2009 Ninth International Conference on Intelligent Systems Design and Applications

On The Role Of Dialogue And Argumentation In Collaborative Problem Solving

Nadim Obeid
Dept. Computer Information Systems
King Abdullah II School for IT
The university of Jordan, Amman, Jordan
obein@ju.edu.jo

Asma Moubaiddin
Department of Linguistics
Faculty of Foreign Languages
The university of Jordan, Amman, Jordan

Abstract

In this paper, we make a first step towards a formal model of dialogue and argumentation for a multi-agent problem solving. We shall present a multi-agent system for problem solving. We shall the notion of collaborative problem solving and discuss some of the related communication issues. We propose a Partial Information State based framework for dialogue and argumentation. We shall employ a three-valued based nonmonotonic logic for representing and reasoning about partial information. We show via an example that the system can handle collaborative problem-solving tasks.

Keywords: *Problem Solving, Multi-Agent System, Collaboration, Dialogue, Argumentation.*

1. Introduction

Multi-Agent Systems (MAS) are particularly well suited to Collaborative Problem Solving (CPS) whether the MAS comprises cooperative or competitive agents. Issues such as dynamic team formation for cooperative agents and partner selection strategies with competitive agents are important [19]. Furthermore, the most demanding task in computing optimal solutions for complex problems is to find adequate problem decompositions and to detect the dependencies between the derived subtasks. Once a complex task is decomposed, each subtask is delegated to an agent. This decomposition allows each agent to use the most appropriate technique to solve its sub-problem. In CPS situations, agents need to communicate in order to coordinate with one another to properly manage interdependencies. Single messages exchanged by agents are not sufficient if they want to collaborate, devise and exchange proofs, solve conflicts and negotiate. CPS requires agents to participate in complex interactions such as negotiations, persuasions, deliberations, etc. Therefore, agents have to take part in conversations which are coherent sequences of utterances. However, when MASs are employed in larger applications, issues of scaling need to be considered [15].

In this paper, we make a first step towards a formal model of dialogue and argumentation for CPS. We present, in section 2, a Multi-Agent System (MAS) for Problem Solving (PS). Section 3 will be concerned with the notion of CPS. In section 4, we discuss some of the related communication issues. In section 5, we present TFONL, a logic for reasoning with incomplete information. We propose a Partial Information State (PIS)-based framework for dialogue and argumentation. We shall employ a three-valued based NonMonotonic Logic (NML) for representing and reasoning about Partial Information (PI). We show via an example that the system can to handle CPS tasks. It is important to note that issues of scaling and problem decomposition are not addressed in this paper.

2. A Multi-Agent System for Problem Solving

Agents are specialized PS entities that have well-defined boundaries and interfaces. An agent may need to have some of the following properties such as : autonomy, reactivity, pro-activeness and social ability. Agent autonomy is a reflection of its control over both its internal state and its behavior. It relates to an agent's ability to make its own decisions, e.g., about what type of information is to be communicated, to which of the agents it knows about and what to make out of incoming information from other agents. Reactivity is related to the agent ability to reason and respond to changes. Pro-activeness reflects the agent ability to take initiatives to exhibit goal-directed behavior. Social ability requires that agents interact with one another via an Agent Communication Language (ACL) in order to achieve goals.

An MAS for PS consists of a collection of agents that interact with each other within a context and a

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.60

dialog framework. The dialog framework specifies for every agent/component its ontology, communication language and communication protocol. It involves the following elements: (1) Specialized agents; (2) Formal inference machinery; (3) Theories which are sets of formulae in the agent language that are considered true and (4) bridging rules which are rules of inference that an agent can use to make use of formulae coming from other agents. Bridging rules are needed when agents employ different ontologies and languages. Agents may also employ different logic systems depending on their knowledge and the purpose of their reasoning.

In this paper, we make the assumption that all agents use the same language and an agent g1 could make use of a formula to which another agent g2 is committed even if g1 does not accept it as true.

3. Collaborative Problem Solving

There are many PS tasks that are characterized by one or several of the following properties: (1) large, (2) complex, (3) spatially distributed, (4) need extensive communication and (5) a large degree of functional specialization between the agents. A problem that satisfies one, or more, of these properties cannot be effectively and efficiently handled by a single agent.

