systemach wspomagających podejmowanie decyzji finansowych, [in:] A. Nowicki (Ed.), *Informatyka Ekonomiczna. Wybrane zagadnienia*, Wydawnictwo Akademii Ekonomicznej, Wrocław 2006.
Sobieska-Karpińska J., Hernes M., Value of information and distributed decision support system, [in:] M. Pańkowska (Ed.), *Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology*, IGI Global, USA, 2009.
Sobieska-Karpińska J., Hernes M., Using consensus methods in multiagents environment of enterprise monitoring system, [in:] P. Jałowiecki, P. Łukasiewicz, A. Orłowski, (Eds.), *Information Systems in Management VIII*, Wydawnictwo SGGW w Warszawie, Wuls Press, Warszawa 2011a.
Sobieska-Karpińska J., Hernes M., Determining consensus in distributed computer decision support system, [in:] J. Dziechciarz (Ed.), *Ekonometria. Zastosowania Metod Ilościowych*, No. 31, Wydawnictwo UE we Wrocławiu, Wrocław 2011b.
Sobieska-Karpińska J., Hernes M., Consensus determining algorithm in multi-agent decision support system with taking into consideration improving agent's knowledge, [in:] M. Ganzha, L. Maciaszek, M. Paprzycki (Eds.), *Proceedings of the Federated Conference on Computer Science and Information Systems FedCSIS 2012*, Polskie Towarzystwo Informatyczne, IEEE Computer Society Press, Warsaw, Los Alamitos, CA 2012, pp. 1035–1040.
Soto J.B., Vizcaino A., Portilo-Rodriguez J, Piattini M., An agent system to manage knowledge in CoPs, [in:] N. Karacapilidis (Ed.), *Web-Based Learning Solutions for Communities of Practise: Developing Virtual Environments for Social and Pedagogical Advancement*, Hershey, New York 2009.
Winikoff M., Intelligent agents: An industrial strength platform, [in:] R.H. Bordini, M. Dastani, J. Dix, A. El Fallah Seghrouchni (Eds.), *Multi-Agent Programming: Languages, Platforms and Applications*, Springer, Berlin 2005.

ROZWIĄZYWANIE KONFLIKTÓW WIEDZY W WIELOAGENTOWYM SYSTEMIE WSPOMAGANIA DECYZJI Z WYKORZYSTANIEM WIELOETAPOWEGO WYZNACZANIA CONSENSUSU

Streszczenie: W artykule przedstawiono wykorzystanie algorytmu wieloetapowego wyznaczania consensusu w celu rozwiązywania konfliktów wiedzy w wieloagentowym systemie wspomagania decyzji. W pierwszej części artykułu opisano problem konfliktów wiedzy pomiędzy agentami, strukturę decyzji, profil i kryteria wyznaczania consensusu. W dalszej części został opracowany algorytm dwuetapowego wyznaczania consensusu. Algorytm ten umożliwia, między innymi, skrócenie czasu niezbędnego do podjęcia decyzji, ograniczenie ryzyka związanego z tym procesem oraz doskonalenie wiedzy agentów, co w konsekwencji może prowadzić do zwiększenia efektywności procesu podejmowania decyzji.

Słowa kluczowe: systemy wieloagentowe, systemy wspomagania decyzji, konflikty wiedzy, algorytmy wyznaczania consensusu.