IJICTR International Journal of Information & Communication Technology Research

ITRC

Volume 4- Number 4- September 2012(43-53)

Context Aware Mental Model Sharing in Dynamic Inaccessible Environments

Sajjad Salehi
ECE Department
University of Tehran
Tehran, Iran
salehi.sajjad@ut.ac.ir

Fattaneh Taghiyareh
ECE Department
University of Tehran
Tehran, Iran
ftaghiyar@ut.ac.ir

Mohammad Saffar
ECE Department
University of Tehran
Tehran, Iran
saffar@ut.ac.ir

Kambiz Badie
IT Research Faculty of
ICT Research Institute (ITRC)
Tehran, Iran
k_badie@itrc.ac.ir

Received: March 18, 2011- Accepted: June 24, 2012

Abstract—**One of the well-known cognitive concepts, commonly used in multi-agent systems, is the Shared Mental Model (SMM). In this paper we introduce a context aware mental model sharing strategy in a dynamic heterogeneous multi-agent environment in order to maximize collaboration via minimizing mental conflicts. This strategy uses a context aware architecture that is composed of three primary layers and a cross layer part which facilitates mental model sharing between agents. We model a complex inaccessible environment with specific dynamisms where agents must share their mental models in order to make correct decisions. Our proposed strategy is compared with other methods applying some important criteria such as shared information accuracy, communication load and performance in time constraint situations. Our findings may be interpreted as strong evidence that our method enables heterogeneous agents for a qualified teamwork as well as facilitating collective commitments.**

Keywords- shared mental model, mental model, dynamic environment, context awareness, sharing method

I. INTRODUCTION

The concept of Shared Mental Model (SMM) is well known in literature of teamwork in humans. It's defined to improve team performance producing a mutual awareness that makes each teammateable to act properly in its situation and predict state and activity of other teammates. Inspired by this theory, a similar concept has been introduced in agents' teamwork [1], [2], [3], [4], [5], [6], [7], [8]. One of the functional definitions of shared mental model that we use in this work is as follows: "An overlapping understanding among members of the team regarding their objectives, structure, process, etc." [2].SMM can be used to improve team performance by better role assignment in agents' team, improving decision making, proactive communication among teammates, etc. But the process of extracting shared mental models from local mental models is a challenging domain. In the real world, agents must operate in complex and dynamic environments that change quickly. In these environments agents have to share a

large amount of information dynamically in order to obtain shared mental model from local mental models.
It should be mentioned that the agent's mind structure isnotperpended in most of research in this area. Indeed, these works presume predefined agent's mental model has been existed and they are going to introduce a sharing algorithm to extract shared mental model. Although these works claim that using shared mental model enhances teamwork performance, the process of extracting shared mental model is not elaborated enough. Most of these algorithms have a special emphasis on the role of agent and actually are relevant to agent's role. Also a few of them have tackled the problem of conflict resolution within the entire process of knowledge transference between agents. Furthermore, the functional context of agents as a tool for decreasing amount of information transferring between agents has not been considered in this domain. Also, a considerable amount of work related to mental model is about predicting humans' mental model in human-agent team to suggest appropriate behaviors in different situations.
In order to rectify these shortcomings, we introduced a context aware architecture for agent's mental model in [9]. This architecture has three layers including "context layer", "agent layer", "mental model layer" and a cross layer called "background". This architecture is designed to facilitate mental model sharing between agents. Also, a sharing strategy was proposed that resolves harmful conflicts by applying 'semantic movement'. Semantic movement is a kind of transition from the present mental states to some new states [10]. We tried to simplify the process of extracting shared mental model via mental model projection by using context layer and then resolving harmful conflicts. Also we maintain our architecture as general as possible so that it can be applied in different domains regardless of agents' role. In this paper we want to apply this architecture and sharing strategy in a dynamic environment in order to prove the claim of good performance of our method in time constraint situations that has been expressed in [10]. According to the quality of shared informationaccuracy and volume of messages passing in introduced sharing strategy, we expect this strategy to have a good operation in dynamic environments [10].We have modified the mental model structure and sharing strategy for this environment. This environment is a new complex and dynamic version of famous wumpus world which is impossible to solve by individual agents without sharing. We compared result of our method with a set of sharing methods and results are promising.
The rest of this paper is organized as follows. In the next section a brief explanation of previous work is presented. Section 3 discusses an overview of our context aware architecture and sharing strategy that proposed in [9,10]. Next section explains the properties of experiment environment and different sharing protocols are introduced in Section 5. Section 6 discusses the experimental results and finally the conclusion and future works are presented.

II. PREVIOUS WORK

This paper covers three main domain areas. These areas include context awareness, shared mental model and dynamic environment. In this section, we are going to explain main works represented in these domains.
Works that concentrate on context can be classified in two general categories. The first category focuses on context as a basic concept and explains its properties [11], [11], [13], [14], [15], [16], [17], [18]. One of the predominant issues in this category is architecture recommendation for context aware system [11], [12], [13], [15]. These works explain elements of context aware architecture and describe the context generation by this architecture. The effectiveness of some types of explanations with the goal of increasing user's trust and acceptance are investigated in [16], [17]. The second category includes those that apply context as a feature in their works [18], [21], [22], [23],[14].Most of the works in this category consider users' context and use it to provide appropriate service for them. Also, some others regard context in their works in another applications such as event detection or better collaboration in teamwork. In this paper we apply context as a property to improve mental model sharing.
A basic concept which has a fundamental role in our work is shared mental model. Researchers in this domain concentrate on two major areas including shared mental model in teams containing only intelligent agents and shared mental model in human-agents teams.[24] Focuses on creating shared mental model between agents in the system and explains that one of its preliminaries is multi part proactive communication. [1] Considers mental models of robots and figures out elements that influence it.
CAST is a team model based on a Petri-net model. It contains information about agents' responsibilities and the team's current status. CAST is used for predicting others' information needs and proactive information sharing between agents or humans in the system. A decision making component is added to CAST by [2]. It decides when others need particular information or summarizes information before sending it to others and optimizes the proactive information sharing behavior.
[25] Introduces an appropriate description for mental model and representsa mental model ontology. [3] Proposes a role based shared mental model that is a graph. This graph represents a full-fledged team plan in whichthe role of each agent has been defined. In this work shared mental model of agents is defined at first and thereafter each agent extracts its mental model from it. But we believe that a shared mental model should be formed from local mental models while agents interact and collaborate with others. A shared mental model for improvisational agents is introduced in [4]. This model is introduced for situations where users show unpredictable behaviors. It detects the conflicts by cognitive divergence and resolves them by cognitive convergence.

In the second category, the main issue is that in teams of human and agents, intelligent agents can predict human mental models [5], [6], [17], [18]. In these teams agents can operate as a teammate or simulate the behavior of a human.

In most of the related works in mental models, they assume that the mental models are already presented and they try to introduce the appropriate algorithm for sharing them. These algorithms are dependent on the role of agents and share agents' mental models based on their roles. Also, all of these works donot consider the context as a part of mental model.

