Coherence and Robustness in a Disturbed MAS

Quang Anh Nguyen Vu, Benoit Gaudou, Richard Canal, Salima Hassas

Coherence and robustness in a disturbed MAS

Quang Anh NGUYEN VU*†‡, Benoit GAUDOU*†, Richard CANAL*† and Salima HASSAS‡
* UMI 209 UMMISCO, Institut de Recherche pour le développment (IRD), Bondy, F-93143, France
† MSI, Institut de la Francophonie pour l'Informatique (IFI), Ha Noi, Viet Nam
Emails: nguyenvu.quanganh@yahoo.com, benoit.gaudou@alumni.enseeiht.fr, richard.canal@auf.org
‡ Laboratoire LIESP, université Claude Bernard Lyon 1, France
Email: hassas@bat710.univ-lyon1.fr

***Abstract*—This paper presents a study on how to ensure the coherence of a distributed information system in which information is collected by a multi-agent system with the hypothesis that some agents of the system are dissonant, i.e. can produce or communicate incorrect information.**

***Index Terms*—Multi-Agents System, trust, coherence, robustness, TrustNet, mapping.**

I. Introduction

We consider a multi-agents system (MAS) in which each mobile agent aims at obtaining the most precise representation of its environment by collecting information directly via sensors or indirectly via communication with other agents. We assume that some agents disturb the system by transmitting false or inaccurate information about the environment either because their perception is flawed (*i.e.* their sensors are awry or inoperative) or because their interest goes against the community's one and encourages them to lie.

In this article, we study ways to ensure the coherence (*i.e.* the adequation between the agents' environment representation and the actual environment) and the robustness (*i.e.* the agents' capacity to adopt strategies allowing them to obtain this coherence despite the perturbed communication) of such a disturbed distributed information system. To limit the influence of agents transmitting incorrect information, we propose to use the social concepts of trust and reputation. TrustNet [1] appears to be a promising way to allow each agent to build its own network of trust. Evaluating the TrustNet before making interactions can help one agent to compute the trustworthiness of its partners and thus to decide which reliable partner to interact with. The main aim of this research on robustness is to provide tools to agents and MAS to define an efficient communication strategy.

This paper is organized as follows. Section II introduces preliminary definitions used in the whole paper: cognitive dissonance, disturbance, trust. An example of application is detailed in Section III. Section IV introduces the basic ontology and details the various kinds of information which are stored and exchanged by agents. The study of information exchanges between agents and their impact on stored data is presented in Section V before describing a method to compute the reliability of information in Section VI. Section VII presents some preliminary results on a simplified model using trust tables. Finally Section VIII concludes this paper and presents the future planned works.

II. Preliminary definitions

A. Cognitive dissonance and disturbance

In his cognitive dissonance theory, Festinger [2] considers that two elements of cognition (which includes perceptions, mental attitudes and behaviors) are dependent if there exists a link between their object, or independent, otherwise. Two dependent cognitions are called '*consonant*' when one involves or supports the other. Conversely, two cognitions are called '*dissonant*' when one involves or supports the opposite of the other [3]. The theory assumes that agents in a dissonant state will try to reduce this dissonance, for example by changing or forgetting some mental attitudes.

By applying Festinger's theory to MAS context, we call '*dissonant act*', any action performed against the achievement of the community's objective and '*dissonant agents*', the ones who perform these dissonant acts, *i.e.* agents giving false or inexact information in our context. A piece of information is called *dissonant* with relation to a set of information, when one element of this information set and the new piece of information are inconsistent. In that case, we say that both information are *dissonant*. Moreover we call '*disturbed multi-agents system*' a system that includes one or several dissonant agents. By applying the cognitive dissonance assumption described above, we assume that the community purpose is to reduce the dissonance, for example by eliminating false or incoherent information or by avoiding to interact with dissonant agents.

