Supporting Agent CoT Groups Formation by Trust

Gianfranco Fortino*, Lidia Fotia§, Fabrizio Messina†, Domenico Rosaci‡, Giuseppe M. L. Sarné §
*Department DIMES, University of Calabria, Via P. Bucci, cubo 41c, 87036 Rende (CS) giancarlo.fortino@unical.it
§Department DICEAM, University of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal., Italy, {lidia.fotia, sarne}@unirc.it
†Department of Mathematics and Informatics, University of Catania, Viale Andrea Doria Catania, Italy, messina@dmi.unict.it
‡Department DIIES, University of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal., Italy, domenico.rosaci@unirc.it

***Abstract*—IoT devices dealing with complex tasks require powerful hardware capabilities or to get resources on the cloud. When an IoT device is "virtualized" on the Cloud, it can rely on one or more software agent that can exploit its social attitude to interact and cooperate. In this context, the choice of a partner to cooperate is a sensitive question but when an agent cannot perform a reliable choice then, like real communities, it can ask information to other agents it considers as trustworthy. This process can be improved by partitioning the agents in groups by using trust relationships to allow agents to interact with the most reliable partners. To this aim, we designed an algorithm to form agent groups based on reliability and reputation information and the results of some simulations confirmed its potential advantages.**

***Index Terms*—Cloud Computing; Cloud of Things; Internet of Things; Multiagent system; Reputation; Trust; Voting**

I. Introduction

Recently, the "Internet of Things" (IoT) and Cloud Computing (CC) converged into the so called Cloud-of-Things [1], [2] (CoT) for supporting computational and storing requirements [3] of ubiquitous and heterogeneous IoT devices, also in nomadic scenarios [4]. Moreover, to promote cooperation among IoT devices, they can be also associated with software agents [5]–[8] for taking benefit from their social attitudes.

In this context, the choice of a reliable partner needs of suitable information that can be also required as recommendations to trustworthy agents. To this aim, we propose of supporting this process by encouraging agents to form groups of reliable recommenders exploiting some type of social relationships existing among the group members [9], [10]. For instance, an important property within a community is a high level of mutual trustworthiness among its members [11], [12]. Therefore, we consider the trust-based processes to form agent groups of reliable recommenders over a CoT context as potentially capable to significantly improve the IoT devices activities.

To this aim, we consider a CoT environment where heterogeneous devices consume/produce services and/or extract/exchange knowledge assisted by personal software agents working over the CC. We take into account a specific scenario where each IoT device and its associated agent are considered a single entity; moreover, we also take into account the dynamicity of agents in the CoT environment, i.e. their ability to change groups based on their own convenience [13], [14].

The basic idea is that, the generic consumer agent, when using some data services (s) from a provider agent, should consider its past experiences. When no data about paste experience does exist, the agent will exploit the recommendation given by the community [15], [16]. In particular, the agents belonging to the same group of the agent who has requested the opinion/recommendation of the provider agent, will provide the information for free, otherwise a fee has to be paid for the recommendation/opinion. This approach leads to a competitive scenario on which groups/agents are interested in accepting/belonging to those agents/groups having a high reliability and helpfulness. Moreover, to evaluate the helpfulness of an agent we consider the effectiveness of its recommendations, while for a group it is the average of the helpfulness of its members.

In order to maximize the benefits of an agent to join with a group (and vice versa), we exploit trust measures to model a distributed group formation process. In particular, we designed a distributed algorithm matching devices and groups to improve individual and global satisfaction [9], [17] into the CoT on the basis of trust measures considering the agent helpfulness in providing useful recommendations. In this respect, as it happens in real user communities, in place of the global reputation, we adopt a *local reputation* [18] approach where the reputation value is based on the opinions coming from the friends (or friends of friends and so on) of an agent. This local approach gives important benefits in a CoT context, among which *i*) heavy computational tasks and communication overloads can be avoided when collecting opinions and evaluating the trustworthiness of their sources and *ii*) the system reactivity is increased.

Moreover, likely processes having place in human societies [19], groups are formed by using a voting mechanism, where each vote combines reliability and local reputation measures. Finally, to form groups with a high level of mutual trust among its members we designed a distributed algorithm for group formation (see Section III) that we verified, in terms of efficiency and effectiveness, by means of some experiments, on a simulated agent CoT scenario, which confirmed our expectations.