In this paper we want to apply our sharing strategy in a dynamic environment. Dynamic environments are closer to real-world problems that researchers pay more attention to them. These environmentschange when time passes therefore beliefs about them must getupdated periodically. Therefore decision-making in these environments are usually bounded to time constraints. Although there are many studies in this area, we consider some of them in this section. [26] Introduces the flux agent for dynamic wumpus world. FLUX represents a method for encoding incomplete states stated by a set of lists. Indeed this paper provides a way for using FLUX to design an intelligent agent. A method for tracking recursive agent model is introduced in [26]. Then it proposes a model sharing to optimize its method for real time environments. Intelligent pilot agents are used to simulate air combat as a real time environment. [28] Uses mobile agents for monitoring dynamic environment. It applies a divide and conquer approach by the help of remote agents. A multi-agent model that used agents for simulating ants is introduced in [29]. Cooperative behavior between agents in dynamic environments is investigated to make an order of behavior.[30] Introduces a cooperation approach so that some agents called distributed agents provide local knowledge and communicate it to the central agent called administrator. Administrator creates a global picture from this local knowledge with Growing Self Organizing Map (GSOM).

III. AN OVERVIEW OF THE ARCHITECTURE AND SHARING STRATEGY

In this section we want to review the context aware architecture and sharing strategy that are proposed in [9]. This architecture has three layers including "context layer", "agent layer" and "mental model layer". It also has a cross layer called "background" that is accessible from all three layers.

Context layer is the top layer of architecture and includes all possible contexts. We defined a context to be a set of properties which are affected while system works in this context. Indeed a criteria function is used to define context that it decides which fields of mental model are important in which context. This layer is a common layer that all agents use jointly. This layer has the flexibility of defining new context as well as editing existing context. Also this layer has shared properties so that working in a context might affect other contexts. It is used for mental model projection and selection of properties that belong to current context of system.

The second layer is agent layer. It is an intermediate layer for managing agents in the system. Indeed agent layer is used to manage the specific properties of agents; arrival of new agents; exit of existing agents and etc.Each agent has a record in this layer that is instance of agent's mental model with properties at the lowest level of abstraction. Indeed this layer contains the knowledge of agents about the teammates.

Mental model layer is the lowest level of architecture. This layer stores all of agent's properties and their values in a set of data structures called history. Since storing the whole history of the system is impossible, a window is defined to put an upper bound on the size of history. Agents can use this history to analyze the trend of transition between mental states. It can provide required data for data mining and learning algorithm in order to adjust agent's behaviors. Since the environment is stable in static version, the history of mental model presenting the trend of mental model changing is unusable. While we apply this structure in dynamic version, we need to consider this component. This history presents the mental model of agents in different times and agents need to know these mental states for appropriate decision making. Using this component is a main requirement of reaching to agreements and conflict resolution in dynamic environments.

Also our architecture has a cross layer named "Background" that is accessible from all three layers. Indeed it is a shared ontology that describes the name and meaning and relation of properties in the mental model and it has the hierarchy of abstraction levels. Also in order to switch the level of abstraction of the mental models, the data aggregatingmethods for generating higher level abstractions of mental model has been defined. We have modified this layer by replacing the hierarchical tree with conceptual graph. Also we categorize entities of conceptual graph in different level of abstraction. This level is the maximum number of step that is necessary to deduct one property from observations. This new structure has an appropriate flexibility for defining the different type of relations and considering different constraints on relations and entities. These capabilities can improve the performance of our structure in dynamic environment.Figure1 presents a sample of this structure for experiment environment introduced in Section4. In this figure, theproperties which

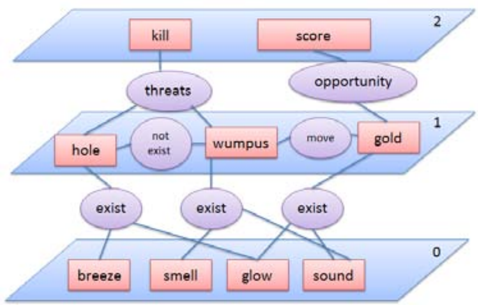


Fig.1Structure of background layer

arepresented in one plane have a same level of abstraction.

As is shown in Figure2, mental model sharing strategy has three steps including "conflict detection", "value sharing" and "conflict resolution". Criteria for conflict detection are domain dependent and must be defined by domain expert. Apredefined function is used for conflict detection that accepts properties of two agents and returns a Boolean value indicating the existence of conflict. For value sharing, we use context layer for mental model projection and

