Introducing Conviviality as a New Paradigm for Interactions among IT Objects

Assaad Moawad and **Vasileios Efthymiou** and **Patrice Caire** and **Grégory Nain** and **Yves Le Traon** [1]

Abstract.
The Internet of Things allows people and objects to seamlessly interact, crossing the bridge between real and virtual worlds. Newly created spaces are heterogeneous; social relations naturally extend to *smart* objects. Conviviality has recently been introduced as a social science concept for ambient intelligent systems to highlight soft qualitative requirements like user friendliness of systems. Roughly, more opportunities to work with other people increase the conviviality. In this paper, we first propose the conviviality concept as a new interaction paradigm for social exchanges between humans and Information Technology (IT) objects, and extend it to IT objects among themselves. Second, we introduce a hierarchy for IT objects *social interactions*, from low-level one-way interactions to high-level complex interactions. Then, we propose a mapping of our hierarchy levels into dependence networks-based conviviality classes. In particular, low levels without cooperation among objects are mapped to lower conviviality classes, and high levels with complex cooperative IT objects are mapped to higher conviviality classes. Finally, we introduce new conviviality measures for the Internet of Things, and an iterative process to facilitate cooperation among IT objects, thereby the conviviality of the system. We use a smart home as a running example.

1 Introduction

Two decades ago, Mark Weiser coined the term ubiquitous computing. Ubiquitous computing "enhances computer use by making many computers available throughout the physical environment, while making them effectively invisible to the user" [22].

Today, microelectronic devices have become so small and inexpensive that they can be embedded in almost everything, making everyday objects "smart" [15]. The new paradigm of the Internet of Things (IoT) has emerged. The basic idea behind it is the pervasive presence around us of a variety of smart objects which, "through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to achieve common goals" [1].

Smart objects carry chunks of application logic. They sense, log, and interpret what is happening to them and the world, they act on their own, interact with each other, and exchange information with human users. They know what "has happened to them in the past" [15]. In this heterogeneous world, consisting of both human users and objects, social relations naturally extend to objects.

The concept of *conviviality*, defined by Illich as "individual freedom realized in personal interdependence" [12], focuses on the cooperative aspects of the interactions among humans. It has recently been introduced as a social science concept for multi-agent and ambient intelligent systems to highlight soft qualitative requirements like user friendliness of systems [5].

In this paper, we extend conviviality as a new paradigm for IoT Information Technology (IT) objects in two ways. First, convivial relations among IT objects and human users allow the latter to fulfill their needs for social interactions, and second, convivial relations among IT objects facilitate cooperation among participants. The aim is to enable knowledge sharing for the collective achievement of common objectives among entities which form various groups or coalitions [3]. The challenge of capturing social relations among IT objects breaks down into the following research questions: 1) How to distinguish the different kinds of social interactions of IT objects? 2) How to map the social interactions among IT objects to conviviality classes? 3) How to measure the conviviality of an individual IT object? and 4) How to use conviviality in the Internet of Things?

Tools for conviviality are concerned in particular with dynamic aspects of conviviality, such as the emergence of conviviality from the sharing of properties or behaviors whereby each member's perception is that their personal needs are taken care of [12]. In such dynamic circumstances, the conviviality of each participating member is a key criterion.

In [4], conviviality measures were introduced by counting, for each pair of agents, the possible ways to cooperate, indicating degree of choice or freedom to engage in coalitions. In this paper we build on these measures to define conviviality measures for each agent. Our coalitional theory is based on dependence networks [6, 19], labeled directed graphs where the nodes are agents, and each labeled edge represents that the former agent depends on the latter one to achieve some goal. Furthermore, in order to increase the conviviality of the system, we establish an iterative process through which the least cooperative IT objects are identified, then, upgrades for these objects are proposed to enhance their cooperations and increase their inclusions into more coalitions.

Our motivation lies in the vision that IT objects will be endowed with all the capabilities needed for a society of objects fully integrated into human society. In [14] smart objects differ from simple tracking objects such as RFIDs, in that they are autonomous physical/digital objects augmented with sensing, processing and networking capabilities. Here, we refer to both kinds of objects as IT objects.

The structure of this paper is the following: In Section 2, we provide the background for our IT object interaction classification, we then introduce our motivating example, in Section 3. We propose our mapping between IT objects interaction classes in Section 4, and the conviviality measures for individual IT objects in Section 5. We discuss these measures in Section 6 and present some related work in Section 7, and conclude in Section 8.

---

[1] Interdisciplinary Center for Security, Reliability and Trust, University of Luxembourg, email: firstname.lastname@uni.lu

2 IoT Evolution and conviviality issues

The *Internet of Things* relates to the interconnection of communication enabled-IT objects [9, 1]. IT objects from our everyday life are getting more communication abilities every day. TVs, phones or cars, are now able to share information and offer services to each other.