The rest of the paper is organized as follows. Section II describes the adopted local trust model and voting mechanism, while Section III presents an algorithm to form groups. The

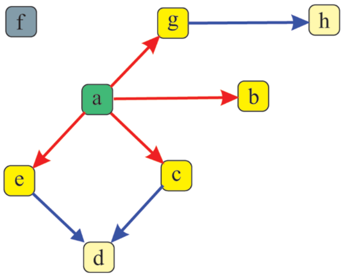

Fig. 1. The ego-networks of the agent a including all the nodes of the virtual community (nodes from a to f) for which a direct link (red colored) to a there exists and some other agents indirectly connected to a by a path of length 2 (e.g., all the agents connected to a by red and blue links).

experimental results are dealt in Section IV and in Section V the related literature is presented. Finally, in Section VI some conclusions are drawn.

II. The Local Trust Model

For convenience, we represent the agents trust relationships as a graph G, in which a direct edge linking two nodes (i.e., agents) is associated with the trust level (ranging $[0,1] \in \mathbb{R}$, where 0/1 means the minimum/maximum value) an agent has in another agent, and the ego-network E_i of an agent $a_i \in A$ as a sub-graph $E_i \subseteq G$ including those nodes (i.e., agents) connected to a_i in a fixed depth (see Figure 1).

For the generic nodes $i, j \in G$ (i.e., the associated agents a_i and a_j), the measure of the local trust $\tau_{i,j}$ that i has about j combines the reliability $\rho_{i,j}$ (i.e., a measure of the confidence that a_i has about the capability of a_j of providing good suggestions) and the *local reputation* $\sigma_{i,j}$ (i.e., a measures of how much, on average, the agents of E_i estimate the capability of a_j of having good interactions).

Usually, $\rho_{i,j} \neq \rho_{j,i}$ is an asymmetric measure computed as:

$$\rho_{i,j} = \frac{1}{q} \cdot \sum_{k=1}^{q} f_{i,j}^{\,k}$$

by means of all the feedback $f_{i,j} \in [0,1] \in \mathbb{R}$ assigned by a_i to a_j for each of the q interactions carried out with it. To this aim, let $rec_{r,j} \in [0,1]$ be the suggestion given by a_r about a_j and let $\epsilon_{i,r} \in [0,1]$ be the (average) helpfulness perceived by a_i about the capability of a_r to provide suggestions[1]. In detail, the helpfulness $\epsilon_{i,r}$ of a_r perceived by a_i is computed, with respect to the feedback released by a_i for each of the m accepted suggestions provided by a_r to a_i about other agents, as:

$$\epsilon_{i,r} = \frac{1}{m} \cdot \sum_{s=1}^{m} |f_s - rec_s|$$

To give relevance to the recommender agents in E_i which are closer to a_i, it is used a parameter ω computed as:

$$\omega_{i,r} = 2^{-(\,\widehat{l}_{(i,r)}-1)}$$

where $\widehat{l}_{(i,r)}$ is the shortest path between a_i and the recommender agent a_r. Now, by assuming that a_i, in its ego-network, is able to exploit a number p of recommenders to receive recommendations about a_j, then $\sigma_{i,j}$ can be calculated as:

$$\sigma_{i,j} = \frac{1}{p} \cdot \sum_{r=1}^{p} \left(\epsilon_{i,r} \cdot \omega_{i,r} \cdot rec_{r,j}\right).$$

The trust measure that an agent a_i has about an agent a_j can be computed by combining reliability and local reputation (which also takes into account the helpfulness) as:

$$\tau_{i,j} = \alpha_i \cdot \rho_{i,j} + (1 - \alpha_i) \cdot \beta_i \cdot \sigma_{i,j}$$

where α and β are two parameters ranging in $[0,1] \in \mathbb{R}$. The parameter α simply weights reliability and local reputation for giving more or less relevance to one or other. The parameter β is computed as $\beta_i = p/\|E_i(x)\|$ and takes into account the dependability of $\sigma_{i,j}$ on the number of p nodes belonging to E_i that contributed to compute $\sigma_{i,j}$ (indeed, if the number of these nodes is small then the local reputation measure loses of relevance because a_i will not have a sufficient information from its E_i about a_j). Note that for a newcomer agent, suitable "cold start" values of reliability, reputation and helpfulness are adopted.

