



HAL
open science

Using trust and cluster organisation to improve robot swarm mapping

Quang Anh Nguyen Vu, Benoit Gaudou, Richard Canal, Salima Hassas, Frédéric Armetta, Serge Stinckwich

► To cite this version:

Quang Anh Nguyen Vu, Benoit Gaudou, Richard Canal, Salima Hassas, Frédéric Armetta, et al.. Using trust and cluster organisation to improve robot swarm mapping. Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN) in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 2010, Taipei, Taiwan. hal-00538104

HAL Id: hal-00538104

<https://hal.science/hal-00538104>

Submitted on 21 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using trust and cluster organisation to improve robot swarm mapping

Quang Anh NGUYEN VU^{*†‡}, Benoit GAUDOU^{*†}, Richard CANAL^{*†}, Salima HASSAS[‡], Frédéric ARMETTA[‡]
and Serge STINCKWICH^{*†}

^{*} UMI 209 UMMISCO, Institut de Recherche pour le développement (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.enseiht.fr, richard.canal@auf.org,
serge.stinckwich@info.unicaen.fr

[‡] Laboratoire LIESP, université Claude Bernard Lyon 1, France

Emails: hassas@bat710.univ-lyon1.fr, farmetta@bat710.univ-lyon1.fr

Abstract—In a distributed spatialized information collecting system managed by a swarm of elements some of which are supposed disturbed, maintaining system coherence and cooperation between reliable elements is a challenge. This paper tackles the problem of finding an efficient mechanism to ensure the coherence of the system and to optimize system performance. The main contribution of this paper consists of two major steps: (i) *use trust-based mechanism to ensure the coherence and the robustness of the system*; (ii) *allow reliable elements to create dynamic clusters based on trust*. We propose two different organizations in order to manage these issues and show how they must interact: a social one in which each element maintains a TrustSet to estimate trust on others; a spacial one in which reliable elements are grouped in an ad hoc type network to improve cooperation between themselves.

Index Terms—Multi-Agents System, Mobile Ad Hoc Network, Trust, Coherence, Robustness, Connectivity Maintenance.

I. INTRODUCTION

Let's consider as an example a swarm of potentially defec-tuous mobile robots that have to collaboratively map a zone affected by dangers. Robots can collect information from the ground and exchange their data with other agents. Perception and communication are supposed range limited. The system could be represented as a multi-agents system (MAS) in which each agent (representing a robot) aims at obtaining the most precise representation of its environment by collecting information directly (*e.g.* via sensors) and indirectly (*e.g.* via communication with other agents). Assuming that some agents disturb the system by transmitting false or inaccurate information about the environment because of their flawed perception (*e.g.* their sensors are awry or inoperative), we study ways to improve the coherence of the system (*i.e.* adequation between the agents' environment representation and the real environment) and its robustness (*i.e.* the agents' capacity to adopt strategies allowing to maintain this coherence despite the disturbed communication system) [1], [2]. To limit the influence of unreliable agents, we intend to give agents the capacity to dynamically build a two layers auto-organization:

(i) a social organization: each agent builds a personal data structure called 'TrustSet' on which it computes its own trust evaluation toward other agents in the system. The TrustSet

is mainly composed of a TrustGraph and a TrustTable. Both structures are updated by using direct and indirect interactions between agents. While interacting, the model of trustworthiness is refined and used to appreciate the reliability of other agents in order to reject undesirable communication. Trust is also used to evaluate the reliability of gathered information.

(ii) a spatial organization: in a second step, agents that are spatially close and recognized as reliable organize themselves into physical clusters (*e.g.* ad hoc networks). This organization allows a better coordination between agents. Optimization of system performance (in terms of exploration time, energy saving) is achieved by avoiding situations such as the overlapping of exploration areas (*e.g.* multiple agents exploring the same target) and by reducing communication exchanges between agents belonging to the cluster.

The management of group communication based on the concept of cluster appears to be a promising way due to benefits it brings to the system:

- a better agents cooperation: cluster can improve the moving strategy of agents by sharing tasks to complete their work faster.
- a minimization of data redundancies induced by competing instead of collaborating agents.
- information exchanged percolates quickly in a stable cluster where perturbation is minimum.