New services taking advantage of these communication links and shared data are emerging from these new abilities of IT devices. But the way toward seamlessly interacting devices and smart services is still long.

The miniaturization of hardware material for computation made it possible to introduce programs in electronic devices. That is how autonomous regulation devices made it possible to automate several household tasks and duties (e.g.: in Heating Ventilating and Air Conditioning (HVAC) systems). This automation of basic tasks resulted in an increase of the comfort and security for users. These autonomous regulation systems were the foundation of Internet of Things.

Along with the democratization of computers, the Internet and communication technologies, autonomous regulation systems got enriched with customization capabilities and sometimes remote accesses [18]. Now, users can specify the behavior of such configurable devices to enhance their own comfort and usage. Many devices have been equipped with bi-directional communications links, for reading and writing their configurations. With simple user interfaces, non electronic-specialists are now allowed to configure and/or remotely use their devices.
The availability of Internet everywhere and at any time, opens the door to remote accesses to IT devices, being at the office or at home. One can cite media centers, alarm or heating systems, or video camera for example. However, the configuration of such communication-enabled devices can sometimes turn into a nightmare for the uninitiated. As a consequence, protocols have been set up to allow automatic device recognition and connections. Also, zero-configuration devices[20] that are able to self-configure and get ready for use are more and more present in the IT environment.

The paradigm of *Cloud* tend now to get rid of the precise location to access a device, a service or a content. Resources can be accessed at any time, from anywhere and in several ways, with no idea about the precise location of this resource.

Today, Things (i.e. IT Objects) are able to communicate, are remotely accessible and are available from anywhere at any time [21]. But the services offered by these devices do not adapt or evolve with the presence of other services from other IT Objects. The next generation of Internet connected Things should be able to autonomously collaborate, adapt their behavior and services offered, according to their capabilities and to surrounding objects' capabilities and needs. They will participate at a time, in a community of devices by providing a new service, and integrate later another community as a backup for an already existing service.

Some classification or measures have to be developed to categorize these interactions among Things from a simple data provision to a collaborative decision making capability. As a social interaction measure, the conviviality can be applied to interactions between IT objects, and with humans, and provide a first set of tools for the next generation of smart devices. They could then be able to make more accurate decisions when adapting to their surroundings and evolving. They could be able to choose the community of devices to connect to by maximizing the benefit for both the community and themselves. They could even be able to improve their social involvement by acquiring new skills or taking charge of some duties.

3 Running Example: Smart-House

In this section we present a scenario of a smart-house automation system, regulating the temperature of a room. The IT devices in this scenario communicate, trying to figure out the cause of a heating problem. Such automation systems could be used to improve the energy-efficiency of a house and also reduce the cost of living. In similar ways, smart-home or smart-city automation systems could achieve a better quality of life, improved public services, ambient assisted living, or simply entertainment.

In our example, illustrated in Figure 1, we use five types of IoT objects that can communicate with each other; a refrigerator and its log, a heating system, door sensors and a phone. To accomplish their goal and find the source of the heating problem, the devices exchange information, query their logs and perform reasoning.

The heating system is responsible for keeping the room in a specified temperature at all times. However, in the last several minutes it has not been able to reach this temperature. Therefore it informs the refrigerator that it has problems heating the room (step 1). Like the heating system, the refrigerator is responsible for keeping its interior in a specified temperature. In other words, they have similar tasks. Hence, if the refrigerator has encountered a similar problem and solved it in the past, then there is a possibility that this solution could also work for the current problem of the heating system. Consequently, every time the heating system encounters a problem that it cannot solve, it "consults" the refrigerator.

The refrigerator receives and processes this transmission. It discovers from its log (step 2) that the last time it had a problem reaching a specified temperature, this was because its door was open. After its door was closed, the refrigerator could function properly, so this was a confirmed solution to this problem. Therefore, the refrigerator searches for a signal from the door sensors and receives that one of the house doors is open (step 3). The heating system is informed by the refrigerator that the problem comes from an open door (step 4).

Finally, the heating system stops functioning, until the problem is resolved (step 5). It also informs the phone that there is a heating problem and the recommended action is to close the door (step 6).

