Software Agents: Can we Trust Them?

Eugénio Oliveira
LIACC/DEI, Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: eco@fe.up.pt

Abstract—In this paper we briefly describe multi-agent system architectures that are proved to be suited to address problems which simultaneously are of a Distributed, Decentralized and Dynamic nature. Two clearly different scenarios are used to show how the multi-agent system paradigm can be used to cope with realistic situations which are either mainly cooperative or mainly competitive and, thus, asking for specific characteristics that make those systems trustworthy. Negotiation protocols are used for the sake of getting, through cooperative strategies involving several agents, commonly accepted final solutions for an overall goal. We also advocate the use of Trust measures, together with normative environments that enforce legality, to make multi-agent systems useful in real demanding scenarios, like it is the case in competitive B2B situations.

I. INTRODUCTION

The Agents’ metaphor has been in use for a few decades pretending to mimic autonomous entities that coordinate together to achieve either competitive or cooperative goals. In this paper we take the perspective of using agents as delegates of both human users and Enterprises that try to realize up to what extent can they be confident on establishing fruitful relationships with each other for pursuing their respective goals. Most of the complex problems that arise, can be classified as belonging to the so called 3D class of problems. They reflect a reality that simultaneously is of a Distributed, Decentralized as well as Dynamic nature. This means that not just sensors and data inputs, plus effectuators and actions outputs, are disperse at different points (Distributed) but also decision-making can be, at least partially, taken at different nodes of the system (Decentralized). Moreover, the system trying to solve the overall problem at stake, has to deal with a changing, evolving reality (Dynamic). Agent-based systems intend to provide a suitable architecture to cope with such problems. However, if we don’t give the designer the burden to find out the best configuration of available agents that, for each specific situation, will be able to reach the solution, the problem that arises is: How can an agent find out, in runtime, its best partners for the task ahead and why should it trust them? Finally, and most important, how to coordinate joint work together with the others?

We here propose solutions for these issues, either by simply using multi-agent cooperative policies or, for more competitive situations, based on the development of institutional environments that can monitor agent actions as well as providing tools that agents can use to estimate other agents trustworthiness.

The remaining sections of this paper are as follows: Next we define a few basic concepts, then we introduce a cooperative scenario followed by the multi-agent based kind of solution to the problem at stake. We then introduce a competitive scenario and discuss the need for a more sophisticated approach. Normative environment in action and computational Trust models are then briefly presented and the final section states a few conclusions.

II. CONCEPTS

A. Agents

Agents are the natural metaphor for implementing active nodes in a network. For the sake of simplicity, we here stick with the base-line definition of Agents given by Wooldridge and Jennings [1]: ”a hardware or (more usually) a software-based computer system that enjoys the following properties: autonomy - agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state; social ability - agents interact with other agents (and possibly humans) via some kind of agent-communication language; reactivity: agents perceive their environment and respond in a timely fashion to changes that occur in it; pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking initiative.”

Furthermore, whenever researchers want to explicitly refer more specifically to intelligent agents, they use so called mentalistic notions like knowledge, beliefs, intentions, desires, goals, commitments, and obligation [2].

B. Multi-agent systems

An Agent’s goal is either to maximize its own utility or the one of the group it belongs to, if they are of a cooperative nature. Agents have a lot to benefit from engaging into successful relationships with others, leading to their utility (whatever the metric is) increase. Since organizations are defined (Malone) as coordination patterns of decision-making and communication among a set of agents who perform tasks in order to reach intended global states, multi-agent systems (MAS) appear as good candidate computer-based architectures to support them.

A formal definition of MAS can be:

$$\text{MAS} = (S, Ags, Act\_M, fa, L)$$

where:

- $S$ is a non-empty set of situations;
- $Ags$ a non-empty set of Agents
- $Act\_M$ is a non-empty set of primitive actions in MAS, such that $Act\_M \subseteq (\bigcup_{A \in Ags} Act(A))$
\textit{fa} is the function assigning to each \textit{Act} \in \textit{Act}_M the Agent in charge of executing it. 

\textit{L} is the Language expressing possible actions in MAS.

A simplest and more general definition can also be  \textit{MAS} = \textit{(Ags, Env)} where \textit{Ags} is a set of Agents and \textit{Env} a set of environment states.

In a MAS, \textit{L} always include a set of primitives for building up protocols for agents interaction. Moreover, MAS mapping organizations, include negotiation protocols backing the decision-making process that leads to specific assignment of Actions to Agents.