The idea of CPS is to decompose a problem into a set of sub-tasks/objectives where each has some form of relation/association with other sub-tasks/objectives that must be dealt with by the appropriate agents. Each sub-task is handled by an appropriate agent that possesses enough PS knowledge to apply its own expertise to its sub-task. Each agent has some abstract view of some of the other sub-tasks that are related to its sub-task in order to guarantee that its PS strategy is in agreement with that of the others. Each agent can communicate with other agents. Beside expertise, the distribution of CPS knowledge among a set of cooperating agents could follow various criteria such as physical proximity and functionality. We define PS as a process that roughly consists of the following three general phases:

(p1) Determining Objectives/Goals: an agent deals with goals, determining to which it is committed.

(p2) Deciding on methods/techniques for achieving the selected objectives: an agent determines methods/techniques (e.g., proofs, actions) to employ in order to make progress towards achieving its objectives.

(p3) Applying methods: an agent applies the methods/techniques and monitors the process/success that is made towards achieving the objectives.

We shall assume that an agent is quite dynamic and flexible in the way it employs the techniques and methods it can handle and in how it works towards achieving its objectives/goals. Depending on its 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. We believe that the general CPS process remains the same, regardless of the task and domain [1]. Thus, a CPS model has to be domain-independent. Furthermore, our purpose is not to provide a specific PS strategy, instead we aim to address some of the issues of communication and collaboration in a PS context. It is important to note that the level of collaboration in the PS may vary considerably depending on the nature of the problem and the involved agents' organization, specialties, degree of autonomy and (reasoning) capabilities. Furthermore, collaboration may cover some or all of the phases depending on the problem and the expertise needed by one or more of the participants.

It is difficult to identify the possible range of CPS. A CPS task may encompass planning, scheduling, and collaborative diagnosis [10]. Collaboration and different expertise has its problems. For instance, what constitutes an adequate problem formulation for one agent may not be something that another agent can describe appropriately. PS requires the collaborators to be involved in a process of reformulation and questioning until they reach a point of consensus. CPS requires that we deviate from one complete PS solution and go about developing solutions in small steps refinements [9].

4. Reasoning With Incomplete Information

Agents in an MAS have incomplete knowledge. The agent's knowledge and reasoning capability are expressed in a Temporal First Order NML (TFONL). The system is based on the quantified version of the non-temporal system T3, which is a three-valued based NML system [12]. The language has a third value *undefined* which was used by Kleene to describe computations that may not terminate. One of the advantages of T3 is that *defaults* can be represented as sentences in the object language in the system. There is a one-to-one correspondence between extensions of a default theory and appropriate minimal PIS which provide the semantic account (models) of T3 [13].

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). In T3, L is the dual of M and N be the dual of P, i.e., LA ≡

~M~A and NA ≡ ~P~A. In [12], a truth-functional implication ⊃ is defined as follows: A ⊃ B ≡ M(~A & B) V ~A V B.

NonMonotonic (NM) reasoning is represented via the *epistemic possibility operator* M and the *plausibility operator* P. Informally, MA states that A is not established as false. Using M, we may define the operators U (*undefined*), D (*defined*) and ¬ (*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. We shall employ the following notational definitions: UA ≡ MA&M~A, DA ≡ ~UA and ¬A ≡ DA & ~A. TFONL is suitable for argumentation and dialogue frameworks [11, 14].

5. PIS-Based Based Dialogue Management

Agents, in an MAS, are expected to have the ability to be involved in coherent conversations. Several approaches to modeling communication have been proposed [4, 2]. Agents' collaboration requires a sophisticated ACL. There are two main ACLs: KQML [5] and FIPA [7]. These languages have been designed to be widely applicable. This feature can be both a strength and a weakness: agents participating in conversations have too many choices of what to utter at each turn, and thus agent dialogues may endure a state-space explosion. The need for a language that allows sufficient flexibility of expression while avoiding state-space explosion had led agent communications researchers to the study of formal dialogue games [8]. Dialogue games are rule-governed interactions between two or more dialogue participants (or agents), where each participant makes a "move" by making utterances, according to a defined set of rules. The rules typically define what locutions may or must be uttered in different circumstances.