The "trustworthiness" of a group g, as perceived by a_i (i.e., $\tau_{i,g}$), is determined by simply averaging all the trust measures computed by a_i for all the agents belonging to g. Similarly, the "trustworthiness" of an agent a_i, as perceived by a group g (i.e., $\tau_{g,i}$), is obtained by averaging all the trust measures about a_i computed by all the agents belonging to g.

Finally, when a decision about a new membership with a group g has to be taken, all the agents belonging to g give a preference (i.e., a vote) $v \in \{0,1\}$ to accept or not this agent into g (e.g., 0/1 means "not accept"/"accept") [20]. The vote depends from *i*) the local trust measure that the voter computed about the candidate, also exploiting the recommendations coming from its ego-network and *ii*) a suitable threshold $\Gamma_g \in [0,1]$ that worth 0 (i.e., 1) if $\tau < \Gamma_g$ (i.e., $\tau \geq \Gamma_g$). In the following, we represent the voting process referred to a group g for a potential new member y by adopting the voting criterion v proposed above, as the output of a function $V(g,v,y)$. For instance, a reasonable strategy may be of adopting a majority criterion for accepting a requester into a group.

III. The Distributed Agent Grouping Algorithm

This section presents the distributed agent grouping algorithm formed by two procedures respectively executed by each agent: *i*) belonging to the CoT for finding the "best" groups to join with, in terms of average value of $\tau_{i,g}$ (where g identifies a generic group); *ii*) acting as group *administrator* to evaluate if affiliating a new member with the group itself based on the mutual trust among the group members and the potential new member. The symbols used in the description of the algorithm are listed in Table I.

[1] If any recommendation was provided by a_r to a_i, then the helpfulness of a_r perceived by a_i will be $\epsilon_{i,r} = 0$.

TABLE I
TABLE OF THE MAIN SYMBOLS

Symbol	Description
A	set of agents associated to the IoT devices
G	graph representing the set of agents and their relationships $G = \langle N, L \rangle$
E_i	set of agents belonging to the ego-network of a_i, with $E \subseteq G$
Gr	set of all the groups
H_i	set of the groups which a_i is affiliated, with $H_i = \bigcup g_i \subseteq A$
K_g	set of agents affiliated with a group g
M	maximum number of new groups the single agent is able to analyze
W	maximum number of groups that an agent can join with
R	maximum number of agents belonging to a group
S_c	set of candidate groups
$V(\cdot)$	voting function
Y	set of groups randomly chosen, with $\|Y\| \leq M$
a	agent
a_g	agent administrator of the group g
g	generic group
$\tilde{t}$	time elapsed from the last execution of the procedure for an agent
$\hat{t}$	time elapsed from the last execution of the procedure for a group
θ	threshold on the level of trust between an agent and a generic group
ϕ	time threshold fixed by the agent administrator of a group
π	time threshold fixed by an agent
τ	trust

Algorithm 1 The procedure executed by a CoT agent.

Input: $H_i \subset Gr, W, \pi_i, \theta_i; Y = \{g \in G\}$ a set of groups randomly selected : $\|Y\| = M \leq W$, $H_i \bigcap Y = \{\ \}$, $Z = (H_i \bigcup Y)$

```
 1: for $g \in Z : \hat{t}_g > \pi_i$ do
 2:     Compute $\tau_{i,g}$ by exploiting the agents belonging to $g$.
 3: end for
 4: $m \leftarrow 0$
 5: Let be $S_c = \{g \in Z : \tau_{i,g} \geq \theta_i\}$, with $\|S_c\| = W$
 6: for all $g \in S_c : g \notin H_i$ do
 7:     send a join request to the agent administrator of $g$
 8:     if $g$ accepts the request then $m \leftarrow m + 1$
 9:     end if
10: end for
11: for all $g \in H_i : g \notin S_c$ do
12:     Sends a leave message to $g$
13:     $m \leftarrow m - 1$
14:     if (m==0) then break
15:     end if
16: end for
```

Algorithm 2 The procedure executed by a group administrator.