C. Decision Making

A credible agent, makes decisions in an informed way. In our decision-making model, agents use as primary information, evaluation of past direct observations on the target agent behaviour regarding itself. Computational Trust model aims at aggregating this direct past experience. However, in a society, a rich flow of indirect information, coming from third parties is most of the time available. It would not be wise to ignore that kind of available information. However, reputation should be parsimoniously used and should not interfere with Trust measure. Trust and reputation are two different components which the Truster-Trustee relationship takes place and other differences between the intrinsic property of trustworthiness and the images that can be projected into different assessing agents (trusters), the Trust values. We have been arguing \cite{[3],[4]} that several dimensions, mostly due to the context in which the Truster-Trustee relationship takes place and other indirect social factors, should be taken into account for a more complete perception of Trust. However, since the focus of this paper is not that particular discussion, we follow a more consensual, basic and established definition for the concept. Closely, but not completely following \cite{[5]}, a more formal definition of Trust is: \textit{Trust}(\textit{i}, \textit{j}, \phi) meaning that the \textit{Truster}(\textit{i}) Trusts \textit{Trustee}(\textit{j}) to do \textit{Action}(\alpha) leading to the achievement of \textit{Goal}(\phi) if:

\begin{align*}
&\text{(GOAL}_i \phi) \\
&\text{(BEL}_i \text{POWER}_j \phi) \\
&\text{(BEL}_i \Box (\alpha \mid= \phi)) \\
&\text{(BEL}_i \text{INTEND}_j \alpha)
\end{align*}

where \Box is the usual temporal modal operator and \textit{INTEND}_j \alpha is \textit{j}'s intention to do action \alpha according to Cohen and Levesque's well accepted theory\cite{[6]}.

E. The role of Reputation

We do not consider Reputation as a component of the Computational Trust Model. Reputation is here considered as a different information, gathered from several different agents' opinion on a specific Agent that will be considered at the same level as Trust (although, possibly, with different weights) for the decision making process. These two concepts should be kept separate since they rely in different assumptions: On one hand, those concrete and potentially verified results coming from direct interactions (Trust) and, on the other hand, those collected from third-parties, thus indirect and subject to many possible uncertainties according to these intermediate interests or perception capabilities (Reputation). Their relative importance is inverse proportional to each other, being the latter more important whenever the mutual knowledge to directly evaluate Trust is short. We still did not include Reputation in our model.

III. COOPERATIVE SCENARIO

We at LIACC\footnote{Artificial Intelligence and Computer Science Group at University of Porto} have been challenged by a rather complex and relevant problem of automatic management of plans that are, during the execution phase, subject to disruptions. Well-formed plans happen to fail due to unexpected events. This is what often happens with Airline Operations Control Centres when trying to fulfill their obligations in the presence of unexpected events, disruptions, that affect previously established flights schedule plan. The economic relevance of the resulting problem is huge and goes up to between 2 or 3\% of the airline annual revenue.

The solution we are proposing to Portuguese Airlines services, in order to bypass the current all human-based procedure, relies on a multi-agent system (MASDIMA) \cite{[7],[8]} in which agents representing different experts cooperate with each other to reach a final solution to the overall problem.

During the execution of the airline operation, several events might occur that can be responsible for irregular operations. In short, we may classify these events as those leading to \cite{[8]}:

- Flight Arrival Delay (due to enroute air traffic delay, enroute weather, enroute aircraft malfunction,flight diversion, flight departure delay);
- Flight Departure Delay (Crew delay, crew absenteeism, loading delay, passenger delay, air traffic control delay, aircraft malfunction, weather conditions and a flight arrival delay);

These unexpected events are thus responsible for scheduling problems that will affect one or several of the three involved entities: The passengers, the aircraft, the crew.

Each one of these three entities are also interested in getting a solution that minimizes their respective costs, what might be conflictuos. The best solution from the crew perspective is often different from the one that minimizes the passengers costs, for example.

This analysis, naturally leads to the possibility of trying to have at least three different agents, embedding their respective different expertise, (Figure 1), according to those perspectives mentioned before, and make them converge through negotiation to the final overall solution. Although one might believe that this agents will compete for the best solution for each one of them, the goal of the negotiation process is to reach a common solution with an acceptable utility. Decision Making is achieved through two different layers. First of all, each ”Manager Agent” asks for a possible best solution from their utility perspective. For example, the AIRCRAFT (A/C) Manager Agent asks to agents (called specialists) using different algorithms (simulated annealing, genetic, tabu search…) for possible solutions and ranks them according to their utility. The same happens for the CREW and PAX Manager Agents. Each one of them try to build up a proposal including its best partial solution and trying to acomodate it into a complete solution. They then engage together with the supervisor Agent, the agent in charge of the final decision, in a multi-round negotiation process trying to find out a complete solution in which its own utility is still acceptable (Figure 1).