Various types of dialogues are distinguished [17]. Among these are: (1) persuasion, which is needed to resolve conflicting points of view, (2) negotiation where the participants aim to reach an agreement that is beneficial for individual parties, (3) inquiry where the aim is to collectively obtain more reliable knowledge, (4) deliberation, which is driven by the need to take a collective decision and (5) information-seeking where one participant asks for information known by another. The distinction between the types of dialogue is based on collective goals, individual goals and reasons for starting the dialogue. It is possible that in the course of communication, there occurs a shift from one type of dialogue to another. Dialogue embedding takes place when the embedded dialogue is functionally related to the first one. For instance, a persuasion dialogue may require an information-seeking sub-dialogue.

5.1. Argumentation System

We shall adopt the argument-based approach [2] in which the agents' reasoning capabilities are associated with their ability to argue and the strength of their arguments. Arguments, in NML systems, are logical proofs where some of their steps can be defeated because it is possible to provide an argument for both a proposition and its negation. Hence, a defeasible argument is a structured piece of information that might be defeated by other (defeasible) arguments.

One of the essential features of the proof system of T3 is that it allows free and complete access to all stages of the proof process. The proof method proceeds by the construction of a tableau. This is a tree-structure in which all the possible models allowed by the premises and negated conclusion are set out and examined for consistency. The construction of the tree is governed by rules for each logical connective in the language. These rules are closely related to the semantics of the language. The method performs a case-wise analysis of all models in which the premises might be true while contradicting the conclusion, if no such models are found to exist, the theorem is proven. We employ this method because it allows an agent absolute access to every stage of the proof process. Thus, unlike a proof, in classical monotonic logic, an argument does not establish warrant for a conclusion, in a definite way as it may be defeated by counter-arguments which are defeasible.

5.2. PIS-Based dialogue

We shall adopt a model of dialogue that is based on commitment. Dialogues are viewed in terms of the relevant PI that the participants have at each stage. Hence, the logic employed for reasoning by the agents is based on PISs. For each agent g, PIS consists of a Knowledge Base, KB(g), that embeds a dialogue Context and Goals. The reasoning capability of g is expressed in the logic system TFONL.

A dialogue consists of a structured sequence of utterances (moves) made by the dialogue participants.

Definition 5.1. A dialogue system is a triple

$D = (L_{COM}, \theta, \sigma)$

where L_{COM} is the communication language that specifies the locutions which the participants are able to express.

Let L_{COM} = {Assert A, Retract A, Accept A, Reject A, Question A and Challenge A} where A $\in L_{\mathbf{Topic}}$ (e.g., topic of the dialogue). θ is a protocol for L_{COM} and σ is a set of rules that specify the effects of locutions in L_{COM} on the participants' commitments. The coherence of a dialogue moves is tied to local interactions that are dependent on the agent's particular situation reflected in the changes in its PIS and agent's goals..

Definition 5.2. A dialogue move M is a 5-tuple M= <ID(M), SD(M), δ(M), LOC(M),TRG(M)> where ID(M) is the identifier of M (i.e., ID(M) = i indicates that M is the i^{th} move in the dialogue sequemce), SD(M) is the participant that utters the locution, δ(M) ∈ {Assert, Retract, Accept, Reject, Question, Challenge}, LOC(M) is the sentence which the sender utters and TRG(M) is the target of M.

The PIS of the agents may change as a result of the interpretation of the moves and that these changes may help in triggering the production of a succeeding move. The interpretation involves an integration of the exchanged information with the PIS of the receiver. The context is considered to be a consistent subset of an agent's PIS, namely those propositions which bear on the interpretation of the utterance on hand and on the propositions that are relevant to producing the objectives/goal(s).

5.2.1. Dialogue context and effect rules. Agents employ context in a dialogue to judge the relevance of moves and to determine the continual change needed to their PIS throughout the different stages. It can be defined as the set of all conditions that may influence the understanding and generation of appropriate locutions. A model of context should include: (1) information needed for the interpretation (and generation) of appropriate locutions; (2) information about participants' goals and beliefs; (3) information about the interaction (e.g., protocols, interpretation, evaluation and application of previous utterances).

Let g and g1 be agents and D_k ($1 \leq k < \infty$) refers to a finite sequence of moves $M_1, . . ., M_k$. It is not possible to give a precise definition of context within the scope of this paper. We shall employ "Context(D_k, g, g1)" to refer to the context of a dialogue d between g and g1, at stage k, from the perspective of g. We shall just present the effect rule for "Assert" [11]. Let $j < i$, M_j a move made by participant g1, and M ≡ <i, g, Assert, A, j> then Context(g, D_i, g1)= Context(g, D_{i-1},g1)∪{A} and Context(g1, D_i, g) = Context(G1, D_{i-1}, G).