After the introduction, this paper is organized as follows : Section II presents spatial and social organization. The notion of dynamic cluster is introduced in Section III before examining its dynamics. Interaction between spatial organization and social organization is analysed in Section IV. Finally Section V presents the conclusion and future research.

II. SOCIAL / SPATIAL ORGANIZATION - LOGIC / PHYSICAL LAYER

As presented above, the main contribution of this article is to propose a mechanism for agents to build coalitions [3], [4] that we name clusters. A cluster can be regarded as a type of emerging ad hoc network where nodes decide to join together on the basis of different criteria. Literature defines mobile ad hoc network as mobile groups of wireless nodes

which cooperatively form a network independent of any fixed infrastructure or centralized administration [5], [6]. Routers are free to move randomly and organize themselves arbitrarily. Network topology may change rapidly and unpredictably [7].

We are therefore interested in the question of the auto-organization (without human intervention) of agents in a kind of meta-agent structure inside which we can maintain connectivity to ensure the existence of a reliable communication channel between its members throughout the mission. Our goal is to build the topology of a distributed information collecting system on two different layers in order to better manage communication inside groups of agents. The “logic layer” built from social organization is elaborated from the computation by each agent of a TrustSet based on communication with other agents. The “physical layer” built from spatial organization is constituted by groups of neighbouring agents trusting themselves, below named “clusters”, in which the connectivity must be maintained as in an ad hoc network.

In the sequel, we use the following notations : V denotes the set of the agents, $E(V) = \{\overline{XY} \mid X, Y \in V, X \neq Y\}$ the set of the links between the agents of V and $E'(V) = \{\overrightarrow{XY} \mid X, Y \in V, X \neq Y\}$ the set of the directed links between the agents of V .

A. Social organization

As far as social organization is concerned, each agent builds a data structure called TrustSet in which it can compute its own trust evaluation toward other agents in the system. By associating a reliability to information and a trust to members of the community, each agent improves its perception of the world. As the TrustSet is presented in details in [2], we present here only the main data structures on which it is composed: the TrustGraph and the TrustTable. The public part, the TrustGraph, is a directed valued graph which contains both direct trust values and indirect trust values. The private part, the TrustTable, is a simple table in which the agent stores the intrinsic trust values (trust values estimated from direct and indirect trust values by using typical algorithms that can differ from one agent to another one).

Definition 1. (TRUSTGRAPH). A *TrustGraph* is a weighted digraph with an origin. If $A \in V$ is the origin of the digraph, let $TG_A = (V_A, E'_A, w_A)$ the *TrustGraph* of the agent A where $V_A \subseteq V$ is a set of vertices, E'_A a set of directed edges $E'_A \subseteq E'(V_A)$ and $w_A : E'_A \rightarrow [0, 1]$ a weight function.

A TrustGraph is a directed graph without loops (i.e. paths joining a node to itself) associated to an agent A representing the set of connected agents to A (either agents it has met, either agents it has heard about when communicating). When two vertices are connected by an edge, it means that the agents represented by the vertices have met each other. For instance, if B meets C then meets A , two directed edges are added to the TrustGraph of agent A : \overline{AB} et \overline{BC} . Moreover edges carry information about agents' trust estimation. The TrustGraph is build thanks to collected or transmitted information. It will be communicated to other agents at each meeting.

Definition 2. (DIRECT TRUST). The confidence value $w_A(\overline{AX})$ assigned to an edge connecting the origin A to a node X in TG_A represents the direct trust DT_{AX} of A in agent X .

Direct Trust is computed by comparing information collected by the agent itself with information collected by the agent it meets.

Definition 3. (INDIRECT TRUST). The confidence value $w_A(\overline{XY})$ assigned to an edge connecting a node X different from the origin to a node $Y (Y \neq X)$ in TG_A represents the indirect trust IT_{XY} of agent X in agent Y .

Indirect Trust is computed from trust values obtained via communication.

Definition 4. (INTRINSIC TRUST). The intrinsic trust T_{AX} represents the trust the origin agent A computes about any other agent X taking into account its own DT_{AX} and all the trusts along the various paths linking A to X in TG_A .

