Coordination Ontology for Multi Agent based Distributed Decision Making

Sudipta Acharya
Department of Information technology
National Institute of Technology
Durgapur, India
email: sudiptaacharya.2012@gmail.com

Animesh Dutta
Department of Information technology
National Institute of Technology
Durgapur, India
email:animeshnit@gmail.com

Abstract—Software agents can take decisions to perform different tasks to achieve a goal independently. When a number of agents take decisions collaboratively in distributed environment to achieve same goal, they need to coordinate with each other in order to avoid inconsistencies. Coordination is the process of managing interactions between these agents by identifying and possibly resolving the inconsistencies which can occur between their activities. In this work we propose a methodology by which number of agents can take their decisions independently and collaboratively by maintaining coordination between them. We propose a concept of coordination ontology for distributed decision support system. It consists of set of rules in order to coordinate actions of different agents and make a decision to achieve a goal successfully.

Keywords: Multi agent system, Expert agents, Coordination ontology, Distributed decision support system.

I. INTRODUCTION

Discussion is the most feasible thing to solve a complex problem. In human society discussion mean where two or more persons discuss to each other i.e. exchange their thoughts, views, support or oppose one another’s opinions regarding to a problem. They coordinate with each other to take a decision collaboratively. To coordinate with each other they abide by some rules based on basic human intelligence to overcome different preventive situation to take final decision successfully. But sometime human intervention in discussion to solve a problem is not possible. In that situation the role of software agent comes. For example in Medical system when a patient comes to hospital with a critical disease then if the physician is not confident about diagnosis of the patient then medical board forms. But it is costly treatment method and in rural side hospital forming group of expert physicians is beyond of imagination. So to provide treatment like medical board to common people virtual medical board can be formed. Where a virtual discussion mechanism will be formed with no human intervention.

An agent [1], [2] is a computer system or software that can act autonomously in any environment, makes its own decisions about what activities to do, when to do, what type of information should be communicated and to whom, and how to assimilate the information received. Multi-agent systems (MAS) [1], [2] are computational systems in which two or more agents interact or work together to perform a set of tasks or to satisfy a set of goals.

Multi agent system(MAS) based decision support system [3] is a system where number of software agents take a decision of a given problem collaboratively. That is here each agent plays role of a human entity in human-based group discussion methodology. Like human based discussion method each agent communicate with each other i.e. share their opinions or decisions regarding to a given problem for its solution. This communication is based on an agreed common vocabulary with explicit semantics so that all the agents can communicate in the same terms. We define this common vocabulary as coordination ontology which contains set of rules by maintaining them agents can coordinate with each other to negotiate in a final decision successfully.

Ontology [4] is the most suitable representation of domain knowledge because concepts, relationships and their categorizations in a real world can be represented with them. With the concept of Ontology we can say that coordination ontology is the domain specific ontology which contains some rules and methodologies about how to take a ultimate decision by a number of decision maker by resolving different obstacles during group discussion.

Agents in MAS are goal oriented i.e. all agents perform collaboratively to achieve some goal. We use the term user requirement, problem and goal interchangeably in this paper. So here we represent user requirements in the form of goal graph.

Happened before(hb)relationship defines the order in which goals should be achieved i.e which goal achieving should happened before another goal achieving. The concept of goal has been used in many areas of Computer Science for quite some time. In AI, goals have been used in planning to describe desirable states of the world since the 60s. More recently, goals have been used in Software Engineering to model requirements and non-functional requirements for a software system. Formally we can define Goal Graph as G=(V, E), consisting of