5.2.2. Rules of protocols of some dialogue types. We may define the rules of protocols for all the dialogue types. However, due to lack of space, we shall only give these rules for inquiry dialogue about a proposition A involving g and g1.

(1) g seeks a support/proof for A. It begins with a move "Assert B →A" or "Assert B ⇒ A", for some sentence B or "Assert UA".
(2) g1 either reply "Accept B →A" or "Accept B ⇒ A" or "Challenge B→A" or "Challenge B ⇒ A".
(3) g replies to a challenge with "Assert P" where P is a proof of the last proposition challenged by G1.
(4) Go to step (2) for every proposition C∈P. That is, substitute C for B → A or B ⇒ A.
(5) g1 seeks a support/proof for B, i.e., it replies with "Assert E → B" or "Assert E ⇒ B", for some sentence B or "Assert UB".
(6) If Context(g, D, g1) ∪ Context(g1, D, g) |- A then the dialogue terminates successfully.
(7) The agents reverse roles and the appropriate agent seeks a support/proof for E (step 5).

In CPS contexts, (partial) solutions to sub-problems can be developed, by one or more participants, incrementally, on the basis of the previous steps captured in the dialogue history, and the current PISs of the participants. NM theorem proving (of T3) can be used to determine what can monotonically be accomplished and the dialogue is used to communicate what is considered necessary or required to complete a specific task step. Failure to provide the needed/missing information by one agent may leave no choice by to make assumptions, i.e., by invoking NM inference rules. In doing so, dialogue is integrated closely with PS and is invoked when an agent is unable to complete a task using its own knowledge.

6. A CPS Example

MAS are particularly applicable to CPS in many application domains, such as distributed information retrieval, traffic monitoring systems, Distributed diagnosis and Grid computing [19]. In this section we choose a very simple example from the domain of distributed diagnosis that shows some aspects of agent-based CPS and how a participant can use the proof method of T3 in a dialogue and can have access to every stage of the proof process.

Consider a case where we have a device D connected to a circuit S12 that consists of a series, S12, of two batteries, B1 and B2. , connected to a device D. Suppose that D is not working and the task of a group of agents is to restore the normal operation of D. The agents have to establish if either S12 or D is faulty in order to determine what repair(s) has(ve) to be performed. To do so they have to start with various testing. g1 will carry the process of testing the voltage of B1 and B2. However, g1 need the help of g2 to decide how to proceed in the testing process.

We assume that g1 and g2 have different skills and diagnostic knowledge and S12 is faulty. We aim to show how agents g1 and g2 collaborate in performing tests and reasoning in order to determine the faulty battery(ies) so that the appropriate repair procedure (e.g., RP1 or RP2), described below, could be performed. g2's task is to find out which component of C is faulty. Suppose that there is one new battery, New-B, kept in the store and that there are two repair procedures: (1) RP1(B,NB) which replaces B with NB and (2) RP2(B_1, B_2, NB_1, NB_2) which requires a drive to town, fetch two new batteries NB_1 and NB_2 to replace B_1 and B_2 respectively. We employ (1) Batt(B) to mean that B is a battery, (2) Volt(X,V) to mean that the voltage of X (S12, B1 or B2) is V and (3) Ok(V) to mean that $1.2 \leq V \leq 1.6$.

Suppose that the knowledge of g1, KB(g1), contains the following facts and/or rules:

($Ik1_1$) S12 is faulty iff its voltage is less than 2.4.

($Ik1_2$) if a battery is faulty then replace it with a new one.

More formally, KB(g1) contains the following facts and/or rules:

($K1_1$) Volt(S12,V_0) & V_0<2.4 $\leftrightarrow$ Faulty(S12)

($K1_2$) ~Ok(V_2) $\rightarrow$ RP1(B_2, New-B),

($K1_3$) ~Ok(V_1) $\rightarrow$ RP1(B_1, New-B)

Suppose g2 is more knowledgeable and its knowledge base, KB(g2), contains the following facts and/or rules:

($IK2_1$) the same as ($Ik1_1$)

($K2_2$) if the circuit C is faulty and the voltage of the series of batteries S12 is Ok then check(D))

($K2_3$) if one battery is faulty and the other is Ok then replace the faulty bettery with a new one.