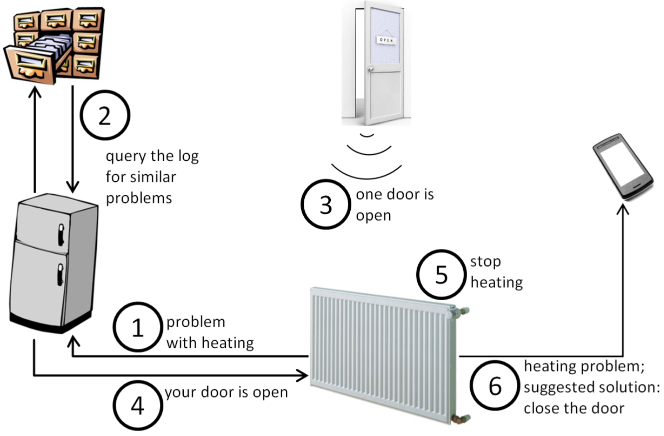


Figure 1. IT objects cooperate to solve a heating problem.

This is a typical example in Ambient Intelligence, where devices with different interaction capabilities have to cooperate. We now formalize the levels of this interaction, that we call *social interaction* of IT objects, by using the notion of conviviality.

4 IT Objects Classification and Mapping

In this section we discuss how IT objects can have a social interaction and how these interactions can be classified. For this classification, we use the notion of conviviality and Dependence Networks.

Definition 4.1 (Dependence networks) . *A dependence network (DN) is a tuple* $\langle A, G, dep, \geq \rangle$ *where:* A *is a set of agents,* G *is a set of goals,* $dep : A \times A \to 2^G$ *is a function that relates with each pair of agents, the sets of goals on which the first agent depends on the second, and* $\geq: A \to 2^G \times 2^G$ *is for each agent a total pre-order on sets of goals occurring in his dependencies:* $G_1 >_{(a)} G_2$.

Moreover, a Dependence Network can be represented by a directed graph, where the agents are the nodes of the graph, and the dependencies form the directed edges. For example, Figure 2 illustrates the graph that represents the Dependence Network derived from our motivating example of Section 3. Note that dependencies are potential, i.e, not all of them are actualized in our scenario. It should be also clarified that DNs are not equivalent to data flow networks; the latter model information exchange, not dependencies. The heating system h depends on the door d to be close, in order to function properly, but h does not have the capability to interact with d.

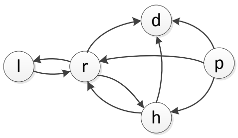


Figure 2. The DN of our example. h is the heating system, r is the refrigerator, l is the refrigerator's log, d is the door sensor and p is the phone.

Conviviality has been introduced as a social science concept for multi-agent systems to highlight soft qualitative requirements like user friendliness of systems [4]. The idea of conviviality is based on the notion of interdependency; *Cycles* denote the smallest graph topology expressing interdependence, and are considered as atomic relations conveying conviviality. When referring to *cycles*, we are implicitly signifying *simple cycles* (as defined in [8]), without repetition, with order and discarding self-loops.

In [4], conviviality is classified as presented in Figure 3, through a ranking of the DNs. Briefly, (W) is the worst class of conviviality because all agents are isolated. On the opposite side, (P) achieves perfect conviviality because the corresponding graph is a *clique*. For the in-between classes, (AWe) class has some dependencies but no cycles, (N) class has at least one isolated node and one cycle and in (APe) class, all the agents are participating in at least one cycle.

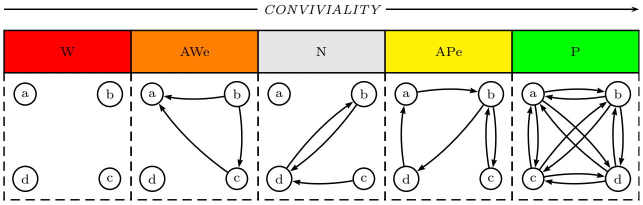


Figure 3. Conviviality Classes.

We use the term *social interaction* for IT objects, in a way similar to the human social interaction, as their ability to communicate with other IT objects and exchange information.

In Table 1, we illustrate the different levels of social interaction that an IT object can have. Level 0 IT objects are those who can only receive information from other IT objects. The phone, in our scenario belongs to Level 0, as it only receives alerts from other devices.

Level 1 is about the objects that only share their information with other objects. The door sensors are of Level 1, because they can only transmit the state of the doors to other devices.

Level 2 IT objects are programmed explicitly to interact with specific objects. The refrigerator's log is of Level 2, as it interacts only with the refrigerator. The heating system is on the same level. Level 2 is the current maximum social interaction level of IT objects.

Finally, Level 3 IT objects have the potential to interact with any other object, in order to achieve a goal. In our scenario, the refrigerator is of Level 3, since it is not explicitly programmed to interact with a specific set of devices.

We ignore IT objects operating only autonomously. The heating system could also work autonomously and try to keep a stable temperature in the room. However, its social interaction led to an improved, a more efficient functionality.

