On Representing and Sharing Knowledge in Collaborative Problem Solving

Heba Al-Juaidy, Lina Abu Jaradeh, Duha Qutaishat, and Nadim Obeid

Abstract. In this paper, we propose an agent-based framework for collaborative problem-solving. We emphasize the knowledge representation and knowledge sharing issues. We employ a three-valued based Temporal First-Order Nonmonotonic Logic that allows an explicit representation of events/actions and can handle dialogue game protocols and temporal aspects explicitly. A prototype is developed with a case to guide and assist evacuees in an emergency evacuation from a building.

Keywords: Knowledge Sharing, Multi-Agent Systems, Knowledge Representation, Problem Solving.

1 Introduction

Collaborative Problem Solving (CPS) is the process by which a collection of intelligent agents work together to partition a complex, large and/or unpredictable problem into an appropriate set of simpler sub-problems where each will be (partially) solved by one or a group of expert agents and finally the partial solutions are integrated to produce a solution to the whole problem[14]. This decomposition allows each agent to use the most appropriate technique to solve the sub-problem to which it is assigned. Multi-Agent System (MAS) represents an appropriate approach for solving inherently distributed problems, whereby clearly different and independent processes can be distinguished. In CPS settings, the use of MAS offers conceptual clarity, flexibility, the ability to handle applications with a natural spatial distribution and with uncertain information.

Heba Al-juaidy · Lina Abu Jaradeh · Duha Qutaishat · Nadim Obeid
Department of Computer Information Systems, King Abdullah II School for Information Technology, The University of Jordan
email: `obein@ju.edu.jo`

M. Ali et al. (Eds.): *Contemporary Challenges & Solutions in Applied AI*, SCI 489, pp. 61–66.
DOI: 10.1007/978-3-319-00651-2_9

In this paper, we propose an agent-based framework for CPS. We emphasize the Knowledge Representation (KR) and Knowledge Sharing (KS) issues in a distributed agent-based system. We employ a three-valued based Temporal First-Order Nonmonotonic Logic (TFONL) that allows an explicit representation of events/actions [cf. 13, 16] and can handle dialogue game protocols and temporal aspects explicitly. A prototype is developed with a case to guide and assist evacuees in a emergency evacuation from building. In section 2 we discuss the suitability of MAS for CPS. In section 3 we discuss KR and KS issues and present the example. Section 4 is dedicated to discussions and related works.

2 CPS and MAS

The idea of CPS is to decompose a problem into a set of sub-tasks where each has some form of relation/association with other sub-tasks that must be dealt with by the appropriate agents that possess enough PS knowledge to apply their own expertise to its sub-task. Decentralization of the tasks seems to be a reasonable way to keep control within large complex problems [8]. Multi-Agents Systems (MAS) have been proposed as a suitable model for handling complex, distributed and heterogeneous systems [7, 15]. An MAS can be defined as: a collection of agents with their own problem solving capabilities and which are able to interact among them in order to reach an overall goal [8]. Agents are specialized problem solving entities. They are autonomous as they have control both over their internal state and over their actions.

In CPS Situation, agents can help each other by negotiating a partition of the problem into manageable sub-problems/tasks among themselves according to their abilities, expertise and skills. The key issue to be resolved in sub-problems and their associated tasks is how tasks are to be distributed among the agents and how to dynamically configure the system by distributing the software components on the available hardware hosts so that they work together as a whole unit to meet changing requirements [1].

Due to the dynamic and unpredictable nature of the environment, in which agents operate, it is not possible to give, at the outset, a complete specification of all the tasks and the knowledge/expertise required. Thus, there is a need for an environment that integrates the knowledge of the various agents and partial results, so that agents could have access to information, expertise and knowledge they need. Depending on its knowledge and reasoning ability, an agent may pursue an objective/goal in order to help it in deciding what method and/or technique to use for another objective/goal. A CPS task may encompass planning, scheduling, and collaborative diagnosis [7]. Collaboration and different expertise has its problems. CPS may require the collaborators to be involved in a process of reformulation and questioning until they reach a point of consensus; we deviate from one complete solution and rather develop solutions in small steps refinements [6].