- A set of nodes V={V₁, V₂,…,Vₙ} where each Vᵢ is a goal to be achieved in a system, 1 ≤ i ≤ n.
- A set of edges E. There are two types of edges, represented by → and ¬→.
- A function, subgoal : (VXV) → Bool. subgoal (Vᵢ,
There have been a significant number of contributions in the area of agent oriented decision support system. In [5] authors have discussed an agent based coordination mechanism for medical collaborative diagnosis. As authors have proposed, data agent used to fetch data from knowledge base for diagnosis. But no checking mechanism is there to check whether proposed diagnosis is correct or not, as well as here no group discussion mechanism is formalized to take a diagnosis decision correctly. Similarly in [6] authors have proposed an intelligent multi agent system, named IMASC for assisting physicians in their decision making tasks. Here also no group discussion mechanism is proposed to achieve correctness in decision. In [7] authors have proposed networked multi agent system. In networked multi agent system, the interaction structures can be shaped into the form of networks where each agent occupies a position that is determined by such agent’s relations with other agents. To avoid collisions between agents, the decision of each agent’s strategies should match its own interaction position. Here it is not shown if two agents take same strategies how problem can be solved. In [8] a trust based collaborative decision making algorithm for distributed environment(GIS) is proposed. Here to solve a problem each agent take their own decision independently. Then based on their trust on each other, total trust value for each decision is calculated and decision with maximum trust value is chosen. But here it is not shown how trust establishment for one agent over other agents is done. In [9], [10] game theory approach for distributed decision support system is discussed. But here no group discussion mechanism to take decision is proposed. In [11], [12] distributed decision making for partially observable environment is proposed. But here it is not shown how agents collaboratively take a decision to achieve a same goal. In [13], [14], [15] probabilistic reasoning in distributed decision support system is proposed. Here decision is taken by number of agents using conditional probabilistic values. Here also lack of group discussion mechanism is present.

There have been a lot of contributions in the area of agent based distributed decision support system. But most of them concern with how agents with partial knowledge about the environment solve a problem collaboratively. Here main goal is divided into number of subgoals and distributed among agents, so that every agent solves main goal partially and collectively it get solved. But there are very few contributions in the area of agent based group discussion mechanism where all agents deal with same problem or goal. For a goal or subgoal they independently take decision to solve it, instead of different agent partially solve given problem to collaboratively solve it. So here we develop a virtualization of human oriented group discussion mechanism. For human oriented group discussion mechanism coordination among different member’s opinion is done based on basic human intelligence. So here for software based group discussion mechanism we propose the concept of coordination ontology which consists of a set of rules to coordinate between different actions of agents and take final decision to solve a problem successfully.

**II. RELATED WORK**

There have been a significant number of contributions in the area of agent oriented decision support system. In [5] authors have discussed an agent based coordination mechanism for medical collaborative diagnosis. As authors have proposed, data agent used to fetch data from knowledge base for diagnosis. But no checking mechanism is there to check whether proposed diagnosis is correct or not, as well as here no group discussion mechanism is formalized to take a diagnosis decision correctly. Similarly in [6] authors have proposed an intelligent multi agent system, named IMASC for assisting physicians in their decision making tasks. Here also no group discussion mechanism is proposed to achieve correctness in decision. In [7] authors have proposed networked multi agent system. In networked multi agent system, the interaction structures can be shaped into the form of networks where each agent occupies a position that is determined by such agent’s relations with other agents. To avoid collisions between agents, the decision of each agent’s strategies should match its own interaction position. Here it is not shown if two agents take same strategies how problem can be solved. In [8] a trust based collaborative decision making algorithm for distributed environment(GIS) is proposed. Here to solve a problem each agent take their own decision independently. Then based on their trust on each other, total trust value for each decision is calculated and decision with maximum trust value is chosen. But here it is not shown how trust establishment for one agent over other agents is done. In [9], [10] game theory approach for distributed decision support system is discussed. But here no group discussion mechanism to take decision is proposed. In [11], [12] distributed decision making for partially observable environment is proposed. But here it is not shown how agents collaboratively take a decision to achieve a same goal. In [13], [14], [15] probabilistic reasoning in distributed decision support system is proposed. Here decision is taken by number of agents using conditional probabilistic values. Here also lack of group discussion mechanism is present.

**III. SCOPE OF WORK**

There have been a lot of contributions in the area of agent based distributed decision support system. But most of them concern with how agents with partial knowledge about the environment solve a problem collaboratively. Here main goal is divided into number of subgoals and distributed among agents, so that every agent solves main goal partially and collectively it get solved. But there are very few contributions in the area of agent based group discussion mechanism where all agents deal with same problem or goal. For a goal or subgoal they independently take decision to solve it, instead of different agent partially solve given problem to collaboratively solve it. So here we develop a virtualization of human oriented group discussion mechanism. For human oriented group discussion mechanism coordination among different member’s opinion is done based on basic human intelligence. So here for software based group discussion mechanism we propose the concept of coordination ontology which consists of a set of rules to coordinate between different actions of agents and take final decision to solve a problem successfully.