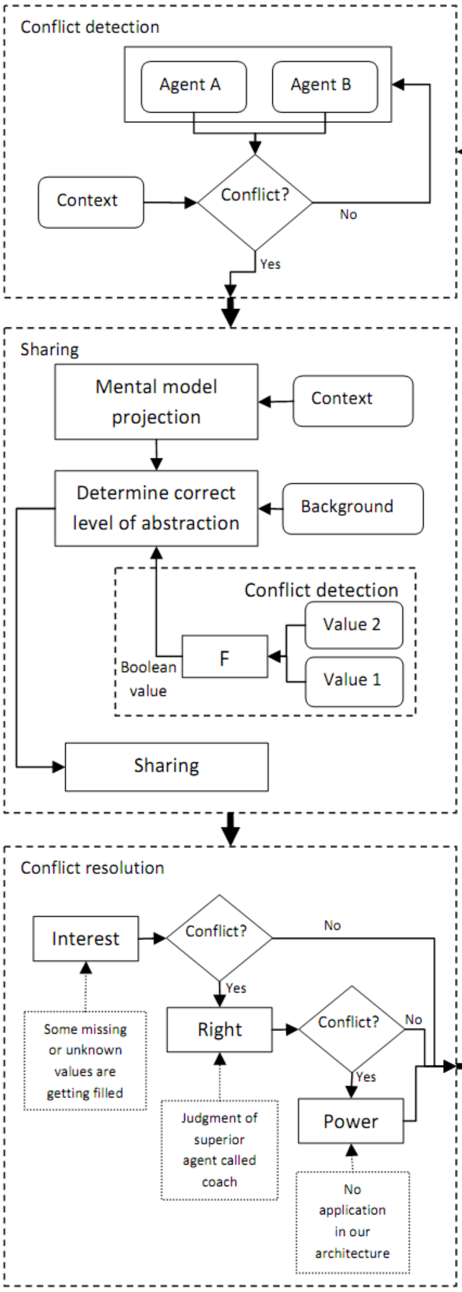


Fig.2Sharing strategy

background layer for selecting appropriate level of abstraction. In other words, context has been used to detect which properties are in active context and which are not, then properties at the highest level is investigated for conflict detection. For getting here, we must aggregate the properties at the lower level and generate properties at the highest level. It is worth mentioning each property is related to one context when it has a child field related to that context or if it is a leaf field and the domain expert assigned it to the proper context. Due to conflict detection at the higher level of abstraction is easier than lower level and information at this level has a high impact on decision making and reasoning, we start with highest level of abstraction in order to detect conflicting properties and drill down to lower level for finding the root of conflict. After identifying agents with conflicting properties and finding the root of conflict and sharing these values, agents need to resolve conflicts between their information and received information. We use a method that was inspired from the interest/right/power (I/R/P) framework introduced in [31].In the interest layer, by sharing of values, missing or unknown fields are getting filed and the agents are forced to change their minds and make new decisions. In this layer, agents explain the observations and facts that result their expressed believes. Also each agent has a grade for presenting the credit score of it in a team. To do this, a simple credit assignment method is added to this layer. This method adjusts the credit of agents based on the environment feedbacks and agents' decision. If the conflict between agents still remains after value sharing, agents move up to the "Right layer". In this layer a superior agent called "coach "which has some application dependent principles, judges between agents. In other words, if agents can not reach to an agreement in previous step, hand over their evidences to a superior agent and this agent judges between them and makes thefinal decision. Since the power layer is meaningful in human's society and its mechanism is unknown in multi agent systems, it has no application in our method.

In [10], we applied this architecture and sharing strategy in a static environment and compared them with a set of sharing methods. In this paper we compare our method with a broader set of sharing methods in static environment. Also we apply these methods in a dynamic environment in order to demonstrate the capability of our method in time constraining situations. As stated before, we make some changes in mental model structure and sharing strategy in order to tailor them for dynamic environment.

IV. EXPERIMENT ENVIRONMENT

In order to evaluate the introduced architecture and sharing strategy, we need a test bed that has several contexts. It not only must give incomplete information to agents for forcing them to share their mental models but also it should limit the time of reasoning and decision making for agents. We want to evaluate several aspect of our work. Since we need to share information and resolve conflicts between agents for making correct decisions, we must ensure the accuracy and correctness of sharing algorithm and conflict resolution framework. Also we need to investigate that our method imposes the reasonable load to system in compare with other methods.Finally we want to evaluate our method in conditions with time constraints.

Our experiment environment is a complex version of wumpus world. It is an N by N table that has three individuals including wumpus, holes and gold pieces. Each of these individuals when placed in a cell creates some special signs in adjacent cells of it. Adjacent cells of a specified cell are those which have common border with it. The number of adjacent cells based on the situation of cell can be 3, 5 or 8. While a wumpus placed in a cell make its adjacent cells to have smell and sound. If a cell contains a hole, adjacent cells of it are shiny and windy and if it containsa piece of gold, adjacent cells of it are shiny and noisy. Also agents move in this environment and sense each cell of table that they enter it. Figure 3 shows a schema of this environment.

We have three types of agents and each type has specific sensors. Each sensor senses one sign and each agent must nothas the ability of sensing all signs belonging to one individual. For example none of the agents have sensors to sense both of sound and smell or color and sound or color and wind. This causes agents to need to share their knowledge about the world map to make a clear and unambiguous world model. All of the agents have two sensors. First type can sense glows and smells while second type can sense smells and breezes and third type can sense breezes and hear sounds. Also each agent's sensor has a degree of accuracy and agentsdonot sense correctly with a pre-specified probability.

We evaluate our work in two versions of experiment environment, static and dynamic. In static version the time of experiment is divided into two steps, movement step and sharing step. In movement stepwumpie, pieces of gold and holes are distributed in map and they are stable in their primary places and donot move over time. Agents move in the map and sense four different senses that are smells, glows, breezes and sounds. Then in sharing step, agents share their mental models and they simulate original map. The duration of second step is unlimited and agents have the required time to simulate map completely.

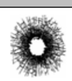
smell sound	smell sound	smell sound	glow sound	glow sound	glow sound
smell sound		smell sound	glow sound		glow sound
smell sound	smell sound	smell sound	glow sound	glow sound	glow sound
	glow breeze	glow breeze	glow breeze		
	glow breeze		glow breeaze		
	glow breeze	glow breeze	glow breeze		

Fig.3Test environment

Dynamic version of environment is formed with several rounds that each round has several steps. At the first of game wumpie, pieces of gold and holes are distributed in map. At the end of each step, wumpie move into a randomly selected adjacent cell. Also if a wumpus is placed in a cell that contains a piece of gold, it can move this piece in new cell with a configurable probability. If this wumpus move the piece of gold into new cell the probability of moving this piece again by this wumpus is decreased. After several steps, agents decide to share their mental models. This sharing step can start by demand of one agent or after certain number of steps. In this step agents must share their mental models before wumpie move. Indeed the time of this step is equal to the time interval between wumpus movement. The sharing mechanism in this step is that each agent suggests one cell for determining its status and all agents make decision about it. Also in each round duplicate cells donot be determined twice. After this step, new round begins and these rounds continue until agents can determine the location of all wumpie and pieces of gold and holes or the number of rounds exceed a predefined threshold.

V. SHARING PROTOCOLS

In order to compare our sharing strategy with other methods, we try to consider different manners that agents can share their mental models with each other. These manners are based on the strategies thatare used in movement step and sharing step by agents. We named these manners as protocols and consider 7 protocols in our test environment. Table1 presents sharing policy of these protocols in each step.

- Protocol-n-p: in this protocol agents do not share any data in movement step and collect data by sensing environment (no sharing). It also uses our architecture and sharing strategy for mental model sharing in sharing step (partial sharing). This protocol is representative of our work.
- Protocol-n-c: in this protocol agents sense the environment with no communication in movement step (no sharing) and share mental models completely in sharing step (complete sharing). In this protocol agents have complete and compatible knowledge about the map.
- Protocol-n-n: agents do not share any information in both steps and only sense environment in movement step for data gathering.
- Protocol-s-n: agents in this protocol request help of neighbor agents and receive their responses whilethey aresurprised in first step (sharing). Strangeness criterion is domain dependent and must be defined with expert. Agents do not share their mental models in second step and use polling mechanism for decision making.
- Protocol-s-p: in this protocol agents share their mental models in first step like protocol-s-n (sharing). Also they use partial sharing as a sharing mechanism in second step.
- Protocol-broadcast-n: in movement step agents send their sensed data in each cell to all of other

Table 1 Sharing protocols

	Movement		Sharing		
	No Sharing	Sharing	No Sharing	Partial Sharing	Complete Sharing
Protocol-n-c	yes	no	no	no	yes
Protocol-n-p	yes	no	no	yes	no
Protocol-s-p	no	yes *sharing with neighbors*	no	yes	no
Protocol-s-n	no	yes *sharing with neighbors*	yes	no	no
Protocol-n-n	yes	no	yes	no	no
Protocol-ant colony-n	no	yes *using ant colony*	yes	no	no
Protocol-broadcast-n	no	yes *broadcast*	yes	no	no

agents (broadcast) and do not share any data in sharing step (no sharing). Since the agents send all sensed data to all of agents in first step, each agent receives all data of other agents and do not need for datasharing in second step.

