http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (21): 6235-6240, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6235.6240

Decision Algorithm for Multi-Agent Intelligent Decision Support System based on Blackboard

Sun Yongyong
School of Computer Science and Engineering, Xi'an Technological University, Xi'an, China

Abstract: Intelligent Decision Support System is a human-computer interaction information system and it can assist decision-maker to make right decision. In this study, Multi-Agent is used in Intelligent Decision Support System. In order to improve the traditional blackboard, blackboard control agent is proposed. It can decompose task and formal description of task decomposition and dependency between tasks are described. Furthermore, an effective attribute alternation algorithm is designed use of decision tree and rough set. The principles of the simplest decision tree is offered. The method is applicable for cooperative decision among agents and can help to complete complicated works in Multi-Agent System. Multi-Agent Intelligent Decision Support System based on can be used to solve large-scale complex problem.

Key words: Intelligent decision support system, multi-agent, blackboard control agent, decision tree, rough set

INTRODUCTION

Now-a-days, most work was done with the occurrence of decision-making process, whether it is good or not will influence the successful completion of tasks. A networking, intellectualized, integrated and coordinating human-machine intelligent decision support system can be achieved by agent technology (Boyuan *et al*., 2011; Zhen and Tianni, 2002). Agent is a procedure that can replace the user to carry out a specific task independently and automatically. It can support the decision making and problem solving in various stage and also can enhance the functions of traditional decision support systems. Multi-Agent system is a loosely coupled network and has good dexterity and capability(Shan and Sihai, 2008) and it is used to create applications in a variety of areas and it can solve large-scale complex problems through mutual scheduling and mutual cooperation among agents(Jin and Jie, 2012; Arokhlo *et al*., 2011).

Blackboard is a highly structurized problem-solving model for timely problem solving (Jin and Zhe, 2012). As an efficient knowledge processing method of multi-knowledge-source, the blackboard is widely used in solving knowledge-based systems and it has become one of the common patterns that used to establish knowledge processing systems. In fact, each knowledge source is equivalent to a problem domain expert and many experts use the way of cooperation and interaction to solve problems. Blackboard system can be used as one kind of Multi-Agent systems to set up model and implement (Hongguo and Hongguo, 2010; Ernuo and Youqun, 2011).

In this study, based on the traditional blackboard system, the black control agent is designed,and it used in Multi-Agent Intelligent Decision Support System In this system, each agent can use the blackboard to exchange the information, data and knowledge adequately, visits the blackboard at any time and queries the contents of release, then extracts the required work information in order to complete their task. Moreover, the black control agent can decompose task. A task can be divided into several sub-tasks. Each decision agent can complete one sub-tasks, then, they can work together to solve a given issue. For the decision of agent, the traditional method of decision tree production is achieved by great amount of entropy calculations. Entropy is done by Rough Set which is incomplete. And this study provides principles of the simplest decision tree and the general production algorithm as well. The agents can work together and achieve cooperative decision.

The rest of the study is organized as follows. In Section 2, the model of blackboard control agent is proposed. In Section 3, mechanism of task decomposition on black control agent is described. In Section 4, the simplest decision tree based on rough algorithm is introduced. In Section 5, the formation of certain decision tree is discussed. Finally, a conclusion is provided in Section 6.

BLACKBOARD CONTROL AGENT

In Multi-Agent Intelligent Decision Support System based on blackboard, the combination of different function agents and decision agents can be regarded as various knowledge sources. And these knowledge sources interact through the blackboard which acts as an integrated database that stores data, transmits information and processes methods. The blackboard is the global workspace in the system. The blackboard model makes use of blackboard control agent to improve the traditional blackboard. The blackboard control agent is the brain of system that controls the blackboard directly and the other agents indirectly in the system. It divides the problem into appropriate sub-problems in light of the self-knowledge and the principle of collaboration which are assigned to the corresponding data plane in blackboard and the blackboard control agent administers the every data plane as a whole, then, eliminates conflicts between different decision agent. The blackboard control agent should be designed to become an initiative control agent in implementation of system, its behavior completes under the support of the internal model, ability and belief, initiative agent has the perception, intentions, beliefs, abilities and learning. The blackboard control agent determines the type of problems according to users' requirements and uses the knowledge base and model base to decompose the problem, then, sends data and control information. At the same time, the blackboard control agent assigns the sub-problems to the appropriate decision agent. Until the data and results are written in the blackboard that are sent from decision agent that is, the contents of every data plane on the blackboard are changed, the blackboard control agent eliminates conflicts between different decision agents, coordinates every decision agent and feedbacks the synthesis of results to users. The structure is shown in Fig. 1.