Example 5. In [2], we propose a formula to compute the intrinsic trust of A in any agent X as follow:

$$T_{AX} = \frac{T_{AA} * DT_{AX} + \sum_{Y \in V_A} (T_{AY} * IT_{YX})}{T_{AA} + \sum_{Y \in V_A} T_{AY}}$$

The computation of the intrinsic trust of A in X accommodates the propagation of trust along a path and the combination of trusts from different paths. We note that $T_{AX} = DT_{AX}$ when only one edge connects A to X and T_{AA} is set to 1 (the trust of the origin agent in itself is initialized to 1 if we consider that it has no reason to have doubts about its own reliability).

Definition 6. (TRUSTTABLE). The set of intrinsic trusts of A denoted by $\{T_{AX} \mid X \in V_A\}$ is stored in a table called *TrustTable* denoted by TT_A .

The TrustTable is computed thanks to algorithms that can be specific to a particular agent and it will not be communicated to other agents. The intrinsic trusts must be recalculated after the update of trusts in the TrustGraph if one of the basic elements has changed or if a new element enters into its calculation.

Definition 7. (TRUSTSET). Let $TS_A = (TG_A, TT_A)$ be the *TrustSet* of agent A which is a pair of a public part, the *TrustGraph*, and a private part, the *TrustTable*.

Each agent stores its own TrustSet. When an agent X wants to cooperate with another agent Y , based on its TrustSet, X can estimate trust estimation toward Y directly or indirectly. Interested readers can refer to [2] for a detailed representation of the TrustSet and of the algorithms built to update the TrustGraph and the TrustTable and to compute information reliability.

The TrustSet is stored in a decentralized way in each agent. This choice induces several advantages. It avoids the failure of a single point in a centralized system. The system can run normally although there are failures in some agents of the system. It can also save network resources such as

power, bandwidth and computation power in a mobile wireless environment.

B. Spatial organization

Due to the mobility of the agents in the distributed information collecting system we study, the topology of the network composed by the agents has a significant impact on the communication system. At first, meetings between agents will induce a particular social organization between agents associated to TrustSets. Once well-grounded, this social organization will induce a particular kind of spatial organization that we call “cluster”: agents that fully trust each other will stay connected in order to collaborate more efficiently. After its formation, a cluster can be regarded as an ad hoc like network that automatically “emerges” from the social organization. With trust established as a criterion of cluster composition, dissonant agents will be automatically excluded from clusters.

Let $\delta(X, Y)$ denote the spatial distance between two agents X and Y .

Let $VN = \{(X, Y) \mid \forall X, Y \in V, X \neq Y, \delta(X, Y) \leq r\}$ be the set of all spatially neighboring agents, where r denotes the agents’ communication range.

Let $EN(V) = \{\overline{XY} \mid (X, Y) \in VN\}$ be the set of all links between spatially neighboring agents.

Definition 8. (NEIGHBOR AGENTS) *In V , two agents A et B are said neighbors iff $(A, B) \in VN, A, B \in V$.*

Definition 9. (CLUSTER) *A cluster on a set of agents V , denoted $Cl = (V_{Cl}, E_{Cl})$, is a graph which satisfies at one moment t the constraints below:*

$V_{Cl} \subseteq VN$
 $E_{Cl} \subseteq EN(V)$
 $\forall \overline{XY} \in E_{Cl}, T_{XY} > Upp$ and, $T_{YX} > Upp$ where Upp ($Upp \in [0, 1]$) is the trust level above which an agent is considered as reliable.

Theorem 10. $(V_{Cl}, E_{Cl}) \subseteq (V, E(V))$

Proof: As $V_{Cl} \subseteq VN, E_{Cl} \subseteq EN(V)$ and $VN \subseteq V, EN(V) \subseteq E(V)$ so $(V_{Cl}, E_{Cl}) \subseteq (V, E(V))$. ■

In the sequel, Cl_X with $X \in V$ will denote the cluster in which X is an element. $X \in V_{Cl_X}$ and $\exists Y \in V_{Cl_X} \mid \overline{XY} \in E_{Cl_X}$.

For any group of agents $T \subset V$, Cl_T will denote the cluster in which $V_{Cl_T} = T$.

Note that Cl_{XY} with $X, Y \in V$ will represent the cluster composed by the two agents X and Y . $V_{Cl_{XY}} = \{X, Y\}$, $E_{Cl_{XY}} = \{\overline{XY}\}$.