($K2_4$) if both batteries are faulty, then apply the repair procedure RP2 described above.

More formally, KB(g2) contains the following facts and/or rules:

($K2_1$) Volt(S12,V_0) & V_0<2.4 $\leftrightarrow$ Faulty(S12)

($K2_2$) Faulty(C)&Voltt(S12,V_0)&($2.4 \leq V_0 \leq 3.2$) $\rightarrow$ check(D))

($K2_3$) Ok(V_1)&~Ok(V_2) $\rightarrow$ RP1(B_2, New-B),

($K2_4$) Ok(V_2)&~Ok(V_1) $\rightarrow$ RP1(B_1, New-B),

($K2_5$) ~Ok(V_1)&~Ok(V_2) $\rightarrow$ RP2(B_1, B_2, NB_1, NB_2)

We could employ $\Rightarrow$ instead of $\rightarrow$ to express defeasibility. For instance, RP1 can only be applied if there is a new battery that is kept in the store.

Test of the voltage of S12 will be performed by g1. g2 then will have to decide whether or not the finding suggests that S12 is faulty. If so, g2 has to determine whether both batteries are faulty or just one. In the latter case, there is a need to find out which one.

Suppose that, g2 begins with an information-seeking sub-dialogue wanting to know the value of the voltage of S12 as in M_1.

M_1 = <1, g2, Question, "Volt(S12,V)", 0>

After carrying out the proper test, g1 replies by stating, in M_2, that the voltage of S12 is 1.45 which g2 accepts in M_3.

M_2 = <2, g1, Assert, "Volt(S12,1.45)", 1>

M_3 = <3, g2, Accept, "Volt(S12,B_2),1.45)", 2>

g2 has now to find out whether or not one of the batteries B_1 and B_2 is faulty and which one. At this point, either of the agents can make a next move that concerns S12. g1 suggests in M_4 that B1 be repaced by a new battery. g2 rejected the proposal in M_5 and replied to g1's challenge posed in M_6 by presenting, in M_7, logic proofs P1, P2 and P3 which argue that there is a need for further testing. g1 accepts g2's argument. The logic proofs are as follows:

P1: Given Volt(S12),1.45, we can infer that it cannot be the case that both batteries are ok.,

P2: if it is established (possibly via test) that one of the batteries is not working normally then we cannot infer that the other is.

P3: if it is established (possibly via test) that one of the batteries is working normally, then we can infer that the other is not.

The moves M_4- M_7 are as follows:

M_4 = <4, g1, Assert, "RepB_1", 3> where

RepB_1 = (Volt(S12),1.45) $\rightarrow$ RP1(B_1, New-B))

M_5 = <5, g2, Reject, "RepB_1", 4>

M_6 = <6, g1, Challenge, "U(RepB_1)", 5>

M_7 = <7, g2, Assert, "P1, P2 and P3", 6>

M_8 = <7, g1, Accept, "P1, P2 and P3", 7>

In further information-seeking sub-dialogue g2, in M_9, asks g1 for the voltage of B_1 alone. g1 replies, in M_{10}, that the voltage of B_1 is 1.3. Using its argument, P3, g2 can now infer that B_2 is faulty. Hence, g1 could now apply the procedure RP1(B_2, New-B) to replace B2 with a new battery.

M_9 = <9, g2, Question, "Volt(B_1,V_1)", 8>

M_{10} = <9, g1, Assert, "Volt(B_1,1.3)", 9>

M_{11} = <11, g2, Accept, "Volt(B_1,1.3)", 10>

M_{12} = <12, g1, Assert, "RP1(B_2, New-B)", 11>

M_{13} = <13, g2, Accept, "RP1(B_2, New-B)", 12>

7. Related Work

Little consideration has been given to dialogue in work on distributed systems and multi-agent PS. In [18], an abstract formal model of CPS is presented. In [16], it is shown how the concept agent can be used to realize a multi-agent system for distributed diagnosis. Most existing spoken dialogue systems focus on simple and constrained tasks. There has been other work on modelling dialogue for complex task domains as in [1, 3]. TRIPS [3] is a distributed, agent-based cooperative dialogue system where its components act asynchronously. The use of the notion of PIS in our system is compatible with that of information state used in [6]. However, PIS is partial and supported by an NML. A method to merge conflicting PIS's based on their preference-based argumentation framework is proposed in [5]. In our proposal, arguments may be built from the PIS of one agent and an appropriate subset of that of another.