- Protocol-ant colony-n: in this protocol for moving in environment and mental model sharing in the first step, we use a method inspired by ant colony algorithm [32], [33] (ant colony). Each agent put mental state in cells of map as a pheromone. While agents move around the map use these pheromones to determine the next step of movement and to share their mental models. Agents have no sharing in second step (no sharing).

VI. RESULTS

As mentioned in Section 4, we want to evaluate several aspects of our work including algorithm accuracy, system load and performance in time limited conditions. For assessment of accuracy and correctness of sharing algorithm and conflict resolution frame work, two measurements are defined, precision and recall. Precision is the fraction of correct predicted cells over predicted cells in compare with original map. Also recall is the fraction of predicted cells over the size of map. We apply all protocols in static environment and compare predicted map with original map and calculate the precision and recall of all protocols. In order to peruse the load of system in different protocols, we define the measurement of total message length that is the sum of all bytes that exchanged between agents. Also for testing protocols in time limited conditions, we apply these protocols in dynamic environment that is explained in Section 5. We compare the number of rounds that each protocol needs to predict the correct place of different individuals for different time of round steps.

A 20 by 20 map has been implemented for calculating above measurements. The number of wumpie, pieces of gold and holes are changeable randomly from 8 to 12. Also the time of movement step in the static environment can be selected from a discrete set like {2ms, 5ms, 10ms, 100ms, 1000ms, 5000ms, 100000ms, and 50000ms}. As mentioned before dynamic environment has several round and each round has several step that last step in each round is us[illegible]or sharing. The time of round steps can be se[illegible]d from a set like {1s, 5s, 10s, 15s, 20s, and 25[illegible]Also environment is changed for 10 steps and ag[illegible] have their own strategy for movement and sh[illegible]; in these steps based on the testing protocol. After that, agents share their mental models only in one step based on their strategy in testing protocol. Both in static and dynamic environments for all protocols we ran the experiment 100 times for each member of time set that was mentioned above and calculate the average of results.

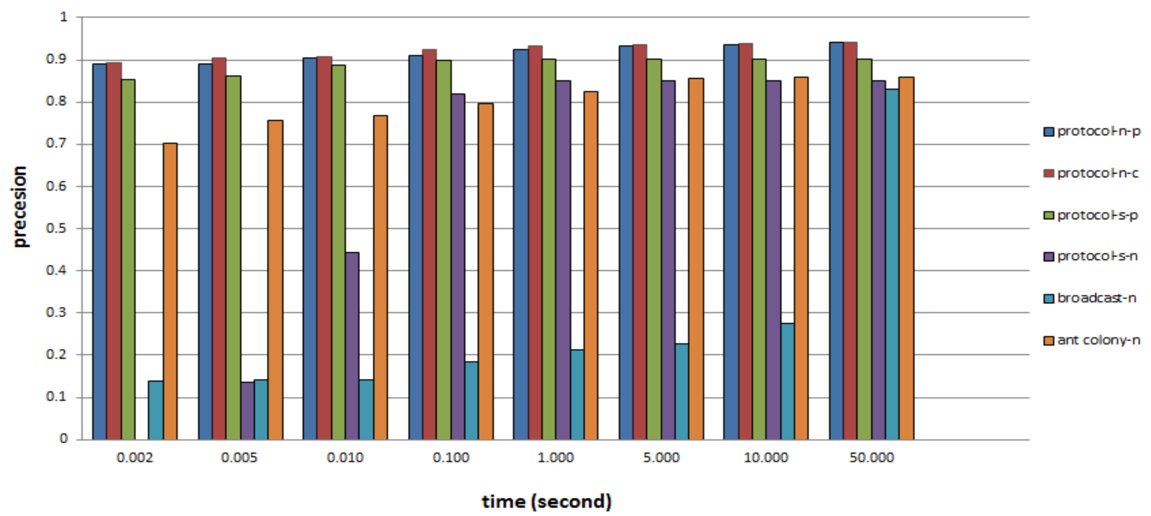


Fig.4 Precision of each scenario

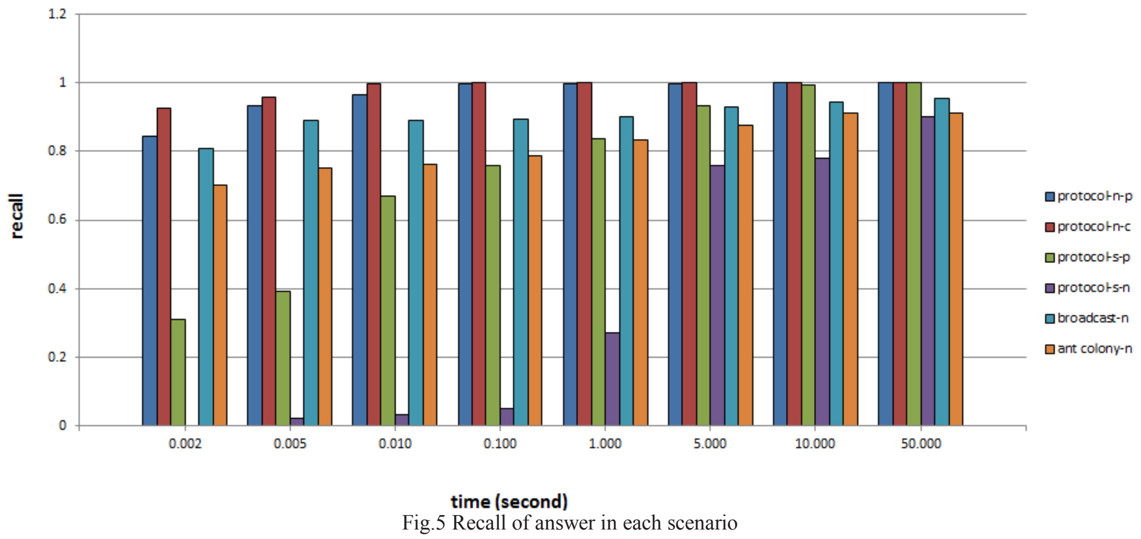


Fig.5 Recall of answer in each scenario

Figure4 presents the precision of all 7 protocols. Since agents in protocol-n-n no share any data in movement step and sharing step, they donot have sufficient data for decision making. Therefore the precision of protocol-n-n for all time steps is zero. As we expected, since protocol-n-c has the complete and compatible information about map, the best precision between all protocols belongs to it. After this protocol, our method (protocol-n-p) has the highest precision and its value is close to the precision of protocol-n-c. The cause of this closeness is that agents in our method share their mental models with conflicting agents based on the context of system. Therefore agents have necessary information for appropriate decision making. Since agents in movement steps are gathering information, they donot have sufficient knowledge for conflict detection and conflict resolution in this step. Therefore it is possible that agents have incorrect data in protocols with data sharing in movement step. For this reason, the precision of protocol-s-p is lower than the precision of protocol-n-p. Among the remaining protocols, protocol-ant colony-n has the highest precision. In this protocol agents use mental states of other agents in order to complete their own mental model and to select the next direction of movement. Since agents in this protocol have a coordinated movement in map, they visit more common cells and they can make more appropriate decisions. But agents in movement step donot have adequate information for conflict detection and conflict resolution. This incorrect data in mental models and no sharing in the second step decreased the performance of this protocol. Between two remaining protocol, protocol-s-n has the higher precision in comparison with protocol-broadcast-n. Because amount of information sharing in protocol-s-n is less than protocol-broadcast-n and agents have more time for movement and data gathering. Also less spreading conflicting data in protocol-s-n is another reason for better performance.

