2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)

Social principles in agent-based trust management for the Internet of Things

Kalliopi Kravari, Nick Bassiliades
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, GR-54124, Greece
(kkravari, nbassili)@csd.auth.gr

Abstract— **The Internet of Things has the potential to change our daily life. It will create a world where everyone and everything will be connected and knowledge will be diffused in every direction. This open, distributed and heterogeneous environment raises important challenges, such as intelligence and trustworthiness. Intelligent Agents are considered as a technology that can deal with these issues, since they are capable of autonomously representing people, devices or services while a wide range of trust and reputation models have already been proposed. This paper reports on identifying and incorporating social parameters involved in the Internet of Things, although it is not considered as a social network, with the use of Intelligent Agents enabling trustworthiness in the environment. More specifically, two paradigms, social graphs and peer-to-peer networks will be discussed while a novel distributed mechanism for locating reliable reports, an important aspect of reputation trust models will be proposed. Finally, a use case scenario is presented that illustrates the viability of the proposed approach.**

Keywords- intelligent multi-agent systems; internet of things; trust management; social principles; report locating

I. INTRODUCTION

Over the last years, the Internet of Things (IoT) attracted much attention mainly due to its potential to change our daily life [22][14][38][28][36]. In the coming decades, the way people live, work and communicate is expected to change dramatically thanks to it. The Internet of Things attempts to create, in essence, a world where everyone and everything, now called Things, will be connected and knowledge will be diffused effortlessly in every direction. The open and distributed network combined with the enormous heterogeneity of things raises important challenges. Intelligence and trustworthiness, critical to its success, are some of them. The heterogeneity, for instance, makes it difficult to standardize the interaction and communication in the Internet of Things. The open and distributed environment allows the rapidly increasing Things to enter into the environment and reproduce themselves or create and delete other Things in the network. Malicious participants could pose a serious threat to the proper functioning of the network, harming its credibility through fake services, denial of cooperation or other malicious behaviors. Hence, Things acting in such an open and risky environment will have to make the appropriate decisions about the degree of trust that can be invested in a certain partner, a vital but still challenging task. [6][12][24][37][15][5][42].

In this context, Intelligent Agents (IAs) are considered as an appropriate and promising technology that will deal with these issues [34][23]. In fact, intelligent agents are not just a software application but a new, different way of interacting for people and objects. Intelligent agents are capable of autonomously representing people, devices or even services, ensuring optimal performance, flexibility and trustworthiness in interactions.[18] At the same time, a wide range of trust and reputation models have already been proposed by the research community, even though they refer mainly to Semantic Web, predecessor of the Internet of Things [7][9][27][42][33].

Although, neither the Internet of Things nor multi-agent systems (MASs) are considered primary social networks, examining the potential societal impacts and relationships among Things, objects and/or people, in the IoT is absolutely essential. In fact, research on the Internet of Things is expected to shift from intelligent objects to objects with a real social consciousness. Trustworthiness in such an environment, where objects and even people will try to preserve their unique characteristics, is complex and crucial. Hence, the social dimension of the Internet of Things is currently a new open research area.[35][21][25][3][26]

This paper reports on identifying and incorporating social parameters involved in the Internet of Things with the use of Intelligent Agents enabling trustworthiness in the environment. The aim is to allow different objects (and people) to establish and maintain social relationships based on their experiences, preferences and requirements without complex underlying network protocols. More specifically, two paradigms, social graphs and peer-to-peer networks will be discussed while a novel distributed mechanism for locating reliable reports, an important aspect of reputation trust models will be proposed. A major challenge for open distributed and sometimes large-scale systems, such as multi-agent systems and the Internet of Things, is how to locate ratings among the rest of the community. Hence, this paper focuses on this first step of every trust management system in order to incorporate promising social principles. Finally, a multi-agent use case scenario is presented that illustrates the viability of the proposed approach.

978-1-5386-2626-9/17/31.00 ©2017 IEEE
DOI 10.1109/SYNASC.2017.00062

II. TRUST, REPUTATION AND RISK

When discussing social principles in trust management, it is important to firstly define the notions of trust, reputation and risk as well as the involved parties and their potential interactions. Most researchers tend to consider trust and reputation as key elements in the design and implementation of modern (multi-agent) systems. However, there is still no single, accepted definition of trust within the research community whereas reputation and trust are still confused and used as synonyms.

Broadly speaking, trust has been defined in a number of ways in the literature, depending on the domain of use, and it is used as the basis for decision making in many contexts. Among the available definitions, there are two that can be used as reference points for understanding trust. The first is provided by Dasgupta [31] and according to him, trust is a belief an agent has that the other party will do what it says it will (being honest and reliable) or reciprocate (being reciprocative for the common good of both), given an opportunity to defect to get higher payoffs. The second definition is provided by Jøsang et al.[2] and it defines trust as "the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible". Both definitions state that dependence and reliability are two core concepts in trust. Fortunately, both of them are values that can be measured in part through reputation. In other words, trust is generally defined as the expectation of competence and willingness to perform a given task.