**IV. PROPOSED METHODOLOGY**

**A. Proposed MAS based distributed decision support system**

**Fig. 1: Architecture of the proposed Automated System**

Figure 1 represents the architecture of our proposed MAS based distributed decision support system. Let in our proposed system there is a set of intelligent software agents, $A = \{A_1, A_2, \ldots, A_n\}$ which may or may not be scattered to different physical location. Where $\forall i \in A_i$ is equivalent to one human expert specific to a domain of discussion, where $i = 1 \ldots n$. As in human based group discussion mechanism, human expert has knowledge about a domain of discussion similarly in our proposed virtual distributed group discussion mechanism each expert agent has knowledge base which represents knowledge of that expert about a domain. Now suppose a query Q from a user U comes to an expert agent $A_i$ for the solution. To take decision for solution of a problem Q, agent $A_i$ goes through a discussion procedure which is held by distributing the query Q among a number of expert agents specific to that domain. Query is represented in the form of goal graph shown in Figure 2, where $G = \{G_1, G_2, \ldots, G_p\}$, where $\forall i \in G_i$ represent an indivisible subgoal where $i = 1 \ldots p$. Every subgoal $G_i \in G$ is solved by every expert agent $A_i \in A_n$. So each subgoal has a set of alternative solution defined by $D_j = \{D_1, D_2, \ldots, D_n\}$, where $D_j$ is the decision given by the expert agent $A_j$ for subgoal $G_i$, where $i=1..n$ and $j=1..p$. Now to take the final decision among all decisions of D, coordination ontology is
used which consists of number of rules. After taking the final decision $D_j$, it is send back to all expert agents to update their knowledge base. Next again all agents take and submit their decision of next subgoal. This process isiterated until the final decision $D_p$ of last subgoal $G_p$ from set $D_p$ is taken. The last and final decision is then send to the user who wants solution of the query $Q$.

**B. Coordination Ontology for decision support system**

Coordination Ontology is a knowledge base which contains knowledge about how to coordinate between different decisions to reach in a final decision. When in a group discussion a final decision is taken among a number of alternative decisions then several consistencies may occur like two decisions may be conflicting to each other. For example for a query what is the capital of India, one agent gives answer Delhi, another agent gives answer Kolkata, then these two answers are conflicting to each other. Similarly, two decisions may be partially conflicting or partially common or one expert agent can give partial solution of a problem or can provide no solution of a problem. Coordination ontology stores a set of rules to identify these relationships between decisions and rules for resolving these constrains to reach in a final decision.

Coordination ontology is formed by three subsections,

- Class
- Instance
- Relation

![Fig. 2: Goal graph representation of user query](image)

In Figure 3 the overview of our proposed coordination ontology is given. There are 5 classes

- Agent
- Decision
- Decision relation
- Query
- Decision coordination rule classes have relationships between them and each class consists of a set of instances. Each class and their instances are described below in detail.

1) **Agents:** Our starting concept is Agent. For the purposes of coordination ontology, Agents have agent-id, which is a unique id for each agent and a trust-value. Trust establishment among number of agents is another direction of our work where agent having more trust-value indicates most knowledgeable agent than other agents. We have not discussed trust-establishment mechanism in our paper. Each agent is associated with three relations.

   - Agent SENDS query
   - Query COMES-TO agent
   - Agent TAKES decision

2) **Query:** The query is the problem for which the user wants its solution. As agents are goal oriented so query is represented in the form of goal graph. Where query is represented as main goal, and main goal is divided in number of subgoals. Subgoals are further divided until those subgoals become indivisible. So all leaf node subgoal represents indivisible or atomic subgoals required to be achieved according to the happened-before relationship between them in order to achieve main goal. Query class have following instances,

   a) **Atomic goals:** Those goals which can not be subdivided in subgoals are atomic goals. A flag ACHIEVED is associated with each atomic goal. If an atomic goal is successfully reached then this goal is called achieved.

   b) **Composite goals:** Those goals which can be further subdivided in number of subgoals is called composite goals. Composite goal can be of two type,

   Conjunctive goals If a composite goal is called conjunctive goal then the main goal will be said to be achieved if and only all leaf node indivisible atomic subgoals of that main goal are achieved.