III. CLUSTER DYNAMICS

A. Creation of the cluster

In this section, we study how an agent can build a cluster with another agent. A cluster is initialized by two agents who satisfy two conditions: they trust each other and they are neighbors.

The clustering at the physical layer will happen once agents trust each other, so this process will take time because trust values changes over time from an initial value set to 0.5. The

algorithm below presents how two neighbors that trust each other build a cluster.

Algorithm 1 Creating cluster with two neighbours

Input : Two neighbor agents X et Y who do not belong to any cluster.

Output : A new cluster Cl_{XY} with two agents is created in case of success. Do nothing otherwise.

Begin

If $T_{XY} > Upp$ Then
 An initiator X sends a cluster formation request to the agent Y
 if $T_{YX} > Upp$ then
 Y sends its acceptance to X
 X creates a new cluster Cl_{XY} and informs Y of the creation of the cluster
 end

End

End

Note that as $Cl_{XY} = Cl_{YX}$ by definition of a cluster. The roles of agents are not significant: the same cluster is produced if it is created by X or by Y .

B. Merging clusters

After a certain time, necessary for agents to get a good evaluation of trust in other agents, agents begin to create clusters according to *Algorithm 1*. This algorithm can be used in case agents do not belong to any cluster. However in case at least one of both agents belongs to a cluster, they must apply another algorithm presented in *Algorithm 2*.

Algorithm 2 Merging clusters

Input : Two neighbor agents X, Y with agent X belonging to Cl_X

Output : The cluster Cl_X updated.

Begin

If $T_{XY} > Upp$ then
 X sends cluster formation request to its neighbor agent Y
 if $T_{YX} > Upp$ then
 Y sends its acceptance to X
 if $\nexists Cl_Y, Y \in Cl_Y$ then $Cl_X = Cl_X \oplus (Y, \{\overline{XY}\})$ end
 if $\exists Cl_Y, Y \in Cl_Y$ then $Cl_X = Cl_X \oplus Cl_Y$ end
 end

End

End

Definition 11. (MERGING OPERATOR \oplus) : Let $Cl_X = (V_{Cl_X}, E_{Cl_X})$ and $Cl_Y = (V_{Cl_Y}, E_{Cl_Y})$ two different clusters. We define $Cl_Z = Cl_X \oplus Cl_Y$ as $Cl_Z = (V_{Cl_Z}, E_{Cl_Z})$ with $V_{Cl_Z} = V_{Cl_X} \cup V_{Cl_Y}$ and $E_{Cl_Z} = E_{Cl_X} \cup E_{Cl_Y} \cup \{\overline{XY}\}$.

Theorem 12. *If agent X belongs to both Cl_X , Cl'_X then $Cl_X = Cl'_X$.*

Proof: If $\exists Cl_X \mid X \in Cl_X$, if $\exists Cl'_X \mid X \in Cl'_X$ then by applying the merging operation on both clusters we get: $Cl_X = Cl_X \oplus Cl'_X$, $Cl'_X = Cl_X \oplus Cl'_X$ so $Cl_X = Cl'_X$. ■

C. General Properties

1) *Maintaining connectivity:* One of the main properties of clusters is the capacity of maintaining connectivity between its elements to ensure a communication channel between reliable agents. The problem of connectivity maintenance is to ensure the existence of a reliable communication channel throughout the mission. The difficulty in the chosen example of mapping mobile robots is that any robot can potentially break down and cause the disjunction of any robot from the rest of the team. Hence the reliable communication channel we are promoting will be broken and we cannot presuppose the availability of another communication infrastructure which will be operational and compatible with our agents. In our context, we use an algorithm to maintain connectivity between cluster agents using the mechanisms of sensitivity proposed by Le et al. [8] in which he uses the sensitivity connectivity - an original concept in MANETs - to build a distributed representation and local connectivity.

To maintain the network connectivity for elements in the network, we use a “fixed” robot as the reference node. While moving to perform its task, each robot must remain in contact with at least one neighboring robot from which a channel of communication with the reference robot can be established. If the robots are all successful, then the connectivity of the whole system will be ensured.