Trust, however, is much more than that; the uncertainties found in the modern MASs and the Internet of Things present a number of new challenges. More specifically, in open distributed systems, sometimes large-scaled such as the Internet of Things, agents represent different stakeholders that are likely to be self-interested and might not always complete tasks requested from them. Moreover, given that the system is open, usually no central authority can control all the agents, which means that agents can join and leave at any time. The problem is that this allows agents to change their identity and re-enter, avoiding punishment for any past wrong doing. One, more, risky feature of open systems is that when an agent first enters the system has no information about the other agents in that environment. Given this, the agent is likely to be faced with a large amount of possible partners with a different degree of efficiency and/or effectiveness.

Hence, since agents, such as individuals, may be dishonest, reputation ended up as a core element at trust establishment, in the sense that a better reputation can lead to greater trust. In general, reputation is the opinion of the public towards a party or an agent. Reputation allows agents to build trust, or the degree to which one agent has confidence in another agent, helping them to establish relationships that achieve mutual benefits. Hence, reputation (trust) models help agents to decide who to trust, encouraging trustworthy behavior and deterring dishonest participation by providing the mean through which reputation and ultimately trust can be quantified [32][4]. In other words, reputation is an estimated opinion of a party for another party. Hence, usually reputation is a personal and subjective quantity, referring not to what behavior a party has but rather what behavior others think that party has [17].

Risk, on the other hand, is often undertaken in the hope of some gain or benefit. Risk is actually a situation that involves exposure to danger or loss, since although the outcome of a transaction is important to a party, the probability of loss is non-zero. Hence, the amount of risk that a party may be willing to tolerate is directly proportional to the amount of trust that the party has in the other party [2]. As a result, the main aim of reputation models is to support the establishment of trust between unfamiliar parties, equilibrating the risk.

Finally, for purposes of better understanding consider at this point an agent A establishing an interaction with an agent X; agent A can evaluate the other agent's performance and thus affect its reputation. The evaluating agent (A) is called truster whereas the evaluated agent (X) is called trustee (Fig. 1). Of course, for some interactions an agent can be both truster and trustee, since it can evaluate its partner while it is evaluated by that partner at the same time. After each interaction in the environment, the truster has to evaluate the abilities of the trustee according the parameters of the used reputation models. Such parameters could be response time, validity or cooperation. In case of distributed models, such as those that can be used in the Internet of Things, the truster usually does not have to report its ratings but just to save them for future use.

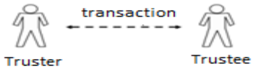

Figure 1. Tuster and trustee transaction.

III. SOCIAL GRAPHS

Social graphs are graphs that depict personal relations of users, usually in the context of internet. A social graph is considered as a model or representation of a social network, where the word graph has been taken from graph theory. Using the knowledge it represents in such a graph, relationships of interaction and / or proximity between the members of the environment can be determined. This, among other things, could help in locating reliable reports, an important issue for the fragmented credibility of models.

More specifically, a social graph is a diagram that illustrates interconnections among individuals or groups in a social network. Individuals and groups are nodes on the graph while interdependencies, called ties, can be multiple and diverse including a variety of characteristics or concepts. In an environment, such as the Internet of Things, the social graph for a particular party consists of the set of nodes and ties connected, directly or indirectly, to that party.

Practically, a social graph is demonstrated as a diagram with a set of points connected by lines. The points represent the parties (here agents) and the lines represent the ties. Fig. 2 presents a social graph, adopted by [40]. Usually, such a diagram due to the complexity of interconnections between parties is too massive and it needs a well-structured approach in order to be processed and provide useful conclusions.

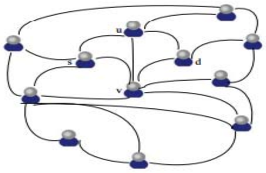

Figure 2. A social graph.

A social graph has to be transformed to something meaningful and editable in order to be used in domains such as trust management. To this end, Fig. 3 (adopted by [40]) presents a so called trusted graph, a graph that explicitly depicts opinions and relationships among parties. For instance, in the graph below, u and v have already known party d and, as a result, they have an opinion about its trustworthiness. How they formed their opinion is out of the scope of this study, however it usually depends on the estimation mechanism of the adopted trust (reputation) model and the influence or recommendation among parties.

The important here is that using such a graph we can find and study trusted paths among parties. A trusted path can be constructed through iterative recommendations. For instance, path (s, u, d) representing s's trust of d via u's recommendation. Multiple parallel and sequential paths are overlapped to form a trusted graph from s to d. Hence, using efficiently social trusted graphs could lead to promising partner locating, which is aim of this paper.