B. Trust

In order to reduce dissonance, we give to agents the ability to reason about other agents and to choose with which they want to interact. For this purpose one of the most efficient tools is to introduce concepts of trust and reputation. We use trust in order to identify and isolate untrustworthy agents, to evaluate an interaction utility and to decide whether and with which to interact [4]. In the sequel we refer to the following definition of trust and reputation because it appears to be general and complete enough to take into account several existing models of trust: "Trust is a measurable level of the subjective probability with which an agent A assesses that another agent B will perform a particular action in a favorable way to A, both before A can monitor such action (or independently of its capacity ever to be able to monitor it) and in a context in which it affects its own action" [5]

Moreover Marsh [6] has proposed one of the first trust models. His model takes into account only direct information to compute trust. Up to now lot of trust and reputation models have been published [7]. Our work is based on the model suggested by Schillo *et al.* [1]. In this model, agents communicate factual information but also trusts they have in other agents. Agents can thus build a network of trust values transmitted by witnesses, called the 'TrustNet'. The final trust value of an agent towards another one is thus an aggregate of direct experiences and testimonies.

III. Ongoing example: danger mapping

In this paper we consider the following example, that we will refer to as the 'Mapping example' in the sequel. Let a swarm of perfectly localized robots modeled by agents patrolling in an urban zone affected by a natural disaster. The objective of each agent is to build the most complete, precise and reliable map of the environment by using least resources possible. In particular, its job consists in detecting dangerous places and their danger level represented by a color. To map a territory, robots can detect directly the state of one zone thanks to their sensors. They can also communicate with other robots to exchange knowledge necessary to build the danger map. We assume that each robot has a local perception and that the communication in the system is also local. Among agents, we assume that some can transmit false or inaccurate information.

In the sequel, we assume that agents map the territory as a grid. Coordinates (x, y) of each patch are thus integers. We also use following notations: $AGENT$ denotes the set of agents and $COLOR$ the set of colors that robots can assign to a place depending on its danger level.

IV. Information

Among information that agents store into their memory and communicate to other agents, we distinguish two kinds of information: information about the environment and information about agents. We also split the former in two sets depending on the information origin.

A. Information about the environment

1) Direct information: The direct information set, noted D_X, represents information collected by agent X via its sensors.

In the Mapping example, we denote direct information:

- $< (x, y), color >$ an explored zone $(x, y) \in \mathbb{Z}^2$, with a danger level coded $color \in COLOR$;
- $< \neg(x, y) >$ a location (x, y) without danger.

For instance, the set $D_A = \{< (x_1, y_1), blue >\}$ means that agent A has detected a dangerous $blue$ level zone on location (x_1,y_1).

2) Indirect information: The indirect information set, noted IND_X, represents information that agent X receives from other agents. An element of indirect information is composed of two parts: a first one representing information about the environment that is domain-dependent (coordinates and color in the Mapping example) and the list of agents by which information have traveled since they have been collected from the environment. We call this list the '*information path*' and the first agent which has collected it, the '*information source*'.

In the Mapping example, we note indirect information:

- $< (x, y), color, [A_1...A_n] >$ with $(x, y) \in \mathbb{Z}^2$, $color \in COLOR$, $A_i \in AGENT$.
- $< \neg(x, y), [A_1...A_n] >$a location (x, y) without danger transmitted through $A_1...A_{n-1}$ from the source A_n.

So, the set $IND_A = \{< (x_1, y_1), blue, [C, D] >\}$ means that agent A has received an information transmitted by C (C got this information from D before) about a dangerous $blue$ level zone on the location (x_1, y_1).

B. Information about agents

Agents store trust on other agents in a graph called TrustNet. TrustNet [1] is a directed graph where nodes represent agents and edges carry information about agents' trust estimation. We note a TrustNet built by the agent X as $TN_X =< \{Node_X\}, \{< Arc_X, Value_X >\} >$. An example of TrustNet is proposed in Figure 1: this graph has been built by agent A, nodes represent the three agents A, B, C, α is the trust of A towards B and β the trust of B towards C communicated to A by B. It can be formally represented by: $TN_A =< \{A, B, C\}, \{< AB, \alpha >, < BC, \beta >\} >$. We consider that agents have an *a priori* trust value in other agents and that trusts have values in $[0, 1]$. Given this TrustNet, A can evaluate the trust in all connected agents.