Figure5 compare the recall ofall protocols. As be defined, recall is the fraction of predicted cells over the number of all cells in map. Therefore, since the number of all cells is constant, the recall of each protocol is related to the predicted cells by agents. With this description, if agents have adequate information for decision making, regardless of response accuracy, the recall of protocol willrise. As expected, protocol-n-c has the highest recall and after that the recall of our work is higher than other protocols. The cause of this is that in two protocols agents have adequate information for decision making. Protocol-s-p is in the next rank. In this protocol agents share their information in movement

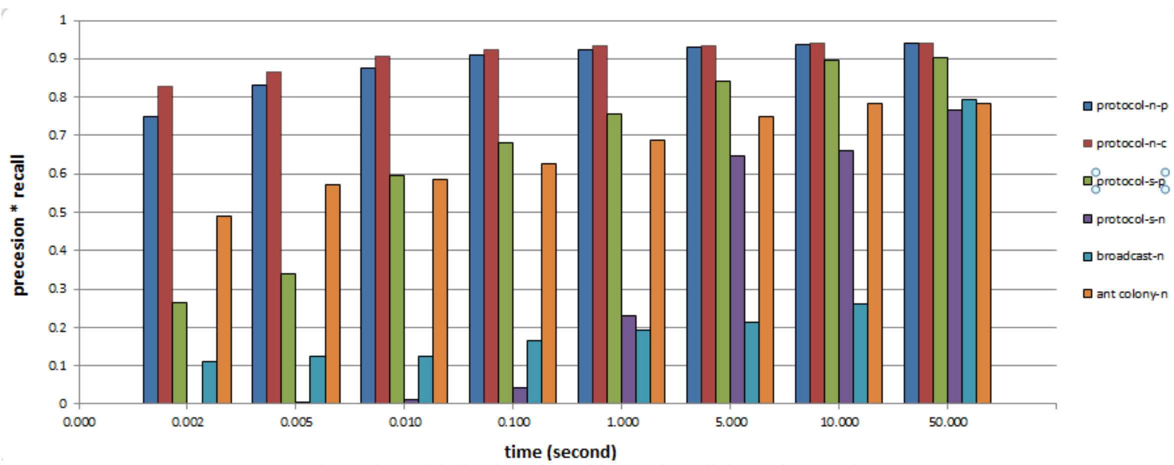


Fig.6 The multiplication of precision and recall for each scenario

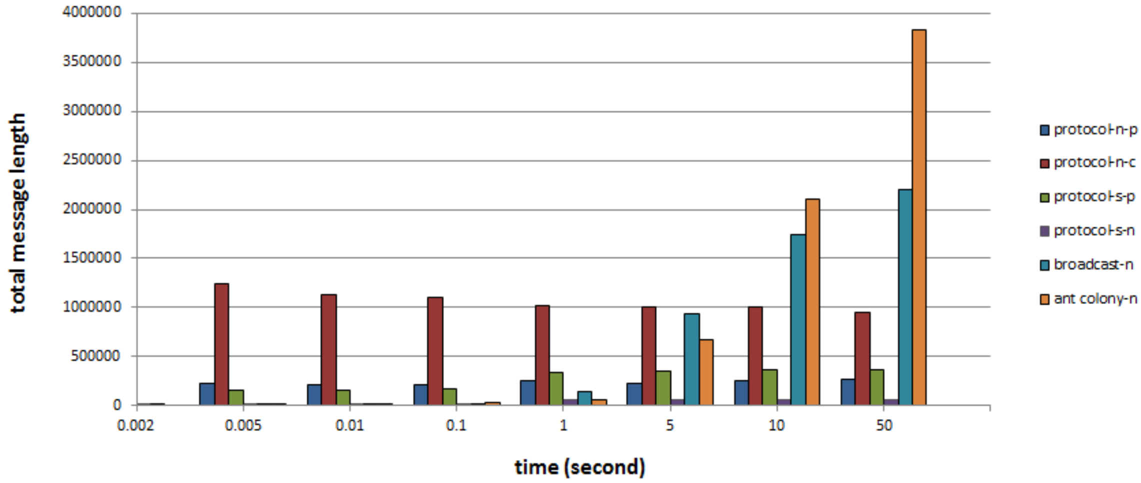


Fig.7The total number of bytes in the messages passed between agents in all scenarios

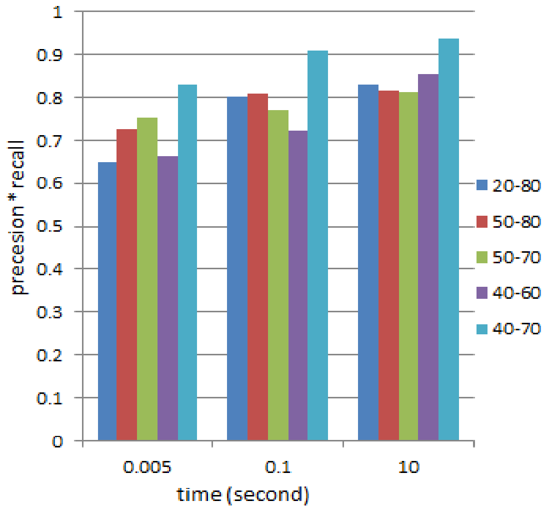
step and have limited time for data gathering. Since in data gathering agents can earn more information than message passing, the recall of this protocol is lower than protocol-n-p. Agents in protocol-broadcast-n send their information to all of other agents and therefore they have enough information for decision making. Since in this protocol agents share their information in first step, they have no adequate time for movement in environment and data gathering. As stated in protocol-ant colony-n agents have a coordinated movement and have sufficient information for decision making and for this it has an acceptable recall. In protocol-s-n agents have no adequate information in shorter time steps but in longer time steps agents have enough time for message passing and the recall of this protocol noticeably rises over time.

The multiplication of precision and recall is presented in Figure6.This figure represents the difference of all protocols properly. As we expected this measurement for protocol-n-c has the highest value and after this protocol, our work has a better performance in comparison with other protocols.