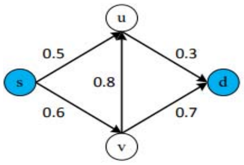

Figure 3. A trusted graph

IV. PEER-TO-PEER NETWORKS

In general, network structures affect the level of trust in social environments. For instance, a higher interconnectedness among parties could lead to a higher level of trust in the environment. Peer-to-peer (P2P) networks [39] could provide useful information and methods for this purpose. A peer-to-peer network is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session. Unlike the client/server model, in which the client makes a service request and the server fulfills the request, the P2P network model allows each node to function as both a client and server. In other worlds, it is an open, distributed communication environment, pretty much like the Internet of Things. Typically, peer-to-peer applications allow users to control many parameters of operation: how many member connections to seek or allow at one time; whose systems to connect to or avoid; what services to offer; and how many system resources to devote to the network. Fig. 4 presents a simple peer-to-peer network.

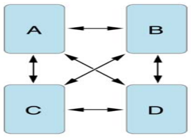

Figure 4. A peer-to-peer network.

Following, the peer-to-peer paradigm, there are two core ways to propagate messages in order to locate peers (or ratings in our case) [10]. The first approach assigns a maximum time-to-live (TTL) parameter to each request message hence the requesting peer sends the message to its neighbors, who relay it to their own neighbors and so on until the time-to-live value is reached. The second approach allows peers to relay the message only to one neighbor at time, since they have to wait the response from a neighbor before forwarding the message to another neighbor. The first approach increases the communication cost, leading to significant higher bandwidth consumption but partners (and so ratings) are located fast. On the other hand, the second approach requires low bandwidth but it leads to time delays since more time is required to get feedback for the requests. Over the last years, a number of researchers have proposed approaches that try to reduce bandwidth or improve response time (e.g. **Error! Reference source not found.**), mainly focusing on how to reach good and far away peers.

V. LOCATING RATINGS MECHANISM

The aim of this study is to incorporate potential social dimensions and relationships within the Internet of Things in order to relate them to the credibility of the interactions. As already discussed the main aim of this study is to provide a social agent-based mechanism, here called LOCATOR, for effective rating locating that could be easily used in any distributed reputation model in order to enable agents to establish and maintain relationships, limiting the disadvantages of the common distributed approaches. For this purpose, features from both social graphs and peer-to-peer networks are adopted.

First of all, we have to define additionally to Truster and Trustee agents (see section II), the notion of Recommender. A recommender is an intermediate agent (between truster and trustee) who helps the truster to evaluate the trustworthiness of the trustee by providing recommendation reports. Hence, in LOCATOR we identify three distinct and interchangeable entities; namely Truster (TR), Trustee (TE) and Recommender (RR).

A. Defining characteristics

Moreover, in an attempt to simulate real life, we assign an extendable (optional) list of characteristics LC to each entity (LC_x^n | n ∈ [1, N], x ≡ entity). Some of these characteristics could be club memberships (assuming that members of the club trust more each other than they trust non-members), occupation or type of service they provide, registration date or time period they are active in the environment and so on. For computational purposes, each agent assigns a value of importance (weight) to characteristics at the range $\mathrm{W}_n \in [0, 1]$ defining how much attention will be paid to each characteristic. Of course, an agent (e.g. representing a device) could be uninterested in characteristics, thus no weights will be assigned.

This way we are able to simulate and take into account two important aspects; namely influence and risk. For instance, consider club membership; a TR agent wants to interact with a TE agent and asks the opinion of an RR agent that belongs to the same club. If TR accepts the opinion of RR, because it trusts RR (they belong to the same club) then this could be called social influence (RR affects the opinion of TR). Risk on the other hand, depends on TR's experiences, preferences and requirements as well as its willingness to rely at same degree on others' recommendations. The more a TR agent trusts another RR agent, the more it takes into account its opinion/recommendation. We will come back to the issue of risk later in this section.

B. *Reward mechanism*

Recommendation, actually, allows opinions to propagate. The question is how parties will be convinced to recommend and propagate messages. Usually, entities are unwilling to sacrifice time and resources. Moreover, agents may change their objectives at any time, in dynamic environments such as IoT and MASs, thus, evolution over time is important and should be taken into account. For instance, a typical self-interested agent could provide recommendations over a period to gain credits and then, profiting from that could stop. It is the same case with a typical dishonest agent who could provide quality services over a period to gain a high reputation score, and then, profiting from that high score could provide low quality services.

In order to overcome this issue, we propose the use of a reward mechanism. Each party will get a credit whenever it provides a recommendation. The credit could be positive or negative (${}^{x}_{y}\mathrm{CR}_t$ |t ≡ time, x ≡ TR agent, y ≡ RR agent) and since there is no central authority to monitor and store credits, each agent should store by itself its credit score. Moreover since time is important, each credit will be valid only for a specific time period, depending on the personal strategy of the TR agent that requests the recommendations. Although it is out of the scope of this paper, we propose these credits to be digitally signed references in order to avoid frauds.