The blackboard control agent has two main sub-functions: Problem decomposition and conflict resolution, it can use an eight-tuple to express:

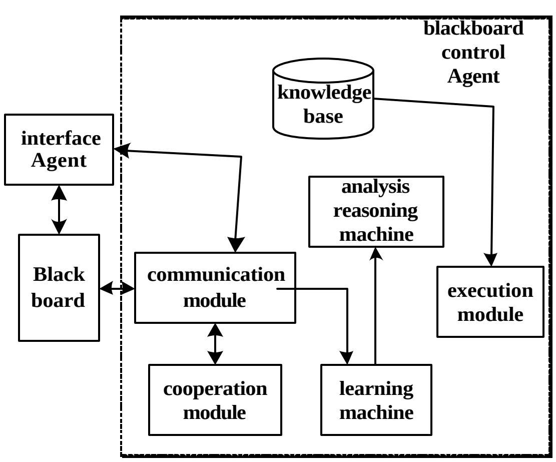

Fig. 1: Model of blackboard control agent

$$CA = (A, St, E, Ta, Ka, Mt, Ks, Bout)$$

where, A is the set of agent identification, St is the agent's internal state, E is the external environment that agent is faced with and the internal state and external environment compose the base of the reasoning and behavior on agent, Ka is the knowledge about the other agent's, usually, it is the partial function of ($A\times Ta$) part, Ta is the set of tasks identification, it represents the agent scheme, Ka and Ta describe how the blackboard control agent decomposes the decision problems. Mt is the message passing mechanism of Agent, Ks is the knowledge systems of blackboard control agent, including methods of task decomposition and conflict resolution about objectives, intentions and reasoning mechanisms among the different agents. Bout is behavior that the blackboard control agent imposed on other agent.

MECHANISM OF TASK DECOMPOSITION ON BLACKBOARD CONTROL AGENT

Formal description of task decomposition: The main function of decomposition task is to divide task into highly parallel sub-tasks as much as possible and decides when and which decision agent execute them. Decomposition task should consider how the task is represented and the approach of breakdown task and also consider organization and resources that execute task and relationship of sub-tasks. The following issues formal description of the decomposition task, decomposition task is defined as follows:

$$\langle K, A, E, I, G\rangle$$

In which K is the knowledge set on task, including the initial conditions of task, objectives and intermediate results. A is the operations set, operations receive the corresponding input and get the corresponding results through the calculations and given tasks will be accomplished by these operations. E is the execution units set, in which each actor has different abilities and can use certain cost to complete the operation in set of operations and if the cost is infinite, it means that the actor can not complete this operation. I is the initial condition set, it is the subset of K, including the knowledge when the task has been produced. G is the target set, it is also a subset of K, including necessary knowledge that complete tasks. I and G define the tasks to be accomplished, K, A and E define the environment that task depends on.Then, the feasible optimal task decomposition can be defined when the following conditons are achieved:

- All operations get necessary input information before they were executed
- All knowledge in *G* will be obtained
- The cost of communication and execution is minimum

In order to realize the third condition, we define a cost function ExecFun and a communication cost function CommuFun, where ExecFun: A, E→R, CommuFun: E, E→R (R is real set). The operations are represented by A_i, (i, j = 1, 2,..., N). The execution unit is represented by E_l, (l = 1, 2,..., L). The knowledge is represented by K_q, (q = 1, 2,..., Q). And binary vectors M_{iq}, D_{jq}, Z_{ik}, X_i, V_i, Y_{ij}, W_{ik} are defined as following:

Def M_{jq} = 1 if q is included in input information about operation j
Def D_{jq} = 1 if q is included in output information about operation j
Def Z_{ik} = 1 if operation i is completed by execution unit k
Def X_i = 1 if operation i is executed during the task is achieved
Def V_i = 1 if information i is necessary for task
Def Y_{ij} = 1 if the input information of operation j can be provided by output information of operation i
Def W_{ik} = 1 if execution unit i communicated with execution unit k

According to these definition, we can get as following:

- Each operation can be executed once that is:

$$\forall i(\sum_k Z_{ik} \le 1) \quad (1)$$

$$\forall i(\sum_k Z_{ik} = X_i) \quad (2)$$

- Output information of all operations must cover target set that is:

$$\forall i \sum_j D_{ji} X_j \ge V_i \quad (3)$$

- Each operation will be executed only input infornation exists that is:

$$\forall q \forall j(\sum_i D_{iq} Y_{ij} \ge M_{jq} X_j \quad (4)$$

- The sequence of operations must be feasible. Introduce binary variable R_{ij}, it is transitive closure of Y_{ij} that is:

$$\begin{cases} \forall i \forall j(R_{ij} \ge Y_{ij}) \\ \forall i \forall j \forall k(R_{ik} + R_{kj} \le R_{ij} + 1) \\ \forall i(R_{ii} = 0) \end{cases} \quad (5)$$

- Communication will be done only need to transfer information that is:

$$\forall i \forall j \forall k \forall l(Z_{ik} + Z_{jl} + Y_{ij} \le W_{kl} + 2) \quad (6)$$

- The cost of task completion is:

$$\sum_i \sum_l Z_{il} \text{ExecFun}(E_i, E_j) + \sum_i \sum_j W_{ij} \text{CommuFun}(E_i, E_j) \quad (7)$$

Dependency between tasks: A task can be described as:

Task = <taskid, descrip, preconds, postconds>

where, taskid is the only task identifier, descrip is description information for the task; preconds is precondition for the task, including the input, control information and the initial state of the task and so on which can be seen as the task's input; postcons is the state that the task was implemented successfully, including the output, control information and the state that has achieved, it can be seen as the task's output. We will organize task in light of level, information will be transferred at the same level of sub-tasks which accord with a certain partial order relation, because there are dependencies among tasks. For the interdependency of preconds and postcons at the same level of sub-tasks, dependencies among tasks will be generated. For the task1 and task2, the relationship between tasks is determined as follows:

- pre(task1, task2)£ºtask2 depends on task1 directly, task2 need to be achieved before task1. It can be defined as:

$$\text{Pre(task1, task2) iff task1.postcondsn} \cap \text{task2.preconds} \ne \Phi$$

- Interdepend (task 1, 2), task 1 is interdependence with task2. It can be defined as:

$$\text{Interdepend(task1, task2) iff task1.postcondsn} \cap \text{task2.preconds} \ne \Phi \text{ And task2.postcondsn} \cap \text{task1.preconds} \ne \Phi$$

Detection of dependency deadlock: Because dependency has transitivity that is Pre(task 1, 2) and Pre(task 2, 3)→Pre(task 1, 3), according to it, we can get transitive closure of dependency on a certain task(Pre?(task)) which is a task set that depend on the task directly or indirectly. If a certain task meet task⊆Pre?(task), then the set of tasks have dependency deadlock that need to decomposition tasks again.

SIMPLEST DECISION TREE BASED ON ROUGH SET ALGORITHM

The decision tree means that represent a decision set by tree structure and the decision set produces rules through the classification of the data set (Yun *et al*., 2004; Fei and Hu, 2012). For certain decision tree, each leaf node of the tree only corresponding to certain rules, the degree of confidence equals to one and it is often obtained by the coordinating decision-making table. For uncertain decision tree, some leaf nodes of the tree corresponding to the object of different decision classification, these leaf nodes are called uncertain leaf node and their corresponding rules are uncertain which the degree of confidence is less than one, they are often obtained by the non- coordinating decision-making table. We hope that select the best way in each node when construct decision tree and the instances can be classified into their classification by the way, thus, select a parameter to reflect the degree that decision classification depends on condition attribute set that is K, Ki(Yj) = max{No. of objects that attributes(i) can be classified into Yj/the number of objects about Yj}. In certain decision-making, each leaf node corresponds to a different classification of decision-making at the end of the branch that is Yj, finally, the various objects in decision-making table will be attributed to Yj and Yi∩Yj = φ(i ≠ j), therefore, the object which any leaf node corresponding to must be lower approximate of Yj, as the following:

$$\sum K(Yi) = \sum \frac{|\underline{\ }(Yi)|}{|Yi|} = \frac{|Yi|}{|Yi|} = 1 \tag{8}$$

ΣK(Yi) is equal to one, it is indication that distinguish certain decision tree. For every classification (Yi) in uncertain decision-making table, there is ΣK(Yi) less than one. If select attribute that K(Yj) is maximum as branch node, the branch must cover object as much as possible that is to be classified object decreases relatively, then, the number of branches will be decreased throughout the tree which will become the simplest tree. Each path from the root to the leaf node corresponds to a simplest decision rule. During the choice of the condition attribute set (branch node), delete attributes with noise and irrelevant attributes and reduce the number of attributes so that increase the intelligibility of the decision-making and reduce the complexity of computation. As a result, simplify the attributes at first and select minimal condition attribute set by Ki(Yj), the algorithm about decision tree as following:

Step 1: Simplify the attributes that is, delete redundant attributes

Step 2: Compute the value of Kand divide the equivalence class of every condition attribute

Step 3: If max {Ki(Yj)} is not equal to zero, select condition attribute corresponding to max{K} as root node to branch and K represents a kind of classification results, mark the value of K at the leaf node of certain decision classification. Combine with other condition attribute, respectively after attribute (i), then, constitute a new attribute set(i). If max{Ki(Yj)} is equal to one, the classification about object of Yj can be determined in light of i and branch is over. If {Ki(Yj)} is equal to zero, combine with other condition attribute, respectively after attribute (I), then, constitute a new attribute set(i) and step 1 and step2 will be repeated

Step 4: Do step 3 once again, the attribute that has been used in current branch will not be selected. If ΣK(Yi) is equal to one, a certain decision tree will be achieved and a uncertain decision tree will be achieved if ΣK(Yi) is less than one, at the same time, the degree of confidence of rule corresponding to uncertain leaf node will be computed

FORMATION OF CERTAIN DECISION TREE

As is shown in Table 1, for condition attribute a, 0 represents sunny,1 represents overcast and 2 represents

Table 1: Decision table

	Condition attributes (c)				Decision attribute (D)
U	Outlook (a)	Temperature (b)	Humidity (c)	Windy (d)	Class (e)
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

Table 2: Discrete unification

	A	B	C	D	E
1	0	0	0	0	0
2	0	0	0	1	0
3	1	0	0	0	1
4	2	1	0	0	1
5	2	2	1	0	1
6	2	2	1	1	0
7	1	2	1	1	1
8	0	1	0	0	0
9	0	2	1	0	1
10	2	1	1	0	1
11	0	1	1	1	1
12	1	1	0	1	1
13	1	0	1	0	1
14	2	1	0	1	0

Table 3: Contraction Unification

	a	c	d	e
1	0	0	0	0
2	0	0	1	0
3	1	0	0	1
4	2	0	0	1
5	2	1	0	1
6	2	1	1	0
7	1	1	1	1
8	0	0	0	0
9	0	1	0	1
10	2	1	0	1
11	0	1	1	1
12	1	0	1	1
13	1	1	0	1
14	2	0	1	0

rain. 0,1 and 2 represent hot, mild and cool, respectively in condition attribute b. Similarly, condition attribute c, condition attribute d and decision attribute e are represented by 0 and 1. The result of discrete unification is shown in Table 2. Assuming that C = {a, b, c, d}, condition attribute b is omissible for decision attribute e because ind(C-{b}) = ind(C), then, attribute b can be deleted. The result is shown in Table 3.

$$U/a = \{E1, E2, E3\} = \{\{1, 2, 9, 8, 11\}, \{3, 7, 12, 13\}, \{4, 5, 6, 10, 14\}\}$$

$$U/c = \{E1, E2\} = \{\{1, 2, 3, 4, 8, 12, 14\}, \{5, 6, 7, 9, 10, 11, 13\}\}$$

$$U/d = \{E1, E2\} = \{\{1, 3, 4, 5, 8, 9, 10, 13\}, \{2, 6, 7, 11, 12, 14\}\}$$

$$U/e = \{Y1, Y2\} = \{\{1, 2, 6, 8, 14\}, 3, 4, 5, 7, 9, 10, 11, 12, 13\}$$

There is $E2 \subset Y2$ for the set of equivalence class about a. That is, all of objects in E2 can be included in class Y2. The degree of dependence about $K_a(Y2)$ is equal to four-ninths, it is Y2 to attribute a. While $K_c = K_d = 0$, a that is corresponding to four-ninths as the first branch node and it is root node. Classification is shown in Fig. 2.

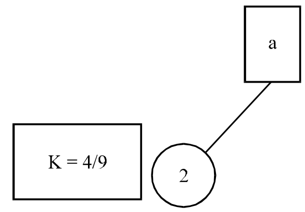

Fig. 2: Process one

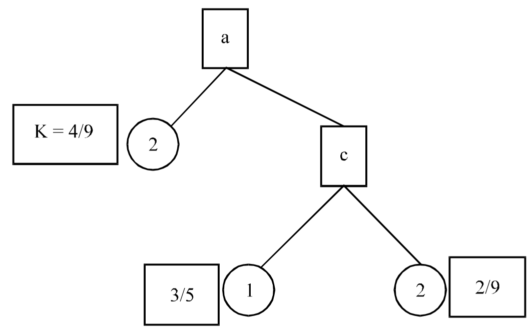

Fig. 3: Process two

$$U/ac = \{E1, E2, E3, E4, E5, E6, E7\} = \{\{1, 2, 8\}, \{3, 12\}, \{4, 14\}, \{5, 6, 10\}, \{7, 13\}, \{9, 11\}\}$$