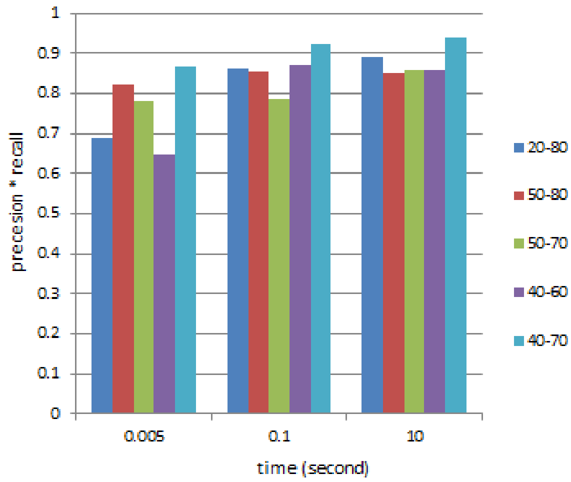
As we stated, the measurement of total message length has been defined to compute the load of system in different protocols. Figure7 presents the length of total messages gents in each protocol. It is noteworthy that we emphasis on the difference between protocol-n-c and our work. While the precision and recall of our proposed method is very close to protocol-n-c that has complete and compatible information about the map, the load of the system in our method is very lowerthatprotocol-n-c. It is clear that in protocols that have data sharing in first step, increasing the movement time will increase the total message length.

In the reasoning procedure, we need a threshold for decision making. In other words we need a threshold that if the confidence of agent about its own result is higher than it, this agent supposes that its result is true. We use two thresholds in our work, probability threshold and confidence threshold. If the confidence value of one agent is higher than probability threshold and it is lower than confidence threshold, this agent supposes its result probably is true. Also if its confidence value is higher than confidence threshold it supposes the inferred result is true. Otherwise it has no opinion about the state of desired cell. In order to detect an appropriate approximation for these thresholds we run all protocols with different thresholds. Figure8 presents the multiplication of precision and recall for all protocols with different thresholds. As an example for protocol-n-p the column that marked 40-70 has the highest value. It means that an appropriate approximation for probability threshold and confidence threshold is 40% and 70%.Also this approximation for all other protocols can be seen in Figure8.

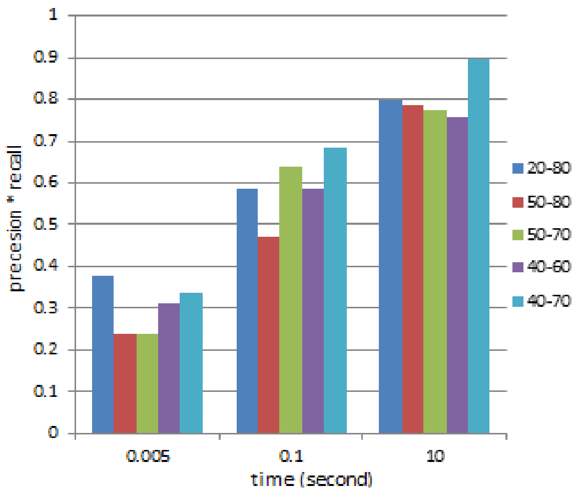
As mentioned earlier, in order to evaluate all protocols in conditions with time constraints, we apply our work and other methods in a dynamic environment. This environment is explained in Section 5 and also more details are expressed at the first of current section. Figure9 presents the number of required rounds for finishing the game. The game is finished while either all of the wumpie, pieces of gold and holes are detected by agents or the number of rounds exceeds a threshold that in our work is set to 20.Vertical axis represents number of rounds and horizontal axis represents the time step of rounds. As can be seen in figure, four protocols, including protocol-n-n, protocol-s-n, protocol-ant colony-n, protocol-broadcast-n, can not finish the game for all time steps. Protocol-n-n and protocol-s-n do not have adequate information for decision making and therefore cannot find the location of desired objects in the map. But in two other protocols the disability of agents for finishing the game is due to the massive message passing between agents that makes them so busy. Protocol-s-p does not show a good performance at first but it improves their result over time. The reason of poor performance of this protocol in primary time steps is that this protocol shares data in movement step and in short time steps agents can not gather required data properly. But in longer times they have adequate time for data gathering and therefore show a good performance. As be expected, our work has the best result. Our method shares information only with those agents that need it.



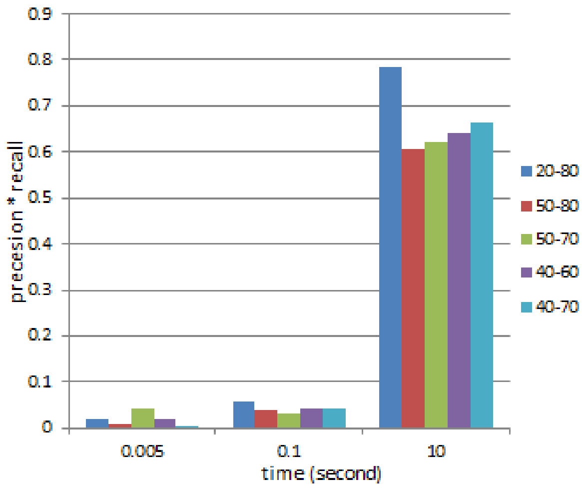
(a)



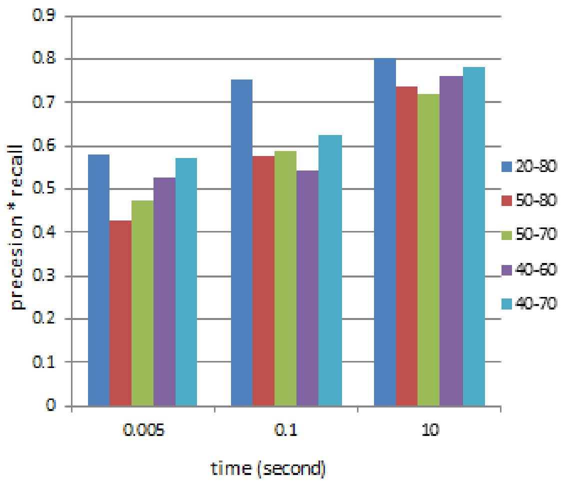
(b)



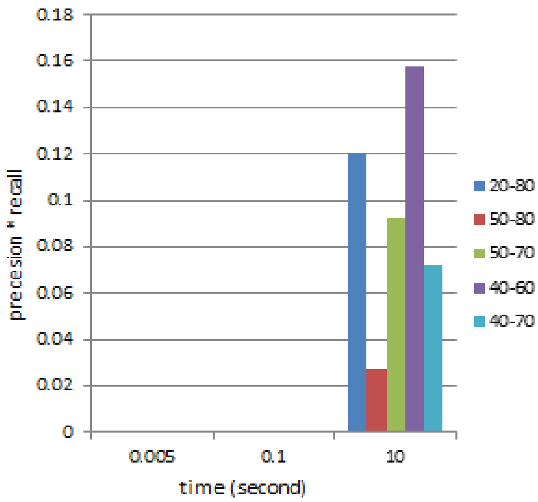
(c)



(d)



(e)



(f)

Fig.8The estimate of probability threshold and confidence threshold for **(a)**protocol-n-p**(b)**protocol-n-c**(c)**protocol-s-p **(d)** protocol-s-n **(e)** protocol-ant colony-n **(f)** protocol-broadcast-n