3 Dialogue Moves in TFONL

In CPS situations, agents have incomplete knowledge of their environments. This makes reasoning complex since the closed world assumption can no longer be applied; an agent cannot assume that a fact is false just because it does not know about it. Therefore, it is important to cater for representing and updating the incomplete knowledge which would be useful to an autonomous agent capable of KS and of manipulating its environment. It is possible to express an agent's knowledge using classical logic. However, it may prove to be more difficult and quite unnatural. Furthermore, agents in a dialogue make statements that are realistic based on the context. However, these statements may become futile during the dialogue when information previously unknown become available which may cause agents to revise their knowledge. Defeasible logic is suitable for the dynamics of argumentation and dialogue where agents could change their beliefs.

In this paper, we employ a three-valued based Temporal First-Order Nonmonotonic Logic (TFONL) that allows an explicit representation of time and events/actions [cf. 13, 16]. TFONL is an extension of the quantified version of the non-temporal system T3 [cf. 10, 11, 12]. The language, L_{T3}, of T3 is that of Kleene's three-valued logic extended with the modal operators "M" (Epistemic Possibility) and "P" (Plausibility). The material implication "$\supset$" can be defined as follows: ($A \supset B = M(\sim A \& B) \vee \sim A \vee B$. In T3, "L" is the dual of "M" and "N" be the dual of "P", i.e., $LA \equiv \sim M \sim A$ and $NA \equiv \sim P \sim A$ where $A \equiv B$ is equivalent to $A \supset B$ and $B \supset A$. $A \Rightarrow B$ represents the default A A:B/B [10].

Nonmonotonic reasoning is represented via the operators M (*epistemic possibility)* and P (*plausibility*). Informally, MA states that A is not established as false. Using M, we may define the operators U (*undefined*), D (*defined*) and $\neg$ (*classical negation*) where UA is true if the truth value of A is undefined and DA is true if the truth value of A is not undefined. More specifically, $UA \equiv MA \& M \sim A$, $DA \equiv \sim UA$ and $\neg A \equiv DA \& \sim A$

Within the framework of TFONL, it is possible to formalize dialogue moves and the rules of protocols of the required types of dialogue. These rules are nonmonotonic because the set of propositions to which an agent is committed and the validity of moves vary from one move to another. Let L_{Com} specify the locutions which the agents participating in a dialogue are able to express. A dialogue consists of a course of successive moves made by the participants. A Dialogue Move can be defined as follows:

Definition 3.1. A Dialogue Move M can be defined as a 7-tuple as follows:

M=<Id(M), Sender(M), τ(M), δ(M), Content(M),Receiver(M), Target(M)> where Id(M) is the identifier of M, Sender(M) is the speaker of <δ(M),Content(M)>, τ (M) is the time of M, $\delta(M) \in$ {Assert, Accept, Reject, Retract, Question, Justify, Challenge }, Content(M) is the content of M, Receiver(M) is the addressee and Target(M) is a previous move to which M is a reply.

We now present our example. Emergency evacuation is the urgent movement of people from a place due to the occurrence of some dangerous event [2] which is challenging. MAS are particularly suitable for assisting in such tasks.

The Computer Information System (CIS) department in at the university of Jordan consists of seven computer labs, five lecture halls, a students' hall and service rooms. Let A_1, …, A_k stand for areas, Z_1, …, Z_r for Zones. We shall use Z_{ij} to denote zone i in area j. The zones in CIS are as follows: Z_{11} = [Exit1 ,Lab206 ,Lab207 , k205], Z_{21} = [2 Elevators, Lab201 ,K204 ,k201 , sitting room], Z_{12} = [Exit2, Service rooms, Lab203, Lab202, K202] and Z_{22} = [Exit3 , Lab 204, Lab 205, K 203] (cf. Fig. 1 and Fig. 2).

With each zone Z_{ij}, we associate an agent group G_{ij} that includes Z_{ij}-Supervisor, Z_{ij}-Monitor, Z_{ij}-Info-Coll, Z_{ij}-Guide. In case of an anomaly in a zone Z_{kl}, the Z_{kl}-Monitor informs all agents in its zone namely, Z_{kl}-Supervisor, Z_{kl}-Info-Coll and Z_{kl}-Guide. The Z_{kl}-Supervisor informs the Al-Supervisor of the anomaly in order to activate the appropriate alarms and/or call rescue teams. It keeps track of the workflow of the evacuation process and manages the actions performed by each agent in its zone. It can negotiate with other zones' supervisors if it needs help to carry out its plans such as guiding people through appropriate zones to avoid congestion. Z_{kl}-Info-Coll collects information about the state of its zone and other zones such as congestion, safety, open/closed gates and the states of evacuees such as injuries, agents breakdown and so on. Z_{kl}-Guide guide people to safe exits using safe paths taking into consideration what is reported by Z_{kl}-Info-Coll.