Input: $K_g, R, a_i, \phi, X = K_g \bigcup \{a_i\}$;

```
 1: for all $k \in K_g$ do
 2:     if $\tilde{t}_i \geq \phi$ then ask to $k$ for updating local trust values of $a_i$
 3:     end if
 4: end for
 5: if $\|X\| < R$ then
 6:     if $V(g, v, a_i) == 1$ then Send an accept message to $a_i$
 7:     else Send a reject message to $a_i$
 8:     end if
 9: else
10:     for all $k \in X$ do compute $\tau_{k,a_i}$
11:     end for
12:     Let $X' = \{k_1, k_2, \ldots, k_{\|K_g\|+1}\}$ with $k_i \in X \bigcup \{a_i\}$,
        ordered by trust with $\tau_{g,m} \geq \tau_{g,n}$ iff $m < n$
13:     if $X[\|K_g\| + 1] == a_i$ then Send a reject message to $a_i$
14:     else
15:         Send a leave message to the node $X[\|K_g\| + 1]$
16:         Send an accept message to $a_i$
17:     end if
18: end if
```

a) Algorithm 1: It is executed by the agent a_i to improve its configuration of groups in terms of overall mutual trust with the related peers. More in detail, let $H_i \subset Gr$ be the set of the groups which a_i belongs to and for which a_i stores the local trust measure $\tau_{i,g}$ of each group $g \in H_i \subset Gr$ contacted in the past and let $\hat{t}_g$ be the time elapsed from its last updating. Moreover, let W be a parameter specifying the maximum number of groups that an agent can join with, let M be the maximum number of groups the generic agent is capable to analyze, let π_i be a time threshold fixed by the agent a_i and, finally, let $\theta_i \in [0, 1]$ be a threshold on the trust value between the agent a_i and the generic group $g \in H_i$.

Firstly, the values of $\tau_{i,g}$ are updated if older than π_i (lines 1-3). Then, it is built a set of candidate groups S_c, with $\|S_c\| < W$, sorted in decreasing order based on the values $\tau_{i,g}$ of the groups, while Y is a set of groups randomly chosen and with the set $Z = Y \bigcup H$. The sets Y, Z and S_c might store the groups already belonging to H_i, while some others might be new groups that were selected at random and put into the set Y. Based on the groups in S_c not belonging to H_i, the agent a_i could improve the quality of its choices by joining with those groups. The two loops in lines 6-16 represents the kernel of the procedure, after that $H_i = S_c$.

b) Algorithm 2: It is performed by the administrator a_g of a CoT group g once an agent, denoted as a_i, sends a join request to a_g. Let $K_g \subset Gr$ be the set of the agents affiliated to g, where $\|K\| \leq R$ (with R the maximum number of agents allowed to be affiliated with g), let the set X be $X = K_g \bigcup a_i$, where a_i is the agent candidate to be affiliated with g and let ϕ a time threshold fixed by the administrator a_g. Moreover, the administrator a_g of a group g stores the values of the local trust computed by the members of its group for a_i which desire to join with, and the timestamp $\tilde{t}_i$ of its retrieval.

Firstly, the administrator a_g asks to the members of its group the updated local trust values about a_i (lines $1 - 5$), then if:

1) $||X|| < R$ (line 6), then all the agents in g give a vote. The function $V(\cdot)$, see Section II, combines all the votes to determine if the agent a_i is admitted or not in g.
2) $||X|| = R$ and the agent a_i is admitted into the group in place of another agent. To make comparable the agents, a natural measure is the trust of the group vs the agent itself, which is computed as explained in Section II (line 16). In particular, $\tau_{g,n}$ denotes the current value of trust between the group g and the agent $k_n \in X \bigcup \{a_i\}$.

The first scenario is dealt with in lines $6 - 11$, while the second one into lines $12 - 18$ of Algorithm 2.

IV. EXPERIMENTS

Some experiments have been carried out to test the capability of the proposed algorithm to form groups having a higher, in average, mutual trust among their members of that obtained from different compositions. The reader may refer to Table II for the list of experimental parameters.