TABLE I. CREDIT VALYES

	contribution	*value*	*contribution*	*value*
CREDIT (CR)	Unsatisfying	-1	Significant	+0.5
	Insignificant	-0.5	Satisfying	+1
	Neutral	0		

For computational purposes we assign quantitative values to the quality contribution of a credit (Table I), in other words, the ability of an RR agent to recommend a good target. Hence the credit score is calculated as a normalized sum ($\mathrm{C}_y \in [-1,1]$ | -1≡terrible, 1≡perfect). A high credit score provides an evidence about the activity and recommendation quality of an RR agent. The higher the credit score is the more weighted is the recommendation (trustworthy partner).

C. *Propagating messages*

Yet, the major challenge for open distributed and sometimes large-scale environments remains how and to whom a request message should be sent. The question is to whom this message should be sent directly and how it will be propagated by the direct and indirect receivers. Inspired by social graphs and P2P networks, LOCATOR proposes an intuitive approach where agents take advantage of their previously experience and established relationships in order to propagate their new requests, finding, quite fast, ratings with small bandwidth cost. More specifically, although the notion of neighbors does not exist in IoT and MASs, agents can use previously known partners in a similar point of view.

To this end, the environment is considered as a social network of agents. Such a social network can actually be represented by a social graph; a graph based on agent interactions. Hence, agents that have already interacted can be considered, in our point of view, neighbors. Using the knowledge represented by the social graph, LOCATOR is able to determine the relationships of interactions among agents and the proximity between parties in the environment. Hence, an agent A that wants to collect ratings referred to agent X, does not send a request message to all agents but only to some previously known partners. Hence, it is necessary for the agent to store some information, e.g. name, characteristics, every time it interacts with another agent. To this end, a list of known agents, called here KA base, is needed.

LOCATOR identifies three categories of neighbors according to their social distance from the truster, called local neighbors, longer ties and longest ties, respectively. At this point, we have to define the notions of trusted path and trusted graph. A trusted path is a path that consists of a truster (TR – the source), several recommenders (RR agents), a trustee (TE – the target), and trust relations among them. In other words, it is a trusted path from the truster to the trustee. A trusted graph, on the other hand, is all the trusted paths starting from a truster and ending with a trustee. Hence, local neighbors are agents that have previously interacted with truster, longer ties are agents that can be connected to the truster with a path length less than five (5) nodes and longest ties are agents that can be connected with greater path length (>5). For instance, consider agent s, from the graph in Fig. 2, as a truster agent. Agents u and v can be considered as local neighbors whereas agent d is a longer tie.

Hence given a trusted path, propagation works in this way: if agent A_1 trusts agent A_2, and A_2 trusts agent A_3, then A_1 can derive some trust towards A_3. The challenge here is to set a proper limitation of path length, since a smaller limitation may lead to fewer paths, while a larger one may cause inaccurate prediction. Usually, in P2P networks there is a maximum time-to-live (TTL) parameter assigned to each request message, which means that a message will be propagated for a specific time period.

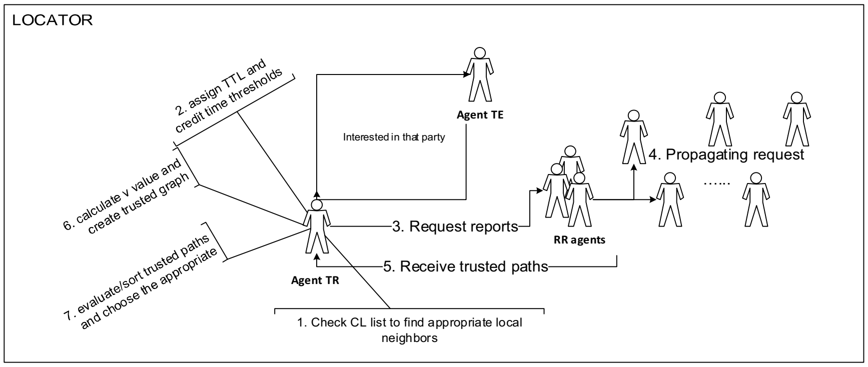
LOCATOR
2. assign TTL and credit time thresholds
Agent TE
Interested in that party
4. Propagating request
……
6. calculate v value and create trusted graph
3. Request reports
RR agents
7. evaluate/sort trusted paths and choose the appropriate
5. Receive trusted paths
Agent TR
1. Check CL list to find appropriate local neighbors

Figure 5. Brief functionality overview of LOCATOR

Adopting the notion of TTL, in LOCATOR, each request message is accompanied with a TTL value, yet it represents neither the time that the message is valid nor the maximum path length (hops in the graph) but rather the time period that the truster will wait for response. In other words, truster does not determine how far the message will be propagated in the network but specifies how fast it needs feedback. This way, truster is able to locate reports quite fast and make quick decisions. Of course, if more accuracy is needed, a longer time parameter should be assigned.