With each area $\mathbf{A_l}$, we associate an agent group G_l that includes A_l-Supervisor, A_l-Info-Coll, A_l-Planner, A_l-Guide. The A_l-Supervisor, A_l-Info-Coll and A_l-Guide have similar tasks to those at zone levels at the area level. The A_l-Planner determines the alternative sets of safe routes through the different zones in A_l and other areas to an exit and send it to A_{L}.Guide.

The agents can make use of fluents such as gas-smell, fire-heat, smoke, emergency and so on. The environment may include propositions such as Exit(Gate1), Exit(Gate2), Location(Gate1, Z11), Location(Elevator1, Z21), State(Gate2, Closed) and State(Gate1, Open). We employ rules such as (R1) and (R2):

(R1) T ⇒ Clear(P) &Path(P)

(R2) Exit(G)&Location(G, Zij)&on-Path(P, G)&State(G, Closed)&
¬Crowded(G) ⇒ Open(Zij-Sup, G).

(R1) states that by default, paths are clear. (R2) states that if there is an exit G that is not crowded, on a safe path, and G is closed, then open G.

Suppose that there is a fire in Z11 (cf. Fig. 1), then Z11-Monitor detects the fire and informs Z11-Supervisor. This move is not a reply to any previous move.

M_1 = <1, Z11-Monitor, t_1, Assert, fire in Z11, Z11-Supervisor, 0>

In M_2, Z11-Supervisor informs Z11-Monitor that it accepts the message.

M_2 = <1, Z11-supervisor, t_2, Accept, fire in Z11, Z11-Monitor, 1>

Z11-Supervisor informs A1-supervisor that there is a fire in Z11.

M_3 = <3, Z11-Supervisor, t_3, Assert, fire in Z11, A1-Supervisor, 0>

In M_4, A1-Supervisor informs Z11-Supervisor that it accepts the message.

M_4 = <4, A1-Supervisor, t_4, Accept, fire in Z11, Z11-Supervisor, 3>

A1-Supervisor can inform A1-Planner that there is a fire in Z11 and so on.

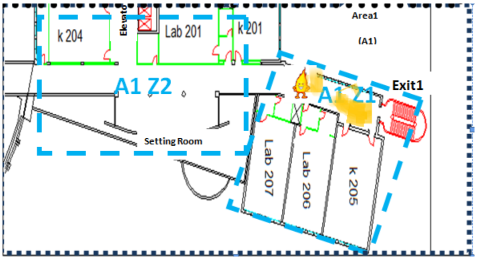

Fig. 1 Fire in Z11

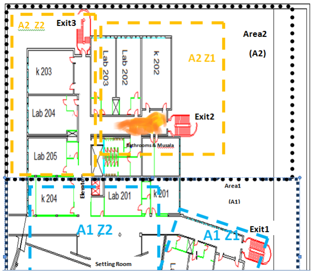

Fig. 2 Fire in Z12

In the same way, A1-Supervisor informs Zj1-supervisor, Ak-supervisors, and A1-Planner. A1-Planner ids required to propose an evacuation route.

Similarly, A2-Supervisor and each zone supervisor in A2 will inform appropriate agents do what is required. Each zone planner will propose a safe route through its zone. The area planners, will integrate these partial solutions to propose safe routes. A1-Planner has to provide a safe route and it may propose

R1: Z11→Z21→ELEVATORS and R2: Z11→ Z21→Z22→Exit3 as in:

M_i = <4, A1-Planner, t_i, Assert , {R1, R2}, A1-Supervisor, 5>

Similarly, if there is a fire in Z12 (cf Figure 2) then the following steps may be taken: Z12-Monitor informs Z12-Supervisor of event. Z12-Supervisor informs A2-supervisor and all agents in Z12 of the event. A2-Supervisor informs Zj1-supervisor, Ak-supervisors and A2-Planner to propose an evacuation route. Each Ak-Supervisor inform agents in its area to do what is required according to their specialization. For instance, Z22-Planner proposes a route through Exit 3. Z21-Planner proposes a route through Exit 1. A2-Planner proposes a route through Exit 3. A1-Planner proposes a route through Exit 1. A1-Supervisor and A2-Supervisor can negotiate a plan to guide people.