2) *Group communication :* In a distributed information gathering system, robots need to communicate to cooperate effectively. Many studies have concluded that even the exchange of a small amount of information improves MAS performance for some tasks [9], [10]. To achieve a high degree of flexibility and autonomy, communication between robots should be based on wireless communication technologies. In addition, the used communication technology must allow robots to auto-organize to be operational without any centralized administration, and must be able to adapt to the mobility of robots during their mission. A network with such characteristics is known as a mobile ad hoc network : MANETs. These characteristics make MANETs very flexible and easy to deploy. For this reason, the use of MANETs for communication between robots belonging to the same cluster, in places where we cannot reasonably suppose the existence of a robust communication infrastructure, is extremely adequate.

A robot belonging to a cluster is not only an “ordinary” networked node but also a router that relays messages for its neighbors. Communication between robots which are not neighbors can thus take place through consecutive intermediate relaying nodes. Communication in clusters can be interpreted as follows in terms of ad hoc network: when neighbouring and reliable robots have constituted a cluster, they are considered as an ad hoc type network; thus they can communicate and rely on MANETs routing protocol for message transmission

so that in a finite period of time (a step), a message sent by a robot is received correctly by all the elements of its cluster.

D. Cooperation of agents in the cluster

Moreover we aim to create a role-based cooperation for cluster agents. We define a role as a clearly identified behavior description. Several roles are generally required to perform a given task. We propose to use roles to describe clusters needed to perform the various tasks necessary to carry off a given mission.

The basic idea of our approach is to consider each cluster as a group. The master of the cluster (*e.g.* the “reference agent”) plays the role of group manager. The second step is to build a description of the system and to provide it to the cluster agents so that they can argue it. Thus, agents independently choose the roles which they think most appropriate for themselves. The reference agent takes on helping cluster agents in case of conflict between agents (*e.g.* many agents want to execute the same task or the same role). To resolve conflicts, a protocol of roles allocation based on auction is usually used. At first, each agent sends to the reference agent its offer about the price it asks to finish the task. Then the reference agent selects the agent proposing the lowest task price. The choice of the reference node depends on the goal of the application and might involve multiple criteria such as energy level, number of neighbors, hardware requirements, etc.

The protocol of roles allocation we need for a cluster must allow dynamical changes in the overall organization. So the protocol we use is inspired by the algorithm DMAC (Distributed and Mobility-Adaptive Clustering) proposed by Basagni [11] to partition a mobile ad hoc network (MANET) in clusters.

IV. LINKS BETWEEN SPATIAL ORGANIZATION AND SOCIAL ORGANIZATION

After the previous static description of TrustSet in both social and spatial organizations, we address in this section the issue of links between TrustSets which are stored in each agent in a social organization and high level clusters in a spatial organization, i.e. how TrustSets of agents are exchanged and how they are merged together.

We aim to build a shared TrustSet which is stored on each individual and can be shared with others in a cluster (i.e. all agents in a cluster could have the same TrustSet). This way, trust information is exchanged locally through individual interactions in order to avoid many drawbacks, such as single point failure, requirement of infrastructure, problem of performance bottleneck, etc.

A shared TrustSet must be designed in a distributed way suited to a group dynamics that uses frequent topology and membership changes. At each communication with an agent Y , the agent X will eventually communicate to Y its data (information items collected directly by the agent) but also some of its metadata (all information about trust). In particular, it will share its TrustGraph, which contains all public metadata, but won't share its TrustTable because it is built by a personal

calculation and thus contains private information. After receiving a TrustGraph, an agent integrates it in its own one. Then it uses the obtained TrustGraph to update its TrustTable. The TrustGraph of agent X is updated when it receives the TrustGraph of agent Y following 4 stages¹ as follows:

- X calculates its trust T_{XY} in Y or updates the existing value by comparing its own data with received ones;
- X changes its current root node by the new one which is composed by both nodes X and Y ;
- X connects the TrustGraph of Y to its own TrustGraph in which its root node has been changed;
- X corrects all inconsistencies in the shared paths.

The TrustGraph sent by an agent must contain its building information (e.g. ids of agents whose TrustSets are merged, the time of mergings, etc.).

When agents meet new neighbours (or when a new agent enters the cluster), they will exchange and update their TrustGraphs using the Merging TrustGraphs algorithms.

When an agent meets old neighbours, after checking the TrustGraphs' building information, an agent decides or not to update its own TrustGraph (i.e. when the agent detects that one of the basic components of its TrustGraph has changed, it will update its TrustGraph).