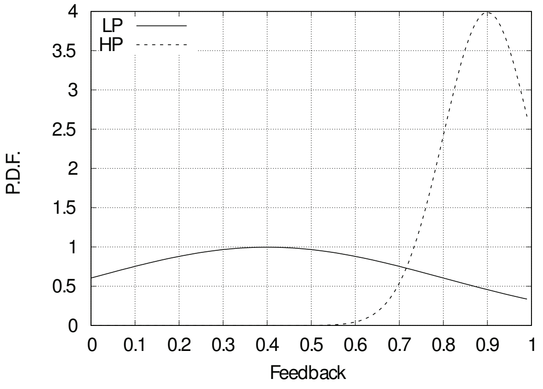

Fig. 2. a) Generated feedback values.

TABLE II
EXPERIMENTAL SETTINGS

Parameter	Value
General	
No. of Agents ($\|\|A\|\|$)	1000
No. of Feedbacks per step (Poisson distrib.)	$\lambda = 50$
Agents Performance (Reliability and trust)	
Low Performance (Normal Distribution)	$mean = 0.9; stdDev = 0.1$
High Performance (Normal Distribution)	$mean = 0.2; stdDev = 0.1$
Cold start value of trust	0.5
Ratio of reliable/unreliable agents	0.5
Group formation	
K (Max no. of agents per group)	20
M (Max no. of groups an agent analyzes)	$\{5, 10, 15, 20\}$
$\|\|Gr\|\|$ (No. of groups)	50
l_{Max} (Maximum recommender distance)	{1,2}
θ (Minimum value of trust for a group to be selected as candidate for group formation)	0.2

A network of 1000 different CoT agents (each one associated with a IoT device), 1000 initial trust relationships and $|Gr|$ groups, randomly formed, were generated. Trust values were set by adopting a normal distribution and with the ratio between trusted/distrusted agents set to 0.5. During the simulation the initial sparsity of the trust network will decrease for the availability of new reliability information.

At each simulation step some interactions among a subset of the agents was simulated and their "quality" evaluated by simulated feedback. For unreliable and reliable agents, the values of feedback were generated based on a normal distribution; these and the other simulation parameters are shown into Table II. More in detail, for each simulation step:

1) a number of interactions is simulated among agents;
2) 100 execution of the algorithm are simulated by triggering the algorithm 1 on 100 different agents randomly selected. For each agent request to join with a group, the administrator executes the algorithm 2 to decide whether or not to accept the requiring agent;
3) some statistics are computed.

To evaluate the simulation results, the measure *Average Mutual Trust* among the components of a group g as:

$$AMT_g = \frac{1}{2\|g\|} \sum_{\substack{i,j=1 \\ i \neq j}}^{\|g\|} (\tau_{i,j} + \tau_{j,i})$$

and the *Mean Average Mutual Trust*, for a certain configuration at a certain time-step, as:

$$MAMT(Gr) = \frac{1}{\|Gr\|} \sum_{i=1}^{\|Gr\|} AMT_{g_i}$$

were defined.

The first set of results is shown in Figure 3 and reports the median value of MAMT measured after each single step of the simulation for the different values of $M = [5 \div 10]$ for the first 30 steps of the simulation. For $M = 5$ is shown a slow convergence of the MAMT values, while for $M \geq 6$ there exist a radical change. Indeed, the parameter M represents the number of new groups analyzed by the single agent a_i in the execution of Algorithm 1, which are then mixed with groups already present in the set H_i, in the new set S_c. Therefore, the higher the parameter M, the higher the number of new groups analyzed in the algorithm 1, the higher the probability to join with a new group containing distrusted agents and replacing that showing the worst value of trust (by increasing the MAMT value because, sooner or later, distrusted agents will leave groups). Moreover, the presence into the all groups of distrusted agents at different simulation steps per different values of M is shown in Figure 4. Results confirm that almost distrusted agents are replaced by trusted agents into the groups.

Therefore, the execution of the distributed algorithm for group formation leads to a configuration of groups with a high level of (average) mutual trust among their members. In particular, in a simulated environment, the convergence of the algorithm towards a group configuration with trusted agents can be very fast, when the algorithm parameters are properly set (e.g. parameter M), leading to ruling out the unreliable agents from the groups very quickly.