At the negotiation level we are using a generalization of the Q-Negotiation protocol introduced in [9], [10]. These authors propose a negotiation mechanism in the context of agent-based Virtual Organisation (VO) formation process, which selects the optimal set of partners that satisfies the VO needs. In that scenario, each partner’s goal is to maximize its own profit and, for that, the negotiation process takes into account the rationality and self-interestedness of the participating agents.

The proposed Q-Negotiation mechanism includes a multi-attribute negotiation protocol with multiple rounds and qualitative feedback to each one of the proposals. Additionally, the agents are able to learn (adapt) their strategies for bidding along the successive negotiation rounds, due to the inclusion of a Q-Learning algorithm. On-line Q-Learning enables agents to learn in a continuous way with information extracted from each one of the feedback items given to each of the formulated proposals, and not only in the end of the process, with the negotiation final result. This feedback is a qualitative one (e.g. good, fair, bad) and regards each one of the attribute-value pairs included in the proposal compared (regarding utility) with the best competitive one so far.

In our Airlines scenario, a Monitor agent sends the problem description to the Supervisor agent, including information about the dimensions affected by the unexpected event, the problem in hands, as well as the previously calculated scheduled time and costs regarding aircraft, crew and passengers. The Supervisor agent assumes the role of organizer of the negotiation and, using that information about the problem, broadcasts a call-for-proposal (cfp) that includes a range of preferred values for delay, flight costs, crew costs, passenger costs, passenger trip time and a negotiation deadline. The cfp triggers the first negotiation round. The AirCraftManager, CrewManager and PaxManager agents (respondent agents) send proposals according to their respective interests. For example, the first one wants to minimize the flight costs and delay and the PaxManager wants to minimize the passengers trip time and cost. It is important to point out that the proposals sent by the respondent agents are based on the candidate solutions found by their specialist agents. All the proposals are evaluated by the Supervisor and qualitative feedback is sent to the respondent agents. Proposal evaluation is done by using an utility function depending on the aircraft delay, crew delay and passenger trip time as well as aircraft, crew and passenger costs for that specific proposal.

By using the Q-negotiation algorithm, each agent getting, from the supervisor, a qualitative feedback about their previous bids, select the next bid that, according to current state-action Q values table, looks the most promising for being accepted by the Supervisor agent. Negotiation ends when all but one of the agents remains in the negotiation or one of the proposals completely satisfies the supervisor preferences.

The relevance of the approach is that partial solutions are found by specialist agents that may then, through negotiation, relax their utility in order to find out a good overall solution. The way each one relaxes, depend on the current competitive situation.

In [8] a discussion is made by comparing results of this MAS approach with current human-based approach in real past disruptive situations. The final outcome shows a decrease of about 23% on the total operation costs by using the MAS approach.
IV. COMPETITIVE SCENARIOS

One of the main breakthroughs of intelligent software automation has been achieved in the B2B domains. Due to globalization and constant delocalization, big multi-national companies always try to re-design virtual consortia they belong to, whenever a new opportunity of business arises. This fact points out to the need of software systems that makes it possible finding out the best potential partners to establish temporary contracts leading to fruitful collaboration. Moreover, selecting partners, establishing and executing contracts become more trustworthy activities if encompassed into the framework of credible (electronic) institutions which are seen as the guardians of legal and fair business activities. Trust and social control are then intermingled.

We, at our LIACC research group, have been involved on mixing together three complementary research areas—automatic negotiation, normative environments and computational trust—that seamlessly glue together in order to be exploited by software agents that, while working as delegates of different stakeholders, interact in order to establish successful contractual agreements. Although detailed description of our protocols and models is not in the scope of this paper, our main argument here is that they provide the framework for agent-based trustworthy systems applied to the network-based scenarios like those for B2B relationships.

V. AGENT-BASED ANTE SYSTEM

For these B2B application scenarios, we have developed ANTE—Agreement Negotiation in Normative and Trust- enabled Environments[11], a software system addressing the issue of multi-agent collective work including both negotiation as a mechanism for finding mutually acceptable agreements, and the normative enactment of such agreements. Agents as delegates (either of enterprises or individuals) are supposed to submit to social norms of the institutional environment as well as to use, and progressively update, the available Trust measures to help in the partners selection process. An evaluation of the enactment phase also improves the chances for better future negotiations.

With a strong automation perspective, the scenario envisages the use of software agents negotiating on behalf of their principals, who are interested in work jointly (for the sake of simplicity suppose they are just buyers and suppliers) in a B2B network. Negotiation is therefore used to select, among a group of potential partners (e.g. suppliers), the best ones to fit a particular business opportunity. Contracts resulting from successful negotiations are validated, registered and digitally signed, before being handed to the normative environment for monitoring and enforcement purposes. Finally, the way agents enact their contracts provides important information for trust building and updating. A repository of agents’ trust and reputation information may then complete the cycle by providing relevant inputs for future negotiations. The integration of all these stages is depicted in Figure 2.