D. LOCATOR mechanism

Taking all the above into account, our proposed approach LOCATOR, works as follow. Firstly, an agent TR interested in a trustee agent TE decides upon the characteristics it considers important, e.g. a club membership. As soon as, it determines a set of desirable characteristics {LC^{n} |n ∈ [1, N]}, it assigns proper weights w_{n} to each of them and searches its KA base, the list with previously known agents (local neighbors in other words) in order to find those that fulfil its requirements.

Characteristics that weight more are more important in the sense that TR believes that partners with these characteristics will be more reliable. As a result, their recommendation is expected to be more valuable (influence – subsection A). In this context, TR depending on its personal strategy could choose to ask (send a rating request) firstly to local neighbors with one or two high-weighted characteristics. For instance, partners that provide the same service or had a previous successful transaction with TR. If the feedback is not satisfying, TR may sent a request message to partners with lower-weighted characteristics.

After choosing the local neighbors that will be the direct receivers of the request, TR assigns two time thresholds to its request message, a TTL value and a requested credit time period, and sends it to them. They acting as recommenders (RR agents), on their turn, propagate the message to their own local neighbors following the same procedure as long as they have time (t < TTL). Finally, these RR agents send the feedback (recommendation and credit score) to TR. Feedback is, actually, trusted paths from TR to TE through RR agents.

At the next step, TR assigns a value V, an indication of relevance, to each received trusted path. This value is calculated as follows: V = pl – 0.5*hp

$$V = \begin{cases} (pl - 0.25 * hp) * \mathrm{C}_{RR}, \forall pl \leq 5 \\ (pl - 0.5 * hp) * \mathrm{C}_{RR}, \forall pl \geq 6 \end{cases} \qquad (1)$$

Where pl stands for the length of the trusted path, hp stands for the number of network nodes while CRR is the credit score of the local neighbor (RR agent) that returned that path. CRR is based on RR agent's credits with a time stamp that fits in TR requested time period. Using this time period, TR has a clue about RR's latest behavior. In this study, CRR is calculated and provided by RR itself, yet TR agent could request also the related credit references in order to justify the score or even calculate it by itself. Credit verification would also support honesty regarding this issue.

The V value attempts to discard feedback, taking into account the concept of risk. More specifically, longest ties (subsection C) are more possible to be completely strangers even for TE's local neighbors. Hence, they can be considered as less trusted, which means that TR will take more risk. On the other hand, longer ties are more possible to be previously known partners of the TE's local neighbors and probably they are more valuable recommendation sources.

At this point TR received feedback and created a trusted graph by combining all available trust paths. For multiple trusted paths in a trusted graph such as this, the main challenge is how to combine the available evidence. In LOCATOR, the V value discussed above will do the job. Each path has an indicator, a value estimating the risk and social proximity. Although, it is a matter of personal strategy, we propose TR to sort trusted paths in descending order and choose those with higher V value. In the rare case that two or more paths have the same V value, the one that includes the local neighbor with the higher weighted characteristics (or features more characteristics) could be chosen. Moreover, the more high-scored trusted paths supports a potential partner the more positive it is for its trustworthiness. Fig. 5 displays a brief functionality overview of the LOCATOR approach.

However, how TR will avoid partners that it is unwilling to interact with or had a previous bad experience, related information will be stored in KA base, is still an issue. To deal with this, LOCATOR, if TR wants to avoid some agents, proposes two possible solutions. The first one is to include one more parameter to the initial request message, a list with the names of unwanted partners. Hence, when the message has to be propagated, RR agents will exclude possible local neighbors that belong to that list. The second solution is to receive the trusted paths and then check the intermediate agents (nodes). If there is any of the black-listed agents there, this specific trust path will be excluded. The second solution increases storage, time and bandwidth cost but, on the other hand, TR does not reveal personal data.

This approach is logic-independent but when adopted at a distributed trust model it can be combined with any logic, like defeasible logic that models the way intelligent agents, like humans, draw reasonable conclusions from incomplete and possibly conflicting (thus inconclusive) information.

VI. USE CASE

In order to use and evaluate the proposed mechanism, we adopt the use of EMERALD [19], a framework for interoperating knowledge-based intelligent agents (Fig. 6). This framework is built on top of JADE [11], a reliable and widely used multi-agent framework. EMERALD was chosen since it provides a safe, generic, and reusable framework for modeling and monitoring agent communication and agreements. Moreover, it proposes, among others, a reusable prototype for knowledge-customizable agents (called KC-Agents) and the use of Reasoners [20]. The agent prototype promotes customizable agents, providing the necessary infrastructure for equipping them with a rule engine and a knowledge base (KB) that contains agent's knowledge and personal strategy

Additionally, EMERALD provides an advanced yellow pages service, called AYPS, that is responsible for recording and representing information related to registered in the environment agents, namely their name, type, registration time and activity. Hence, even if the proposed mechanism is distributed, agents that use it are able to send requests to AYPS in order to get first a list of potential partners, which is the case for newcomers. Next, they will use the locating mechanism in order to find the most appropriate partner. Of course, it is not necessary to use such services; it is up to each agent's personal strategy how it will locate potential partners.