8. Concluding Remarks

In this paper, we have made a first step towards developing a formal model of dialogue and argumentation for a multi-agent CPS. It builds on work on argumentation presented in [2] and proposes a logical framework for dialogue need in CPS. It addresses most of the types of dialogue relevant for collaboration which are classified in [17]. We have discussed the notion of CPS and discuss some of the related communication issues. We have proposed a (PIS)-based framework for dialogue and argumentation. We have employed a three-valued based NML for representing and reasoning about PI. We have shown via an example that the system can handle CPS tasks.

10. References

[1] J Allen, D Byron and M Dzilovska, Towards Conversational Human-Computer Interaction, *AI Magazine*, 22(4), 2001, pp.27-37.

[2] L Amgoud and N Maudet, Strategical Considerations for Argumentative Agents, In Proc. of 10th Int. Workshop on Non-Monotonic Reasoning, 2000, pp. 409-417.

[3] N Blaylock, J Allen and G Ferguson, Synchronization in an Asynchronous Agent-Based Architecture for Dialogue Systems, In Proc. of 3rd SIGdial Workshop on Discourse and Dialog, 2002, pp. 1-10.

[4] M Colombetti, A Commitment-Based Approach to Agent Speech Acts and Conversations, In: 4th Int. Conf. On Autonomous Agent, 2000, pp. 21-29.

[5] T Finin, R Fritzson, D Mackey and R McEntire, KQML as an agent Communication Language, In: *Proc. of 13th Int. Conf. on Information and Knowledge Management,* ACM Press, 1995.

[6] S Larsson and D Traum, Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. Natural Language Engineering, 6, 2000, pp. 323-340.

[7] P McBurney and S Parsons S., A Denotational Semantics for Deliberation Dialogues. Lecture Notes in Artificial Intelligence 3366, 2005, pp. 162-175.

[8] P McBurney and S Parsons, Dialogue Games for Agent Argumentation, In: I Rahwan, G Simari (eds.), *Argumentation in Artificial Intelligence,* 2009, pp. 261-279.

[9] A Moubaiddin and N Obeid, The Role of Dialogue in Remote Diagnostics, 20th Int. Conf. on Condition Monitoring & Diagnostic Engineering Management, 2007.

[10] A Moubaiddin and N Obeid, Dialogue and Argumentation in Multi-Agent Diagnosis, In: *New Challenges in Applied Intelligence Technologies*, N T Nguyen and R Katarzyniak (Eds.), Studies in Computational Intelligence 134, Springer, 2008, pp. 13- 22..

[11] A Moubaiddin and N Obeid, Partial Information Basis for Agent-Based Collaborative Dialogue, Applied Intelligence, 30(2), 2009, pp. 142-1678, **DOI [10.1007/s10489-007-0108-5]**

[12] N Obeid, Three Valued Logic and Nonmonotonic Reasoning, Computers and Artificial Intelligence, 15(6), 1996, pp. 509-530.

[13] N Obeid, A Model-Theoretic Semantics for De*fault Logic*, *WSEAS Transactions on Computers*, 4(6), 2005, pp. 581-590.

[14] N Obeid and R B K N Rao, Diagnostic Temporal Reasoning in Model-Based Diagnosis (MBD) of Dynamic Systems, Int. J. of COMADEM, 7(1), 2004, pp. 13 – 28.

[15] O Rana and K Stout, 2000, What is Scalability in Multi-Agent Systems? In: Proc. 4th Int. conf. on Autonomous Agents, 2000.

[16] M Schroeder and G Wagner, Distributed Diagnosis by Vivid Agents, In: Proc. of 1st international conference on Autonomous agents, 1997, pp. 268 – 275. .

[17] D Walton and E Krabbe,Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning, 1995.

[18] M Wooldridge and N Jennings, Cooperative Problem Solving, Proc. Modelling Autnomous Agents in Multi-Agent World, 1995.

[19] M Zhang, Q Bai, F Ren, and J Fulcher, 2009, Collaborative Agents for Complex Problems Solving, In: C.L. Mumford and L.C. Jain (Eds.): Computational Intelligence, ISRL 1, pp. 361–399. Springer.