V. RELATED WORK

In open, competitive and distributed contexts a large number of potential threats exist and, to this aim, trust systems can avoid to be engaged with unreliable partners [21]–[26].

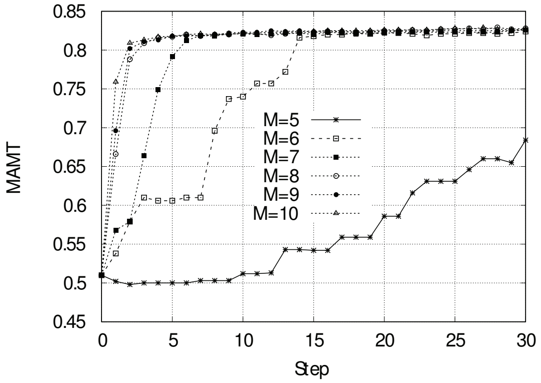

Fig. 3. MAMT - results until 30 Steps

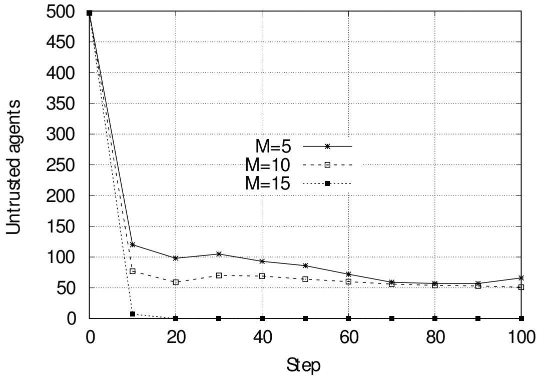

Fig. 4. Sum of untrusted agents vs number of steps of the simulation

With respect to the problem of suggesting to a group (i.e., a member of a community) if accepting (i.e., joining with) a candidate (i.e., a group), several trust-based approaches have been proposed. For instance, in [10], [27] is proofed that trust-based groups are more stable over time with respect to groups formed without to consider trust. Indeed, the expectations of receiving benefits is higher among the members of trust-formed groups. In such a context, the predominance of local trust is particularly true in large communities where each actor usually interacts only with a narrowest share of the community.

Examples of local trust approaches can be find in [28] and [29]. The first one is TidalTrust which exploits the closer neighbors to compute its trust predictions, also by ignoring part of the neighbors if the trust network is too sparse. The second techniques, named MoleTrust, performs a backward exploration by fixing a maximum depth in the search-tree of the trust network to calculate trust scores by using at depth x only the trust scores at depth $x - 1$.

Independently from the adopted group formation modalities, to reach a decision within a group voting mechanisms [30], [31] optimize the social utility [32] and avoid conflicts [33], although any "ideal" voting procedure exists due to the risks of manipulations [34]. This aspect is very critical for software agent communities, where agents can quickly examine the effects of each manipulation strategy [35]–[37]. In this respect, [38] presents a local trust-based voting, working in a mobile wireless scenario, where a node is admitted in a transmission path on the basis of the trustworthiness perceived by the other nodes. The actual trust of a node is propagated by mutual acquaintance among neighbors placed at one hop of distance on an oriented trust network by combining their confidence values considered as trust measures. A node will be trusted/distrusted by using a local voting scheme. In [39] faulting sensors are discovered by using a trustworthiness measure, named *SensorRank*, modeled by a Markov chain on the sensor network. This value is used in a voting scheme, named *TrustVoting*, where each vote implicitly represents the number neighbors referencing the opinions of a node and by weighting each vote proportionally to their proximity to the target node on the sensor network. In [40] a grid of agent-based sensors monitoring traffic flows on the roads is described. Each agent-based sensor of the grid is associated with a road and gathers, analyzes and aggregates acoustical signals generated by vehicles in their motion. Based on a distributed trust-system, each agent improves own performances by interacting with other sensors in its neighboring.

Finally, some trust systems have been conceived for IoT and CC contexts. For instance, in [41] two interacting IoT devices can mutually trust each other device and propagate their evaluations to the other nodes with a *word of mouth* approach. In [42] each node evaluates the trustworthiness of its friend nodes and the opinions of the common friends (by considering reliability and local reputation measures). A Trust Management system for a CC marketplace in [43] evaluates a multidimensional trustworthiness of the CC providers by exploiting different sources and trust information. CC federation are considered in [44], where a fully decentralized trust-based model for large-scale federations is designed to allow any node to find the most suitable collaborators in an efficient way, avoiding exploration of the whole node space by including trustworthiness information about the set of candidate nodes.