In order to evaluate the viability of the proposed LOCATOR approach we designed a testbed consisting of agents providing services and agents that use these services, namely providers and consumers. All provides offer the same service and all consumers buy this service. We do not take into account performance parameters since we are interested just in locating partners in a (social) network of interacting agents. Each experiment is populated with a different number of provider and consumer agents; the population is divided. From experiment to experiment we increased the number of agents approximately about 10% in order to evaluate how LOCATOR behaves in various populated networks.

We run eleven experiments; the first was populated with 20 providers and 20 consumers whereas the last was populated with 100 agents, divided in providers and consumers. Mention that most agents (80% of the population) are equipped with a delay parameter. More specifically, these agents will transact with at least three other agents before starting exchange messages for report locating, with or without LOCATOR. Hence, the environment has enough time to gradually become a social network. Otherwise, at the beginning of the simulations none agent had interaction history and thus known partners (local neighbors) that could ask.

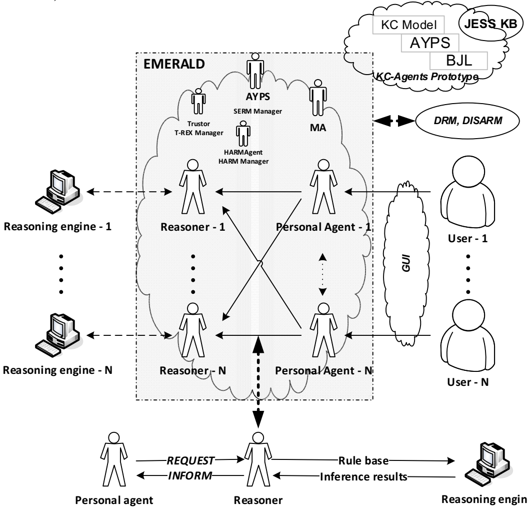

Figure 6. EMERALD overview.

Below we display the results of two sets of simulation. The first (Fig. 7) compares the mean number of required message exchanges using LOCATOR and without using it (pseudorandom message exchange – each agent sends requests to all available parties). It is clear that using LOCATOR results to significant lower message exchange.

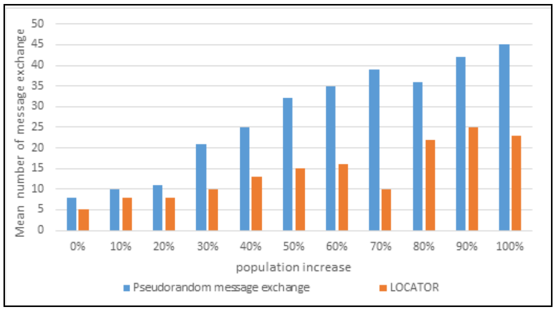

Figure 7. Mean number of message exchange with and without LOCATOR.

The second (Fig. 8) displays the mean number of returned trusted paths compared to the mean number of the returned path length. The aim of this set is to reveal the correlation between path length and number of paths. A few returned paths with high length would be an issue, but Fig. 8 displays a proportional relationship that supports LOCATOR dynamics. There is a sufficient number of returned trusted paths while the path length remains low, approximately close to the category of longer ties. Hence, it is more possible for TR agents to reach fast possible well behaved partners.

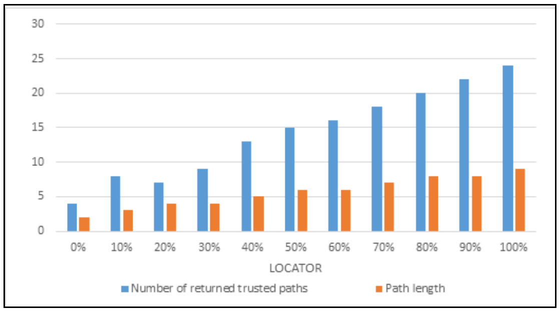

Figure 8. LOCATOR: number of returned trusted paths – path length.

VII. RELATED WORK

Trust and reputation represent a significant aspect in modern multi-agent systems. An interesting and very challenging active research area is already focused on them; various models and metrics have already been proposed in order to deal with the challenging decision making processes in the agent community, yet usually there is no much attention paid in locating ratings. Usually, this aspect is a small part of a more general estimation model. [13]

Social Regret [16], for instance, is a reputation system oriented to e-commerce environments that incorporates the notion of social graph. More specifically, Social Regret groups agents with frequent interactions among them and considers each one of these groups as a single source of reputation values. In this context, only the most representative agent within each group is asked for information. To this end, a heuristic is used in order to find groups and to select the best agent to ask. Social Regret, similarly to LOCATOR mechanism, is one of these cases that the social dimension of agents is taken into account. Yet, Social Regret does not reflect the actual social relations among agents, like our proposed approach, but rather attempts to heuristically reduce the number of queries to be done in order to locate ratings. Taking into account the opinion of only one agent of each group is a severe disadvantage since the most agents are marginalized, distorting reality.