   Disjunctive goals If a composite goal is called disjunctive goal then the main goal will be said to be achieved if at least one of the leaf level atomic subgoal of that main goal is achieved.

3) **Decision:** Decision is a solution given by an expert agent corresponding to query or goal. To take final decision of main goal successful decisions for each leaf node indivisible subgoals should be taken individually in order maintained between leaf node subgoals. The decision of last subgoal according to happened before relationship will be the final solution. This class have the following instances,

   a) **Successful:** A decision will be said to be successfully taken if among all alternative decisions given by all expert agents final decision can be taken successfully by maintaining the rules specified in coordination ontology.
b) Not available: A decision is said to be not available if for a given subgoal no expert can provide a solution.

c) Partial: A decision is said to be partial if an expert is able to give partial solution of a given subgoal.

d) Number-of-decisions: It denotes number of expert agents able to give a decision or solution of a subgoal. It can vary from 0 up to number of experts considered.

4) Decision-relation: There may exists different relations between two decisions which are the constrains to take a final decision. Those are,

a) Conflicting: Two or more decisions of a given subgoal are said to be conflicting to each other if two decisions provide two different solution for same problem(here subgoal).

b) Happened before: Two decisions have happened before relationship between them if and only if two subgoals corresponding to these two decisions have happened before relationship between them.

c) Partially common: Two or more decisions of a given subgoal are said to be partially common if two decisions give partially same solution for same problem(here subgoal).

5) Decision coordination rule: Decision coordination rule consists of a set of rules by which the inconsistencies between decisions are identified and those are resolve to take final decision successfully. The instances of this class are described in detail below,

a) Check-query: A query is said to be atomic if the goal graph of the query contains only root goal. A query is said to be composite if its goal graph consists of two or more than two leaf node atomic subgoal.

if a decision’s successful flag = TRUE then
for corresponding atomic query Achieved flag = TRUE
end if
if for a query conjunctive = TRUE Achieved flag = TRUE
if its corresponding all atomic queries have Achieved=TRUE

conjunctive query will have Achieved flag = TRUE
end if
end if

if for a query Disjunctive = TRUE then
if its corresponding any one atomic queries have Achieved=TRUE

Disjunctive query will have Achieved flag = TRUE
end if
end if

b) Check-decision: To check the status of a decision we have to check,

• whether any agent cant provide decision of a atomic goal or not i.e not available or not.

if for any goal/subgoal
{number-of-decisions = 0
}then
set not-available = TRUE
else Set not-available = FALSE
end if

• whether a decision taken by an agent is partial or full

∀ Decisions Set default value of partial= NULL
for every Decision
{if not-available = FALSE then
if {complete flag = FALSE } then
set partial = TRUE
else set partial = FALSE
end if
end if
}

• To check how many numbers of decision of a given goal/subgoal is available,

∀ subgoal/goal do
{Initially-number-of-decision = 0
for each decision Subgoal/goal number-of-decision = number-of-decision + 1
}

c) Check-decision-rlsn: To check the relationships between two decisions we have to follow rules given below,

• To check whether any two decisions of a same goal/subgoal is conflicting or not, Initially set conflicting = NULL
if for any goal/subgoal (number of decision ≠ 0) and any two decisions are different from each other then
Set conflicting = TRUE between those two decisions
else Set conflicting = FALSE between those two decisions
end if

• To check whether two decisions have happened before relationship between them or not, Initially value of happened before = FALSE
if two subgoals of a goal graph have happened-before relationship between them then
set happened-before = TRUE between those two corresponding decisions of subgoals end if

• To check whether two decisions of a same goal/subgoal is partially common or not.
Initially for each goal/subgoal
if number of decision = 0 then
set partially common = NULL
else if for every subgoal/goal (any two or more decisions have any common opinion) then
set partially-common = TRUE
else Set partially-common = FALSE
end if

d) Decision coordination mechanism: Decision coordination mechanism consists of set of rules to overcome different inconsistencies in the form of different relations between two
decisions in order to take decision successfully. Those are described in detail below.