VI. Conclusions

In this paper, a CoT scenario supporting the virtualization of IoT devices over the cloud in a multi-agent context has been presented. The social attitude of software agents has been exploited to form groups for promoting satisfactory agents interactions. However, a satisfactory interaction depends on the choice of the partner but in absence of suitable information to perform an autonomous choice, some suggestions can be asked to those agents perceived as the mostly trustworthy.

To this aim, we designed a distributed algorithm to guide the formation of agent groups of reliable recommenders, in a competitive and cooperative scenario, exploiting a voting procedure focused on the agent capability of providing useful recommendation on the basis of reliability, local reputation and helpfulness measures. In particular, the adoption of *local reputation* measures avoids the heavy computational tasks and communication overheads required from a *global reputation* mechanism because only a little share of the agent community is involved in this process. Some experiments, in a simulated agent CoT scenario, confirmed the potential advantages deriving by our proposal in improving individual and group satisfaction in terms of mutual trust.

Acknowledgment

This study has been developed at **NeCS** Laboratory (DICEAM, University Mediterranea of Reggio Calabria).

References

[1] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of things: Integrating internet of things and cloud computing and the issues involved," in *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on.* IEEE, 2014, pp. 414–419.

[2] P. Parwekar, "From internet of things towards cloud of things," in *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*, Sept 2011, pp. 329–333.

[3] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[4] G. Aloi, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, and C. Savaglio, "Enabling iot interoperability through opportunistic smartphone-based mobile gateways," *Journal of Network and Computer Applications*, vol. 81, pp. 74–84, 2017.

[5] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in *Architecting the internet of things*. Springer, 2011, pp. 1–24.

[6] G. Fortino, R. Gravina, W. Russo, and C. Savaglio, "Modeling and simulating internet-of-things systems: A hybrid agent-oriented approach," *Computing in Science & Engineering*, vol. 19, no. 5, pp. 68–76, 2017.

[7] P. De Meo, F. Messina, M. N. Postorino, D. Rosaci, and G. M. L. Sarné, "A reputation framework to share resources into iot-based environments," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2017, pp. 513–518.

[8] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using trust and local reputation for group formation in the cloud of things," *Future Generation Computer Systems*, vol. 89, pp. 804–815, 2018.

[9] C.-M. Chiu, M.-H. Hsu, and E. T. Wang, "Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories," *Decision support systems*, vol. 42, no. 3, pp. 1872–1888, 2006.

[10] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarnè, "Trust and compactness in social network groups," *ACM Transactions on Cybernetics*, vol. 45, no. 2, pp. 205–2016, 2015.

[11] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "Combining trust and skills evaluation to form e-learning classes in online social networks," *Information Sciences*, vol. 405, pp. 107–122, 2017.

[12] C. Castelfranchi and R. Falcone, *Trust theory: A socio-cognitive and computational model*. John Wiley & Sons, 2010, vol. 18.

[13] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "An agent-oriented, trust-aware approach to improve the qos in dynamic grid federations," *Concurrency and Computation: Practice and Experience*, 2015.

[14] F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné, "A trust-based, multi-agent architecture supporting inter-cloud vm migration in iaas federations," in *International Conference on Internet and Distributed Computing Systems*. Springer, 2014, pp. 74–83.

[15] L. Palopoli, D. Rosaci, and G. M. L. Sarné, "A multi-tiered recommender system architecture for supporting e-commerce," in *Intelligent Distributed Computing VI*. Springer, 2013, pp. 71–81.

[16] ——, "A distributed and multi-tiered software architecture for assessing e-commerce recommendations," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 18, pp. 4507–4531, 2016.

[17] A. Blanchard and T. Horan, "Virtual communities and social capital," in *Knowledge and social capital*. Elsevier, 2000, pp. 159–178.

[18] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "Recommending users in social networks by integrating local and global reputation," in *Proc. of the 7th Int. Conf. on Internet and Distributed Information Systems*, ser. LNCS, vol. 8729. Springer, 2014, pp. 437–446.