Hang and Singh [8] also employ a graph-based approach for measuring trust, with the aim to recommend a node in a social network using the trust network. The model uses the similarity between graphs to make recommendations. The authors show that by calculating the similarity between the trust network and a structure graph (a path graph of length three), the similarity score can be viewed as a indicator that the agent is strongly connected by the strong neighbors of the requester. This approach similar to LOCATOR attempts to take advantage of graphs in order to locate better partners, although our approach takes into account more social aspects in an attempt to simulate the way people usually behave.

Finally, in [41] the authors propose an approach for computing trust in social networks using a set of trust chains and a trust graph. The model uses a trust certificate graph and calculates trust along a trust chain. Similar to LOCATOR this approach identifies the value of graphs, although its mechanism is quite limited to a chain modeling omitting, opposed to LOCATOR, other social aspects such as networking.

VIII. CONCLUSIONS

This paper presented a social locating mechanism, called LOCATOR, that can be adopted in any distributed reputation model, limiting the common disadvantages of the distributed approaches. The proposed mechanism adopted social principles by social graphs and peer-to-peer networks, bringing their advantages to the area of trust management in an environment such as the Internet of Things. LOCATOR though appropriate message routes, combines personal experience and recommendation reports. Hence, each agent is able to propagate its requests to the rest of the agent community, locating quite fast ratings from previously known and well-rated agents. Finally, we provided an evaluation that illustrates the usability of the proposed approach.

As for future directions, first of all, we plan to study further LOCATOR's performance by testing it in more complex use cases and even real-world applications, combining it also with Semantic Web metadata for trust [29][30]. Another direction is towards improving LOCATOR. There are still some open issues and challenges regarding locating ratings related to social and non-social issues. More technologies could be adopted for these purpose; ontologies, machine learning techniques and user identity recognition and management being some of them. Furthermore, we plan to study other related approaches such as socio-technical networks, that is networks of people and things interrelated in a meaningful manner via typed relations, as an overlay for enhancing hypermedia-driven interaction in IoT environments [1].

ACKNOWLEDGMENT

The Postdoctoral Research was implemented through an IKY scholarship funded by the "Strengthening Post-Academic Researchers / Researchers" Act from the resources of the OP "Human Resources Development, Education and Lifelong Learning" priority axis 6,8,9 and co-funded by The European Social Fund - the ESF and the Greek government.

REFERENCES

[1] A. Ciortea, A. Zimmermann, O. Boissier, and A. M. Florea. "Hypermedia-driven Socio-technical Networks for Goal-driven Discovery in the Web of Things," In Proceedings of the Seventh International Workshop on the Web of Things (WoT '16). ACM, New York, NY, USA, pp. 25-30, 2016.

[2] A. Jøsang, R. Ismail, C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support Systems, vol. 43(2), pp. 618–644, 2007.

[3] A. M Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between internet of things and social networks: Review and research challenges," IEEE Internet of Things Journal, vol. 1(3), pp. 206-215, 2014.

[4] A. Medić, "Survey of computer trust and reputation models – the literature overview," International journal of information and communication technology research, vol. 2(3), pp. 254-275, 2012.

[5] A. Whitmore, A. Agarwal, L. Da Xu, "The Internet of Things—A survey of topics and trends," Information Systems Frontiers, vol. 17(2), pp. 261-274, 2015.

[6] C. C. Aggarwal, N. Ashish, and A. Sheth, "The internet of things: A survey from the data-centric perspective," In Managing and mining sensor data. Springer US, 2013.

[7] C. H. Kuo, and S. E. Chang, "Web services-based trust framework design and applications: A case study. in ubiquitous and future networks (ICUFN)," 2016 Eighth International Conference on, pp. 851-856, 2016.

[8] C. Hang and M.P. Singh, "Trust-based recommendation based on graph similarity," in in AAMAS Workshop on Trust in Agent Societies (Trust), pp. 71-81, 2010.

[9] E. Majd, and V. Balakrishnan, "A reputation-oriented trust model for multi-agent environments," Industrial management & data systems, vol. 16(7), pp. 1380-1396, 2016.

[10] E. Meshkova, J. Riihijärvi, M. Petrova, P. Mähönen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," Computer Networks, vol. 52(11), pp. 2097–2128, 2008.

[11] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, "JADE: A white Paper," EXP in search of innovation, vol. 3(3), pp. 6-19, 2003.

[12] I. Lee, and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58(4), pp. 431-440, 2015.

[13] I. Pinyol, and J. Sabater-Mir, "Computational trust and reputation models for open multi-agent systems: a review," Journal of artificial intelligence review. Springer Netherlands, vol. 40(1), pp. 1-25, 2013.

[14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami., "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29(7), pp. 1645-1660, 2013.

[15] J. Niu, Y. Jin, A. J. Lee, R. Sandhu, W.Xu, and X. Zhang, "Panel security and privacy in the age of Internet of Things: Opportunities and challenges," In Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies, pp. 49-50, 2016.