Conflicting: If two or more decisions are conflicting in nature then to take final decision among number of decisions by number of expert agent the following rule should be maintained. We identify three cases in order to take final decision among number of conflicting decisions.

- **Case 1**
  if for any goal/subgoal number-of-decisions = n [where n is number of expert agents present in discussion] then grouping is done
  
  | if (groups are asymmetric) then decision belongs in largest group is chosen
  
  else The decision which the agent who have max-trust-value has taken is chosen end if
  end if

∀ decisions set
Conflicting = FALSE
Partially-common = FALSE
Partial = FALSE

- **Case 2**
  if for any goal/subgoal (n/2) < number-of-decisions < n then grouping is done
  
  | if (groups are asymmetric) then decision belongs in largest group is chosen
  
  else if (agent having max-trust-value was able to submit decision) then corresponding decision is chosen
  else Decision of the agent whose trust value is highest among all agents is chosen end if
  end if

∀ decisions set
Conflicting = FALSE
Partially-common = FALSE
Partial = FALSE

- **Case 3**
  if for any goal/subgoal number-of-decisions < (n/2) then new (n - number of decisions) agents are added in circle to submit decisions of that goal/subgoal, and submitting decision process is restarted, trust establishment is done and actions are taken according to case 1 and 2 end if

Partially-common and partial:

- **Case 1**: if for any goal/subgoal number-of-decisions = n and \( \exists \) decisions for which Partial = TRUE then grouping is done by following way
Case 2: if \( \text{decisions} < \text{one group} \) let the size of group is \( m \).

Category 3. Partial and partially common decisions are kept into another group.

\[
\begin{align*}
\text{end if } \\
\text{if (groups are asymmetric) then } \\
\text{largest group is chosen } \\
\text{else the group where the agent with max-trust-value belongs is chosen } \\
\text{end if } \\
\{\text{if chosen group falls under category 2 then } \\
\{\text{if (more than } m/2 \text{ partial decisions can produce one or more complete decisions) then } \\
\text{add that complete decisions into the groups which falls into category 1. } \\
\text{else new } n \text{ agents are chosen and decision making process is restarted } \\
\text{end if } \} \\
\text{else if (largest group falls under category 3) then } \\
\{\text{new } n \text{ number of agents are chosen, to whom the part of subgoal/goal in which the decision contradicts as well as not taken is send to them. } \\
\text{In new } n \text{ agents using rules of coordination ontology } \\
\text{final decision is chosen for that conflicting subproblem and return back the decision to previous group of agents, where combining two decisions final decision is made } \\
\text{else our previously defined methodologies to resolve conflict, happened-before, partially-common decisions are chosen to take final decision.} \}
\end{align*}
\]

\( \forall \) decisions set

Conflicting = FALSE

Partially-common = FALSE

Partial = FALSE

- Case 2: if (for any goal/subgoal (n/2) < number-of-decisions < n and \( \exists \) decisions for which partial = TRUE) then
  grouping is done as defined in case 1
  \{ if group is asymmetric then \\
  largest group is chosen \\
  else if the agent with max-trust-value was able to submit its decision then \\
  the group where it belongs is chosen \\
  else the group where the agent with highest trust value among other agents capable of taking decision belongs is chosen \\
  end if \}
  next depending on which category the chosen group belongs actions are taken as defined in case 1.
  end if

V. Conclusion

In this paper we propose a multi agent based distributed decision support system in which a number of agents take decisions collaboratively to solve a common problem. It is a virtualization of human oriented group discussion procedure where software agents discuss to each other through coordination ontology to resolve different inconsistencies occurred between decisions to reach in a final decision. Our future work will be to represent the knowledge base of each expert agent in the form suitable to retrieve data. We have to develop a tool by incorporating the rules defined in coordination ontology and to automate the decision making process in agent oriented fashion. We have to check our proposed system with a case study also.

References