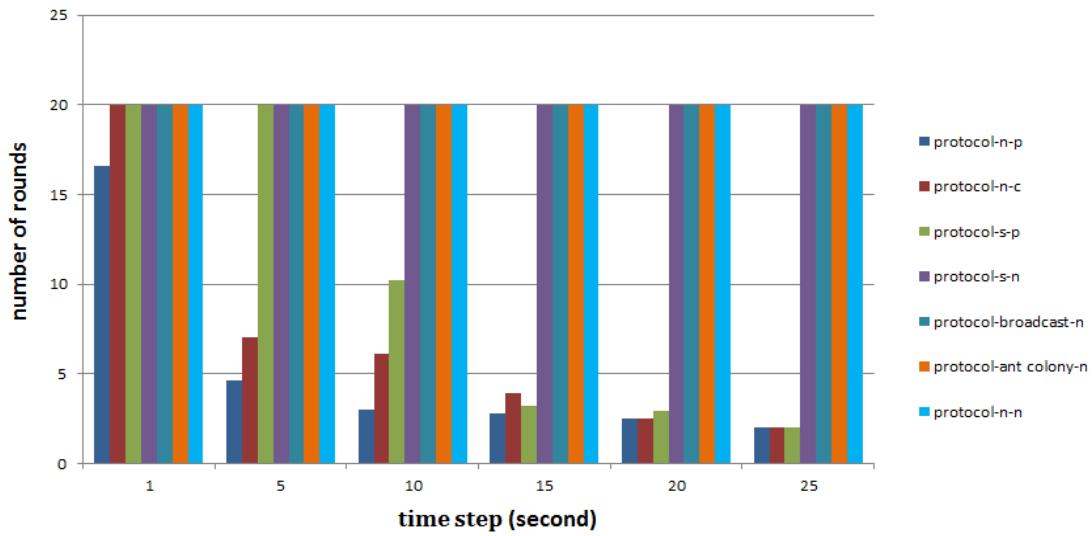


Fig.9 The number of rounds for finishing the game

Therefore agents have their required information for decision making and also do not share unnecessary information. In this protocol although agents do nothave complete information about the map but they possess required information nearly. Protocol-n-c that has the highest precision and recall between all protocols, presents a weaker performance in comparison with our works. Because in protocol-n-c agents share their mental models with all of other agents and the total message length is very higher than our work, in short time step the performance of it is weaker than our work. But in longer time step it presents a good result.Figure10 presents the average percentage of correct response for each protocol. This figure shows that protocol-n-c has the highest value and second highest value belong to our work. Indeed as stated before since protocol-n-c has the complete andcompatibleinformation about the map, it possesses the highest accuracy between all protocols. But the load of system in this protocol is high and causes the performance of it to be weaker than our work in time limited environments.

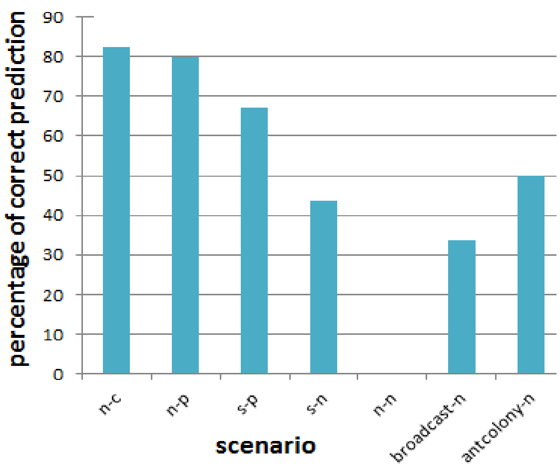


Fig.10 Correct prediction in each scenario

I. Conclusion

In this paper, we have applied our context aware architecture and sharing strategy introduced in [9, 10] to a dynamic environment enabling agents for sharingtheir mental models. This architecture is composed of three layers, “context layer”, “agent layer” and “mental model layer” plus a cross layer part named “background”. We have modified this structure and sharing strategy in order to appropriate operation on

dynamic environment. To evaluate the proposed sharing strategy, a static and dynamic version of a new environment which is a more complex variant of famous wumpus world game is modeled. The accuracy of shared information, the communication load and the performance of all methods in time-constrained conditions are compared with our method and the results show the superiority of our method.

We believe that agents can use this architecture to enhance teamwork activities such as role assignment, knowledge sharing, planning, etc. by constructing shared mental models. Also this sharing procedure can be done prior to any step in negotiation applications in order to form a shared mental model to converge the agents’ mental models. This shared mental model can help agents to propose better offers through better understanding of other agents’ minds. The context and background layers are common among all agents in the system. In applications with large number of agents, these two layers might become a bottleneck if the agents access these layers concurrently. A solution for this limitation is to define local instances of these layers for each agent. It needs appropriate distributed mechanisms in order to update and integrate the information of distributed context layer and background layer. Also in the conflict resolution framework, we use a superior agent named “coach” in “Right layer”. In a distributed environment with a large number of agents, access to this agent is difficult or even impossible. In these domains agents can be clustered and each cluster can have a superior agent named “cluster coach”.

This work can be improved by adding a set of methods to determine the way that each agent can trust to other agents. Also adding a learning mechanism for automatic formation of context and background layer is a future work of this paper. Furthermore this work can be applied in a teamwork application and its impact on team performance can be analyzed.

ACKNOWLEDGMENT

We would like to thank ICT Research Institute (ITRC) for supporting this research.

REFERENCES

[1] E. Phillips, S. Ososky, J. Grove, and F. Jentsch, "From Tools to Teammates," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 2011, pp. 1491-1495.

[2] John Yen,Xiaocong Fan, Shuang Sun, Timothy Hanratty, John Dumer," Agents with shared mental models for enhancing team decision makings," *Decision Support Systems*, vol.41, 2006, pp. 634-653.

[3] Yu Zhang," Role-Based Shared Mental Models," *Collaborative Technologies and Systems,* 2008, pp. 424-431.

[4] Daniel Fuller, Brian Magerko," Shared Mental Models in Improvisational Performance," *Proceedings of the Intelligent Narrative Technologies III Workshop*, ACM, 2010, pp 15-21.

[5] Kennedy, W.G, Trafton, J.G, "Using Simulations to Model Shared Mental Models," Proceedings of the Eighth International Conference on Cognitive Modeling, p. 253-254, 2007.

[6] Brigitte Burgemeestre, Jianwei Liu, JorisHulstijn, Yao-Hua Tan," Early requirements engineering for e-customs decision support: Assessing overlap in mental models," *Proceedings of CAiSE Forum*, 2009, pp. 31-36.

[7] Xiaocong Fan, Po-Chun Chen, John Yen," Learning HMM-based cognitive load models for supporting human-agent teamwork," *Cognitive Systems Research*, vol.11, 2010, pp. 108-119.