[16] J. Sabater, and C. Sierra, "Social ReGreT, a reputation model based on social relations," SIGecom Exch, vol. 3(1), pp. 44-56, 2002.

[17] J. Sabater, C. Sierra, "REGRET: Reputation in gregarious societies," In proceedings of the fifth international conference on autonomous agents, AGENTS'01, ACM, New York, NY, USA, pp. 194–195, 2001.

[18] K. Dautenhahn, A. H. Bond, L. Canamero, and B. Edmonds, "Socially Intelligent Agents: Creating Relationships with Computers and Robots," 2013.

[19] K. Kravari, E. Kontopoulos, and N. Bassiliades, "EMERALD: A multi-agent system for knowledge-based reasoning interoperability in the semantic web," 6th Hellenic conf. on artificial intelligence (SETN 2010). LNCS, vol. 6040/2010, pp. 173-182, 2010.

[20] K. Kravari, E. Kontopoulos, and N. Bassiliades, "Trusted reasoning services for semantic web agents," Informatica: International journal of computing and informatics, vol 34(4), pp. 429-440, 2010.

[21] L. Atzori, A. Iera, and G. Morabito, "From" smart objects" to" social objects": The next evolutionary step of the internet of things." IEEE Communications Magazine, vil. 52(1), pp. 97-105, 2014.

[22] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," IEEE Transactions on Industrial Informatics, vol. 10(4), pp. 2233-2243, 2014.

[23] M. Gelfond, and Y. Kahl, "Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach," Cambridge University Press, 2014.

[24] M. Ma, P. Wang, and C. H. Chu, "Data management for internet of things: challenges, approaches and opportunities," 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp. 1144-1151, 2013.

[25] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social internet of things," IEEE Transactions on knowledge and data engineering, vol. 26(5), pp. 1253-1266, 2014.

[26] N. B. Truong, T. W. Um, and G. M. Lee, "A reputation and knowledge based trust service platform for trustworthy social internet of things," Innovations in clouds, internet and networks (ICIN), Paris, France, 2016.

[27] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "A survey on trust and reputation models for Web services: Single, composite, and communities," Decision Support Systems, vol. 74, pp. 121-134, 2015.

[28] O. Vermesan, and P. Friess (Eds.), Internet of things-from research and innovation to market deployment, Aalborg: River Publishers, 2014.

[29] P. Ceravolo, E. Damiani, and M. Viviani, "Adding a trust layer to semantic web metadata," In soft computing for information retrieval on the web, vol 197: pp. 87-104, 2006.

[30] P. Ceravolo, E. Damiani, and M. Viviani, "Bottom-up extraction and trust-based refinement of ontology metadata," IEEE Transactions on knowledge and data engineering, vol. 19(2), pp. 149-163, 2007.

[31] P. Dasgupta, "Trust as a commodity," Gambetta D. (Ed.). Trust: Making and Breaking Cooperative Relations, Blackwell, pp. 49-72, 2000.

[32] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems." Communications of the ACM, vol. 43(12), pp. 45-48, 2000.

[33] R. Burete, A. Badica, C. Badica, and F. Moraru, "Enhanced Reputation Model with Forgiveness for E-Business Agents," International Journal of Agent Technologies and Systems (IJATS), vol 3(1), pp. 11-26, 2011. DOI: http://dx.doi.org/10.4018/jats.2011010102

[34] R. H Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni, "Multi-Agent Programming," Springer, 2014.

[35] R. L. Hobbs, and W. Dron, "Using intelligent agents for social sensing across disadvantaged networks," In Mobile ad hoc and sensor systems (MASS), 2015 IEEE 12th International conference on mobile ad hoc and sensor systems, pp. 633-638, 2015.

[36] R. Want, B. N. Schilit, and S. Jenson, "Enabling the Internet of Things," IEEE Computer, vol 48(1), pp. 28-35, 2015.

[37] S. C. Mukhopadhyay, and N. K .Suryadevara. "Internet of Things: challenges and opportunities," In Internet of Things, pp. 1-17, 2014.

[38] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," Information Systems Frontiers, vol. 17(2), pp. 243-259, 2015.

[39] S.Androutsellis-Theotokis, and D. Spinellis, "A survey of peer-to-peer content distribution technologies," ACM Computing Surveys, vol. 36(4), pp. 335–371, 2004.

[40] W. Jiang, G. Wang, M.Z.A. Bhuiyan, and J. Wu, "Understanding graph-based trust evaluation in online social networks: methodologies and challenges," ACM Computing Surveys, vol 49(1), Art. 10, 2016.

[41] Y. Zuo, W.-C. Hu, and T. O'keefe, "Trust computing for social networking," In Proceedings of the 6th International conference on information technology: new generations, IEEE Computer Society, pp. 1534–1539, 2009.

[42] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," Journal of network and computer applications, vol. 42, pp. 120-134, 2014.