4 Discussion and Comparison with Related Works

To our knowledge, little consideration is to dialogue and argumentation in CPS. In [7] the role of dialogue in MAS is shown. In [15] the role of agents in the development of a KS system is highlighted. Some formalisms have been suggested for specifying and verifying protocols or tracking agents' commitments during dialogue [3, 4, 9]. In [5], a generic framework for specification of dialogue game protocols is presented These protocols are based on classical logic. These approaches do not commit to a mechanism for agents to think about the acceptability of arguments. TFONL, however, can handle dialogue game protocols and temporal aspects explicitly.

We have in this paper emphasized the KR and KS issues. A prototype is developedd with a case to guide evacuees in a emergency evacuation form a building.

It is important to identify relevant commitments which an agent has to satisfy and to investigate how to integrate this proposal with techniques used in planning to identify strategies to satisfy important commitments [9]. It is useful to further investigate strategic and tactic reasoning in solving more complex problems.

References

1. Al-Areqi, S., Hudaib, A., Obeid, N.: Improving Availability in Distributed Component-Based Systems via Replication. In: Nguyen, N.T., Trawiński, B., Jung, J.J. (eds.) New Challenges for Intelligent Information and Database Systems. SCI, vol. 351, pp. 43–52. Springer, Heidelberg (2011)
2. Bonabeau, E.: Agent-Based Modeling: Methods and Techniques for simulating human systems. Proc. National Academy of Sciences USA 99, 7280–7287 (2002)
3. Chesani, F., Mello, P., Montali, M., Torroni, P.: Commitment Tracking via the Reactive Event Calculus. In: IJCAI, pp. 91–96 (2009)
4. Giordano, L., Alberto, M., Camilla, S.: Specifying and Verifying Interaction Protocols in a Temporal Action Logic. J. Applied Logic 5(2), 214–234 (2007)
5. McBurney, P., Parsons, S.: Dialogue Games for Agent Argumentation. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence, vol. 261 (2009)
6. Moubaiddin, A., Obeid, N.: The Role of Dialogue in Remote Diagnostics. In: 20th Int. Conf. on COMADEM (2007)
7. Moubaiddin, A., Obeid, N.: Dialogue and Argumentation in Multi-Agent Diagnosis. In: Nguyen, N.T., Katarzyniak, R. (eds.) New Chall. in Appl. Intel. Tech. SCI, vol. 134, pp. 13–22. Springer, Heidelberg (2008)
8. Moubaiddin, A., Obeid, N.: Partial Information Basis for Agent-Based Collaborative Dialogue. Applied Intelligence 30(2), 142–167 (2009)
9. Moubaiddin, A., Obeid, N.: On Formalizing Social Commitments in Dialogue and Argumentation Models Using Temporal Defeasible Logic. Knowledge and Information Systems (2012), doi:10.1007/s10115-012-0578-6
10. Obeid, N.: Three Valued Logic and Nonmonotonic Reasoning. Computers and Artificial Intelligence 15(6), 509–530 (1996)
11. Obeid, N.: Towards a Model of Learning Through Communication. Knowledge and Information Systems 2, 498–508 (2000)
12. Obeid, N.: Fault Diagnosis Using Three-Valued Based Nonmonotonic Logic. International Journal of COMADEM 3(2), 17–28 (2000)
13. Obeid, N.: A Formalism for Representing and Reasoning with Temporal Information, Event and Change. Applied Intelligence 23(2), 109–119 (2005)
14. Obeid, N., Moubaiddin, A.: On the Role of Dialogue and Argumentation In Collaborative Problem Solving. In: ISDA, pp. 1202–1208 (2009)
15. Obeid, N., Moubaiddin, A.: Towards a Formal Model of Knowledge Sharing in Complex Systems. In: Szczerbicki, E., Nguyen, N.T. (eds.) Smart Information and Knowledge Management. SCI, vol. 260, pp. 53–82. Springer, Heidelberg (2010)
16. Obeid, N., Rao, B.K.N.R.: On Integrating Event Definition and Event Detection. Knowledge and Information Systems 22(2), 129–158 (2010)