The object of E2 and E6 can be partitioned only in light of a from above expression. It does not need to think:

$$E1 \subset Y1 \; K_{ac}(Y1) = 3/5$$
$$E7 \subset Y2 \; K_{ac}(Y2) = 2/9$$

So, c as the next branch node is shown in Fig. 3:

$$U/ad = \{E1, E2, E3, E4, E5, E6\} = \{\{1, 8, 9\}, \{2, 11\}, \{3, 13\}, \{4, 5, 10\}, \{6, 14\}, \{7, 12\}\}$$

The object of E3 and E6 can be partitioned only in light of a from above expression. It does not need to think, then:

$$E4 \subset Y2 \; K_{ad}(Y2) = 3/9$$
$$E5 \subset Y1 \; K_{ad}(Y1) = 2/5$$

Under the root node a, d should be branch node and it is shown in Fig. 4.

Node 1 represents the classification Y1 and the sum of their value of K is three-fifth plus two-fifth that is equal to one. Node 2 represents the classification Y2 and the sum of their value of K is four-ninths plus two-ninths,

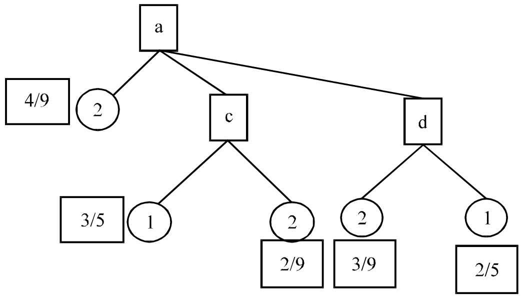

Fig. 4: Process three

then plus three-ninths that is also equal to one. The branch is over. The simplest decision tree and its set of decision rule can be achieved:

$$a_0c_0 \rightarrow e_0$$
$$a_0c_1 \rightarrow e_1$$
$$a_2d_0 \rightarrow e_1$$
$$a_2d_1 \rightarrow e_0$$

CONCLUSION

In this study, blackboard is used in Multi-Agent Intelligent Decision Support System. Agents share information through it and make use of the information in the blackboard to determine their own behavior, collaborate to solve complex problem. The task can be decomposed by blackboard control agent, each decision agent can complete a sub-task in light of its knowledge. The simplest decision tree can be achieved through decision algorithm. As the agent's own uncertainty, the effective coordination needs to be solved among agents for Multi-Agent system. Thereby establishing a unified coordination mechanism so that the agent can coordinate work effectively and improving overall system performance in practical applications, these problems still need to be further improved.

ACKNOWLEDGMENT

This study is supported by 2013 Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 2013JK1191).

REFERENCES

Arokhlo, M.Z., A. Selamat, S.Z.M. Hashim and M.H. Selamat, 2011. Multi-agent reinforcement learning for route guidance system. Int. J. Adv. Comput. Technol., 3: 224-232.

Boyuan, L., F. Weihui and X. Tianyuan, 2011. Development of decision support system. J. Syst. Simulat., 23: 241-244.

Ernuo, J. and S. Youqun, 2011. Agent cooperation mechanism research in MAS. Microcomput. Inform., 27: 125-127.

Fei, H.X. and L. Hu, 2012. Combined rough set and decision tree method for intrusion detection. Comput. Eng. Appl., 48: 124-128.

Hongguo, Z. and X. Hongguo, 2010. Intelligent decision support system for risk management of collaborative projects. Microcomput. Inform., 26: 23-24.

Jin, X. and H. Zhe, 2012. An integrated methodology for multi-agent systems design. J. Convergence Inform. Technol., 7: 428-437.

Jin, X. and L. Jie, 2012. A study of multi-agent based model for urban intelligent transport systems. Int. J. Adv. Comput. Technol., 4: 126-134.

Shan, F. and G. Sihai, 2008. A conceptual framework for the SBA-oriented intelligent decision support system. Trans. Intell. Syst., 3: 201-209.

Yun, J., L. Zhanhuai, Z. Qiang and L. Yang, 2004. New method for constructing decision tree based on rough sets theory. Comput. Appl., 24: 21-23.

Zhen, L. and N. Tianni, 2002. Research and application of an agent-based IDSS model. Comput. Eng., 28: 120-122.