Important synergies are obtained from the integration of the three main research domains identified in Figure 2. Negotiation is informed by trustworthiness assessments of negotiation participants, in one of three possible ways: using trust for preselecting the partners with whom to negotiate; evaluating negotiation proposals taking into account the trustworthiness of proposal issuers; or exploiting trust information when drafting a contract with a selected supplier, e.g. by proposing a (more or less severe) sanction in case the supplier breaches the contract, thus trying to reduce the risk associated with doing business with less trustworthy agents.

Connecting the monitoring facility of the normative environment with a computational trust engine means that we can use contractual evidences regarding the behavior of agents when enacting their contracts to build trust assessments. Our approach to model contractual obligations allows for a rich set of possible contract enactment outcomes (fulfillments, delays, breaches, and so on), which in turn enables a trust engine to weight differently the possible sub-optimal states that might be obtained [13].

A. Sinalpha Trust Model

In this paper we want to mainly highlight the capability of dynamically building up Trust images of other agents and how can they be useful for making the MAS more useful and reliable for users.

Several different Computational Trust Models (CTR) have been proposed in the past. A few examples are [14], [15], [16], [17], [18]. The CTR model already implemented in the heart of ANTE software system, which has been implemented as a JADE-based FIPA-compliant platform, outperforms many of those mentioned before. Experiments are reported in [18], [4], [19], [20].

We have been concerned with the performance of our proposed model when the trust evidences available about a given target agent are scarce and heterogeneous regarding different situations and contexts.

Although still under investigation, a stable version of Sinalpha model includes:
The aggregator component, which is responsible for aggregating the available trust evidences of an agent into a trustworthiness score for this agent. By using a sigmoid-like function we believe we are able to capture the dynamics of Trust.

The Contextual Fitness component, which tunes the outcome of the aggregating step by taking into consideration the specificities of the current business opportunity and the adequacy of the target agent to the specific situation under assessment. A full description can be found in [21].

VI. IMPLEMENTATION

There are three kinds of agents running in the platform: those that provide contracting services, namely negotiator, computational trust, ontology mapping, notary and normative environment; external agents whose role is to make a connection to real-world contract enactment events (e.g. deliveries, payments); and users of the system, representing buyers and suppliers.

Using an appropriate graphical user interface (GUI), a buyer can specify its needs and configure a multi-attribute negotiation to take place using the platform’s negotiator service. Options include how trust is to be used in each of the negotiation steps. Also, the buyer may indicate the type of contract that is to be created, should negotiation succeed; norms governing specific contract types are already available in the normative environment, thus making it easier to establish a contract.

The supplier’s GUI displays the negotiations in which it has participated, together with the messages exchanged during those negotiations.

The GUI for the negotiator service (Figure 3, top) shows the evolution of the proposals exchanged during a negotiation protocol with several different suppliers, during a multi-round negotiation, in terms of their utility for the buyer who started the negotiation process.

Both buyers and suppliers include in their GUI a list of the contracts they have already established and a set of events related to their enactment. These events are automatically reported by the normative environment in the contract monitoring phase (Figure 3, bottom).

The GUI for the computational trust service (Figure 4) allows us to inspect how trustworthiness assessments are being computed, including the contractual evidences that are used as input for each agent. It also allows us to choose the mapping method that associates different weights of the trust engine algorithm to each of the possible contract enactment outcomes. Figure 4 also shows us that the winner agent, after 23 contracts had got a high Trust value which is also responsible for being selected for the next contract.

VII. CONCLUSIONS

Distributed, decentralized systems for dynamic environments are well represented by Multi-agent system architectures. Both for cooperative and competitive scenarios, relevant features like negotiation capabilities, normative control and trust-based decisions are ingredients that make MAS the best suited to deal with real world application problems. We have here briefly introduced two MAS-based systems - MASDISMA for a cooperative problem-solving scenario and ANTE framework for a competitive scenario, that are being tested for domains in which agreement technologies are of great interest. Our main arguments, explicitly stated all along this paper, are twofold: First, by using negotiation, agent-based systems tend to reach good agreements for cooperative environments. Second, by using Trust models (CTR) to guide partners selection under the control and supervision of normative environments, agent-based systems can also be trusted to be used for competitive environments. Deeper discussion on CTR is outside the scope of this paper.

ACKNOWLEDGMENTS

I strongly acknowledge my current or previous PhD students, Antonio Castro, Joana Urbano, Pedro Neto, Henrique Cardoso
and Ana Paula Rocha, that, for several years, worked harder on both scenarios and systems briefly presented in this paper.

REFERENCES