[19] D. M. Kilgour and C. Eden, *Handbook of group decision and negotiation*. Springer Science & Business Media, 2010, vol. 4.

[20] L. S. Lai and E. Turban, "Groups formation and operations in the web 2.0 environment and social networks," *Group Decision and negotiation*, vol. 17, no. 5, pp. 387–402, 2008.

[21] T. Grandison and M. Sloman, "A survey of trust in internet applications," *IEEE Communications Surveys & Tutorials*, vol. 3, no. 4, pp. 2–16, 2000.

[22] G. Lax and G. M. L. Sarné, "CellTrust: a reputation model for C2C commerce," *Electronic Commerce Research*, vol. 8, no. 4, pp. 193–216, 2006.

[26] G. Fortino and P. Trunfio, *Internet of things based on smart objects: Technology, middleware and applications*. Springer, 2014.

[23] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[24] M. Momani and S. Challa, "Survey of trust models in different network domains," *arXiv preprint arXiv:1010.0168*, 2010.

[25] P. R. Vamsi and K. Kant, "Systematic design of trust management systems for wireless sensor networks: A review," in *Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on*. IEEE, 2014, pp. 208–215.

[27] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "Forming time-stable homogeneous groups into online social networks," *Information Sciences*, vol. 414, pp. 117–132, 2017.

[28] J. Golbeck, "Computing and applying trust in web-based social networks," in *PhD Thesis*. University of Maryland, Department of Computer Science, 2005.

[29] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. of the 2007 ACM Conference on Recommender systems*. ACM, 2007, pp. 17–24.

[30] S. J. Brams and P. C. Fishburn, "Voting procedures," *Handbook of social choice and welfare*, vol. 1, pp. 173–236, 2002.

[31] N. R. Council *et al.*, *Public participation in environmental assessment and decision making*. National Academies Press, 2008.

[32] L. Xia, "Computational voting theory: game-theoretic and combinatorial aspects," Ph.D. dissertation, Duke University, 2011.

[33] T. C. Beierle and J. Cayford, *Democracy in practice: Public participation in environmental decisions*. Resources for the Future, 2002.

[34] V. Conitzer and T. Sandholm, "Universal voting protocol tweaks to make manipulation hard," *arXiv preprint cs/0307018*, 2003.

[35] J. Pitt, L. Kamara, M. Sergot, and A. Artikis, "Formalization of a voting protocol for virtual organizations," in *Proc. of the 4th Int joint Conf. on Autonomous Agents and Multiagent Systems*. ACM, 2005, pp. 373–380.

[36] A. Gibbard, "Manipulation of voting schemes: a general result," *Econometrica: J. of the Econometric Society*, pp. 587–601, 1973.

[37] M. Satterthwaite, "Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions," *J. of Economic Theory*, vol. 10, no. 2, pp. 187–217, 1975.

[38] T. Jiang and J. S. Baras, "Trust evaluation in anarchy: A case study on autonomous networks." in *INFOCOM*, 2006.

[39] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, and W.-C. Lee, "Using sensorranks for in-network detection of faulty readings in wireless sensor networks," in *Proc. of the 6th ACM Int. Work. on Data Engineering for Wireless and Mobile Access*. ACM, 2007, pp. 1–8.

[40] M. N. Postorino and G. M. L. Sarné, "An agent-based sensor grid to monitor urban traffic," in *Proceedings of the 15th Workshop dagli Oggetti agli Agenti, WOA 2014*, ser. CEUR Workshop Proceedings, vol. 1260. CEUR-WS.org, 2014.

[41] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proceedings of the 2012 international workshop on Self-aware internet of things*. ACM, 2012, pp. 1–6.

[42] M. Nitti, R. Girau, L. Atzori, A. Iera, and G. Morabito, "A subjective model for trustworthiness evaluation in the social internet of things," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*. IEEE, 2012, pp. 18–23.

[43] S. M. Habib, S. Ries, and M. Muhlhauser, "Towards a trust management system for cloud computing," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*. IEEE, 2011, pp. 933–939.

[44] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures," *Future Generation Computer Systems*, vol. 56, pp. 77–94, 2016.