From its shared TrustGraph, an agent A can compute its private TrustTable to estimate the trust value allocated by its cluster to another agent Y (Y does not belong to A 's cluster) using the formula : $T_Y^{Cl} = \frac{\sum_{X \in V_{Cl}} T_{XY}^A}{Card(V_{Cl})}$, where T_Y^{Cl} denotes the trust of the cluster Cl on the agent Y and T_{XY}^A the intrinsic trust of agent $X \mid X \in V_{Cl}$ on agent Y in the TrustTable of A .

V. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a mechanism to improve cooperation between reliable agents in a disturbed, spatialized data gathering system where information collection and transmission can be altered by unreliable agents. In our approach, each agent computes a trust value assigned to other agents in order to build (from local to global) a highest level structure designed as a cluster on which we can apply improvements on communication, data gathering and role-sharing derived from ad hoc networks algorithms. We show how social and spatial organisations can be linked in order to achieve interesting emerging associations of agents intended to reach faster the objective set to the system.

The advantage of our approach is that each cluster agent cooperates better with all the agents of its cluster by minimizing the data redundancy caused by overlapping agents and by limiting the information latency inside the cluster. Such a cooperation induces at the same time better coherence and better robustness of the system. Aiming to justify our proposition and investigate its performance, extensive simulation studies using GAMA platform [12] are being currently carried out.

The future work will focus on the community self-organization about communication management, on the structuring of sub communities according to their reliability and on

the limits of perturbation a disturbed system can support by using a complex system approach [13].

REFERENCES

- [1] Q. A. NguyenVu, B. Gaudou, R. Canal, S. Hassas, and F. Armetta, "Stratégie de communication dans un système de collecte d'information à base d'agents perturbés," in *Systèmes Multi-Agents*, France, 10 2009, pp. 207–217. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00449093/fr/>
- [2] Q. A. Nguyen Vu, B. Gaudou, R. Canal, S. Hassas, and F. Armetta, "TrustSets - Using trust to detect deceitful agents in a distributed information collecting system," in *2010 IEEE-RIVF International Conference on Computing and Communication Technologies*, 2010, p. To appear.
- [3] B. Horling and V. Lesser, "A survey of multi-agent organizational paradigms," *Knowl. Eng. Rev.*, vol. 19, no. 4, pp. 281–316, 2004.
- [4] L. E. Parker, C. M. Reardon, H. Choxi, and C. Bolden, "Using critical junctures and environmentally-dependent information for management of tightly-coupled cooperation in heterogeneous robot teams," in *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2872–2879.
- [5] K. Drira, H. Kheddouci, and N. Tabbane, "Virtual dynamic topology for routing in mobile ad hoc networks," in *Proceedings of the International Conference on Late Advances in Networks ICLAN'2006*, France, 2006, pp. 129–134, Lyon1. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00188474/en/>
- [6] C. E. Perkins, *Ad Hoc Networking*. Addison-Wesley Professional, 2008.
- [7] M. Haddad and H. Kheddouci, "A survey on graph based service discovery approaches for ad hoc networks," 2007, p. To appear, Lyon1. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00188379/en/>
- [8] V. T. Le, N. Bouraqadi, S. Stinckwich, V. Moraru, and A. Doniec, "Making networked robots connectivity-aware," in *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1835–1840.
- [9] B. Maclennan, "Synthetic ethology: An approach to the study of communication," in *In*. Addison-Wesley, 1991, pp. 631–658.
- [10] T. Balch, Ronald, and C. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, pp. 27–52, 1994.
- [11] S. Basagni, "Distributed clustering for ad hoc networks," in *ISPAN '99: Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks*. Washington, DC, USA: IEEE Computer Society, 1999, p. 310.
- [12] E. Amouroux, C. Quang, A. Boucher, and A. Drogoul, "GAMA: an environment for implementing and running spatially explicit multi-agent simulations," in *10th Pacific Rim International Workshop on Multi-Agents (PRIMA)*, Thailand, 2007.
- [13] S. Hassas, "Systèmes complexes à base de multi-agents situés," Université Claude Bernard-Lyon 1, Mémoire d'habilitation à diriger les recherches, 2003.

¹Interested readers can refer to [2] for detailed stages 1,3,4 in section "Merging TrustGraphs".