[8] Xiaocong Fan, John Yen," Realistic Cognitive Load Modeling for Enhancing Shared Mental Models in Human-Agent Collaboration," *AAMAS'07 Honolulu, Hawai'i, USA*, 2007, pp. 60.

[9] S. Salehi, F. Taghiyareh, M. Saffar, and K. Badie, "A context-aware architecture for mental model sharing in intelligent agents," *IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics*, 2012, pp. 313-318.

[10] S. Salehi, F. Taghiyareh, M. Saffar, and K. Badie, "A Context-Aware Architecture for Mental Model Sharing Through Semantic Movement in Intelligent Agents," *International Journal of Engineering-Transactions B: Applications,* vol. 25, p. 233-248, 2012.

[11] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing,* vol. 2, pp. 263-277, 2007.

[12] J. Hong, E. Suh, and S. J. Kim, "Context-aware systems: A literature review and classification," *Expert systems with applications,* vol. 36, pp. 8509-8522, 2009.

[13] J. Payton, "Simplifying Context-Aware Agent Coordination Using Context-Sensitive Data Structures," DTIC Document2004.

[14] H. J. Lee, J. E. Park, E. J. Ko, and J. W. Lee, "An agent-based context-aware system on handheld computers," *International Conference on Consumer Electronics (ICCE'06),* 2006, pp. 229-230.

[15] W. Chun-Dong and W. Xiu-Feng, "Multi-agent Based Architecture of Context-aware Systems," *International Conference on Multimedia and Ubiquitous Engineering, (MUE'07),* 2007, pp. 615-619.

[16] B. Y. Lim, A. K. Dey, and D. Avrahami, "Why and why not explanations improve the intelligibility of context-aware intelligent systems," *27th international conference on Human factors in computing systems,* 2009, pp. 2119-2128.

[17] B. Lim, "Improving Understanding, Trust, and Control with Intelligibility in Context-Aware Applications," *Human-Computer Interaction*, 2011.

[18] F. Schmitt, J. Cassens, M. Kindsmller, and M. Herczeg, "Mental models of ambient systems: a modular research framework," *Modeling and Using Context,* pp. 278-291, 2011.

[19] H. Harroud and A. Karmouch, "A policy based context-aware agent framework to support users mobility," *Advanced industrial conference on telecommunications /service assurance with partial and intermittent resources,* 2005, pp. 177-182.

[20] K. Arabshian and H. Schulzrinne, "Distributed context-aware agent architecture for global service discovery," *The Second International Workshop on Semantic Web Technology For Ubiquitous and Mobile Applications,* 2006, p.12-17.

[21] N. M. Sadeh, T. C. Chan, L. Van, O. Kwon, and K. Takizawa, "Creating an open agent environment for context-aware m-commerce," *Agentcities: Challenges in Open Agent Environments,* vol. 70, 2003.

[22] M. Hattori, K. Cho, A. Ohsuga, and M. Isshiki, "Context-aware agent platform in ubiquitous environments and its verification tests," *Systems and Computers in Japan,* pp. 547-552, 2003.

[23] O. Kwon, S. Choi, and G. Park, "NAMA: a context-aware multi-agent based web service approach to proactive need identification for personalized reminder systems," *Expert Systems with Applications,* vol. 29, pp. 17-32, 2005.

[24] KaivanKamali, Xiaocong Fan, John Yen," Multiparty Proactive Communication: A Perspective for Evolving Shared Mental Models," American Association for Artificial Intelligence, 2006, pp. 685-691.

[25] C. Jonker, M. van Riemsdijk, and B. Vermeulen, "Shared Mental Models," *Coordination, Organizations, Institutions, and Norms in Agent Systems VI,* pp. 132-151, 2011.

[26] M. Thielscher, "Designing a FLUX Agent for the Dynamic Wumpus World."*PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING-INTERNATIONAL CONFERENCE-*, 2002, pp. 435-448

[27] M. Tambe, "Recursive agent and agent-group tracking in a real-time, dynamic environment," *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95),* 1995, pp. 368-375.

[28] S. Ilarri, E. Mena, and A. Illarramendi, "Using cooperative mobile agents to monitor distributed and dynamic environments," *Information Sciences,* vol. 178, pp. 2105-2127, 2008.

[29] A. Hiura, T. Kuroda, N. Inuzuka, K. Itoh, M. Yamada, H. Seki, and H. Itoh, "Cooperative behavior of various agents in dynamic environment," *Computers & industrial engineering,* vol. 33, pp. 601-604, 1997.

[30] L. Wickramasinghe and L. Alahakoon, "Dynamic self organizing maps for discovery and sharing of knowledge in multi agent systems," *Web Intelligence and Agent Systems,* vol. 3, pp. 31-47, 2005.

[31] G. T. Furlong, The conflict resolution toolbox: models & maps for analyzing, diagnosing and resolving conflict: Wiley, 2005

[32] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 26, pp. 29-41, 1996.

[33] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation,*vol. 1, pp. 53-66, 1997.

Sajjad Salehi has received his BSc in IT Engineering from University of Isfahan, Isfahan, Iran, in 2009 his MSc from Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran in 2012. His research interests include Artificial Intelligence, Multi-Agent System, Mental Models and Data Mining.

Fattaneh Taghiyarehis a faculty member at the Information Technology Department, School of Electrical and Computer Engineering, University of Tehran since 2001. She received her Ph.D. from Tokyo Institute of Technology in 2000. Her research interests are Technology Enhanced Learning, Multi-Agent Systems, and Human-Centered Computing Systems. Her current research involves the intersection of Web-Services, Personalization, and adaptive LMSs as well as applying ontology to semantic web. She was past-manager of the Department of Information Technology at the School of Electrical Computer Engineering, and former Director of Technology Incubator, University of Tehran. She is a member of Editorial Board of International Journal of Information & Communication Technology. Recently she has established eLearning laboratory with the mission of providing an standard view for e-learning materials and platforms. Currently she is supervising IT Foundation lab, e-learning lab, and multi-agent systems lab.

Mohammad TaghiSaffar started his M.Sc. at University of Tehran in 2010. His main research interest are multi-agent systems, distributed planning, non-monotonic reasoning, argumentation and dialogue games. During his M.Sc. time he joined Multi-Agent Systems Laboratory at university of Tehran, under the supervision of Dr. Fattaneh Taghiyareh. His thesis was about cooperative planning by argumentation dialouges based on the idea of sharing mental model in artificial agents.

Kambiz Badie has received his B.Sc., M.Sc. and Ph.D. In Electrical Engineering from Tokyo Institute of Technology, Japan, majoring in Pattern Recognition & Artificial Intelligentce. His majer research interests are Machine Learning, Cognitive Modeling, and Systematic Knowledge Processing in general, and Analogical Knowledge Processing, Experience-Based Modeling, and Interpetative Modeling in particular with emphasis on Idea , Technique, and Content Generation. At presents he is an associate professor in IT Research Faculty of ICT Research Institute (ITRC).