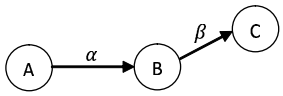

Figure 1. Example of TrustNet

V. Information exchange between agents

After the previous static description of the various kinds of information manipulated by agents, we address in this section the issue of their dynamics due to interactions between agents, *i.e.* how these information are stored and how they alter previous stored information. In a disturbed MAS, we consider that agents may distort every kind of information except the information path because this data can be obtained from the system.

When two agents meet, they share information on the environment and information on the agents. When an agent receives information from another one, it compares each received information with its own information to adjust the confidence it has in this agent. We distinguish three kinds of compared information that we can relate to Festinger's distinction on cognitions: *independent information*, *i.e.* information that are incomparable (in the Danger Mapping example, this includes information about different coordinates); *consonant information*, *i.e.* information that are dependent (*i.e.* same coordinates) and provide the same information (*i.e.* same color); *dissonant information*, *i.e.* information that are dependent

(*i.e.* same coordinates) and provide different information (*i.e.* different colors).

In the sequel, we consider two agents A and B exchanging information. We only study A's side of the interaction because this interaction is symmetric. The process described below is the same for the agent B.

A. Update of information about the environment

We consider that A meets B at the instant t. $(D_X)_t$ and $(IND_X)_t$ represent sets of direct and indirect information of X at this time point.

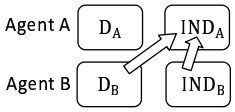

Figure 2. Influence of B's information on A's information sets

Agent A merges information received from B with its own information and produces updated information sets $(D_A)_{t+1}$ and $(IND_A)_{t+1}$. We argue that B's direct and indirect information sets do not alter D_A because D_A represents the information collected directly from the environment. To compute $(IND_A)_{t+1}$, agent A merges all information received from B with its own existing indirect information set by a fusion function Ψ^{Data}. Figure 2 illustrates these influence relations. We thus have:

- $(D_A)_{t+1} = (D_A)_t$
- $(IND_A)_{t+1} = \Psi^{Data}((IND_A)_t, (D_B)_t, (IND_B)_t)$

To highlight main aspects of function Ψ^{Data}, here is a simple example of information exchange, with $(D_A)_t = \{< (x,y), blue >\}$, $(IND_A)_t = \{< (x,y), red, [C,D] >\}$, $(D_B)_t = \{< (x,y), yellow >\}$, $(IND_B)_t = \{< (x,y), blue, [F] >\}$. As explained above, D_A remains unchanged : $(D_A)_{t+1} = \{< (x,y), blue >\}$. Now, A will fusion received information from B with its existing indirect information set. In this simple example, all received information become indirect information for A. It should indicate that B transmitted them. It thus transforms B's direct information in A's indirect one by adding to them a new path initialized with agent B. It moreover adds B to the path of received indirect information. It thus have: $(IND_A)_{t+1}$ = {*<(x,y),red,[C,D]>*, *<(x,y),yellow,[B]>*, *<(x,y),blue,[B,F]>*}.

B. Update of information about agents

Besides exchanging information about the environment, agents also share information on agents, *i.e.* their TrustNet. The update of A's TrustNet will be computed in two steps: the update of its trust α in B and the update of its own TrustNet using a fusion function $\Psi^{TrustNet}$ taking as parameters both TrustNets $(TN_A)_t$ and $(TN_B)_t$ and the trust α: $(TN_A)_{t+1} = \Psi^{TrustNet}((TN_A)_t, (TN_B)_t, \alpha)$.

1) Step 1 - Computing trust by comparing information: To compute its trust in B, agent A makes comparisons between its own information and the ones transmitted by B. The comparison occurs in 4 steps : ① $D_A \rightarrow D_B$ (i.e comparison between D_A and D_B), ②$D_A \rightarrow IND_B$, ③ $IND_A \rightarrow D_B$, ④ $IND_A \rightarrow IND_B$ corresponding to the four arrows of the Figure 3. The result of these confrontations will increase A's trust in B (if B has transmitted more consonant than dissonant information), decrease it (if it has brought more dissonant than consonant information) or let it unchanged otherwise.

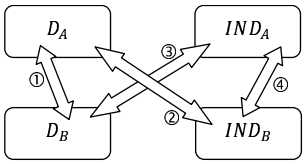

Figure 3. Comparing agents databases

2) Step 2 - Merging two different TrustNets: In a second stage, A builds the updated TrustNet $(TN_A)_{t+1} =< \{Node_*\}, \{< (Arc_*, Value_*) >\} >$ from $(TN_A)_t$ and $(TN_B)_t$. We can write $Node_* = Node_A \cup Node_B$ and $(Arc_*, Value_*) = \Psi^{ArcValue}((TN_A)_t, (TN_B)_t, \alpha)$ where $\Psi^{ArcValue}$ merges values on the arcs of both graphs, considering the trust A has on B. Due to space limitation, we do not detail this function here.

VI. Computing the information reliability

Via communication, agents receive a lot of information. Among these information, some can be false or inaccurate. Agents have thus to make a selection among this flow and determine which are reliable and which are not. In order to calculate the information reliability, each agent uses a probability tree to represent information in its memory. The advantage is that its use is simple and effective. In the Mapping example, each agent uses a tree such as the one in Figure 4 to store the information about one zone. Each node stores an elementary information. T and F represent respectively the existence and the non-existence of danger; $black, blue, yellow, red$ belongs to $COLOR$ and represents the set of danger levels of the zone.

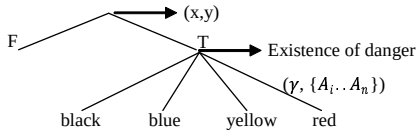

Figure 4. Information tree

We associate to each edge of the tree the pair $(\gamma, \{A_i...A_n\})$, where γ is the probability of the information to be true (its reliability) and $\{A_i...A_n\}$ the information sources. Each time a new information comes, the agent updates these values in the tree.

A major issue is to determine the information reliability when two groups of agents give different information about the same patch, *i.e.* to calculate the reliability of dissonant information. We need to compare the impact of both groups. For this purpose, we assign a trust weight TW to each group depending on individual trusts as follows. We introduce two

thresholds: an upper threshold (Upp) and a lower threshold (Low). Between Upp and 1, agents are regarded as "reliable". Between Upp and Low, agents are under observation. Between 0 and Low, agents are regarded as "unreliable".

To evaluate the trust weight of an agent's community, we first split the community in three sets (reliable, unreliable and others). A weight is assigned to each member of a set (for example an arbitrary $+\lambda$ for one reliable agent, -1 for one unreliable agent, $+1$ to one other agent). Then, considering the cardinal of these sets, we calculate the balanced sum:

$$\begin{aligned} &TW(A_1, ..., A_n) = \\ &\lambda * Card(\{A_i | i \in [1,n], Upp \leq Trust(A_i)\}) \\ &+ Card(\{A_i | i \in [1,n], Low < Trust(A_i) < Upp\}) \\ &- Card(\{A_i | i \in [1,n], Trust(A_i) \leq Low\}) \end{aligned}$$

where $Trust(A_i)$ is the trust in agent A_i calculated according to the TrustNet. This computation method aims at creating an equilibrium between the number of quality of agents in a group and their quantity. The Trust Weight expresses the weight of the group on the reliability of information transmitted by this group in relation with the trust weight of other groups. Formally, we note p_i a piece of information brought by a group G_i of $G_1, ..., G_n$, the set of groups having transmitted a dissonant piece of information. We compute the reliability of a piece of information as follows:

$$reliability(p_i) = \frac{TW(G_i)}{\sum_{k \in [1,n]} TW(G_k)}$$

VII. Some Results

Preliminary simulations under NetLogo [8] have been proceeded on a simplified model of our Mapping application to test the interest of introducing trust to improve the quality of the communication in a disturbed Multi-Agents System. This simplified version deals with danger zones without color codage. It does neither use TrustNet to represent trust, nor makes the distinction between direct and indirect information. Agents only transmit environmental information and trust tables, *i.e.* a table in which agents store trust values on other agents. When an agent B's trust value falls below the Low threshold in the trust table of an agent A, A does not consider anymore information coming from B. Moreover, A doesn't send any more information to B.

We made some simulation and this strategy appears to have an impact on the volume of communications and on the number of steps the system goes to reach its objective: get a complete and trustful map of the environment. In Figure 5, we compare the number of meetings which is also the number of communications in the MAS if there is no communication policy and the number of sent messages when each agent interrupts communication with untrustful agents without impairing the performance of the MAS. These results are significant and encourage us to develop a more accurate model using no more trust tables but TrustNets and a distinction between direct and indirect information.

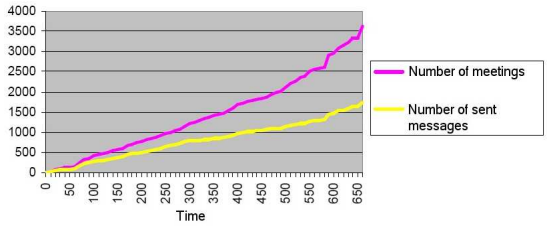

Figure 5. Comparison between the number of sent messages (yellow) and the number of meetings (purple)

VIII. Conclusion and Future Works

This article addresses the problem of dissonance in a disturbed MAS where information collection or transmission can be altered by unreliable agents. We show how each agent associates a reliability to information and a trust to other agents to improve its perception of the world. The advantage of our approach is that each agent can distinguish direct (collected on the environment) and indirect (collected by exchanging information with other agents) information, not only in its stored data, but also in the data transmitted by agents it meets. Considering these information, agents can build and update finely tuned information probability trees and more accurate TrustNets. Actually, variations of one agent's confidence in another agent depend at the same time on the quality of information transmission by the other agent and on the distance between information gathered by both agents. This process finally helps an agent to enforce its communication with trusted agents and to reject communication from untrusted agents, so to reduce dissonance in the disturbed MAS.

Our work is focusing on the best way to merge information in the direct and indirect databases stored by an agent and to compute trust variations on the arcs of the TrustNet based on the confrontation of information.

Future work will focus on the MAS self-organization about communication management, the structuring of agents communities according to their reliability and the limits of dissonance a disturbed MAS can support.

References

[1] M. Schillo, P. Funk, and M. Rovatsos, "Using trust for detecting deceitful agents in artificial societies," *In Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*, vol. 14, no. 8, pp. 825–848, September 2000.

[2] L. Festinger, *A Theory of Cognitive Dissonance*. Stanford University Press, June 1957.

[3] P. Pasquier and B. Chaib-draa, "Cohérence et conversations entre agents : vers un modèle basé sur la consonance cognitive," in *Actes des JFIADSMA'02*, 2002, pp. 188–203.

[4] G. Muller and L. Vercouter, "Computational trust and reputation models," in *10th European Agent Systems Summer School*, 2008.

[5] T. D. Huynh, "Trust and reputation in open multi-agent systems," Ph.D. dissertation, University of Southampton, 2006.

[6] S. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, University of Sterling, 1994.

[7] J. Sabater and C. Sierra, "Review on computational trust and reputation models," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 33–60, September 2005.

[8] U. Wilensky and I. Evanston, "Netlogo. center for connected learning and computer based modeling," Northwestern University, Tech. Rep., 1999.