The social interaction level of IT objects can be associated with the conviviality of a network, in which they participate. To present this association in Table 1, we first suggest four possible DNs, each of them including at least one node of the specific level and then analyze the maximum conviviality in such a DN.

The maximum conviviality class of a network that includes a Level 0 or 1 object is N, since this node cannot be a part of a cycle. Level 0 objects have no incoming edges in a DN and Level 1 objects have no outgoing edges. For networks that include an object of Level 2, the conviviality cannot be better than APe, since such an IT object is not able to interact with every other device. Hence the graph of DN cannot be a clique. The maximum conviviality of a network that includes an IT object of Level 3 is P. Furthermore, P conviviality is achieved only if every node of the DN is a Level 3 IT object. W conviviality can exist if all nodes are Level 1. The maximum conviviality of a graph with only Level 0 and 1 objects is AWe.

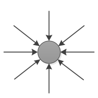
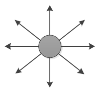
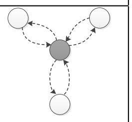
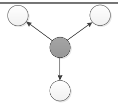
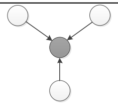
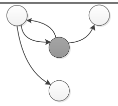
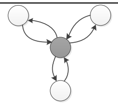
Interaction	0	1	2	3
Data				
DN				
max conv	N	N	APe	P

Table 1. Social interaction level of IT objects and maximum conviviality if at least one such object appears in a DN.

In this section, we have introduced a novel approach to classifying the social interaction of IT objects. We have mapped the social interaction level of an object with the maximum conviviality that can be achieved if this object is included in a DN. To do this mapping, we have established correspondences between DNs and IT objects interaction level. This way, we can have a maximum conviviality estimation, just by knowing the social interaction level of IT objects that participate in our system. However, it is sometimes necessary to get a more accurate measure to improve the conviviality of a system.

5 Conviviality Measures

Social network analysis has been providing many measures to reflect social interactions among agents [16]. However none of these considers cycles as basis. Our measurements meet the following requirements and assumptions.

5.1 Assumptions and Requirements

In this work, the cycles identified in a dependence network are considered as coalitions. These coalitions are used to evaluate conviviality for the network and for each agent.

In our second assumption, we consider the conviviality of a dependence network or a specific agent to be evaluated in a bounded domain, i.e., over a $[min; max]$ interval. This allows reading the values obtained by any evaluation method.

In terms of requirements, the first requirement for our conviviality measures concerns the size of coalitions. It is captured by the statement that larger coalitions are more convivial than smaller ones.

Our second requirement concerns the number of coalitions. It is captured by the statement that the more coalitions in the dependence network, the higher the conviviality of DN would be (all else being equal). Similarly, the more coalitions an agent is participating at, the higher its conviviality measure would be. This requirement is motivated by the fact that a large number of coalitions indicates more interactions among agents, which is positive in term of conviviality according to our definition based on interdependence.

5.2 Conviviality of a dependence network

The *conviviality of a dependence network* DN is defined in [4] as

$$\mathrm{Conv}(DN) \quad = \quad \frac{\sum_{a,b \in A, a \neq b} \mathrm{coal}(a,b)}{\Omega} \tag{1}$$

where $\mathrm{coal}(a,b)$ for any distinct $a, b \in A$ is **the number of cycles that contain both a and b in DN** and Ω is the maximum the sum in the numerator can get, over a dependence network of the same set of goals and the same number of agents but with all dependencies (fully-connected graph).

This way, the conviviality measurement of a dependence network which is a rational number in [0,1], can be used to compare different dependence networks, with 0 being the conviviality of a dependence network having no cycles at all (class W, class AWe) and 1 the conviviality of a fully-connected dependence network (class P).

However, this measurement just reflects the conviviality of the whole dependence network and does not allow to compare, inside the same dependence network, the conviviality of two different agents.

5.3 Conviviality of an agent

In this work, we extend the conviviality measures of a dependence network DN, by defining the conviviality of each agent inside DN. First, Let $C_{DN}(a)$ be the set of all cycles in DN that contains the agent a.

We define the *conviviality of an agent* $a \in A$ as

$$\mathrm{conv}_{\mathrm{DN}}(a) \quad = \quad \frac{\sum_{c \in C_{DN}(a)} (\mathrm{Len}(c) - 1)}{\omega} \tag{2}$$

where $\mathrm{Len}(c)$ is the length of the cycle c and ω is the maximum number the sum in the numerator can get, over a dependence network of the same size but with all possible dependencies (a clique). Moreover, ω is related to the Ω measured in Section 5.2 by the formula: $\omega = \Omega / |A|$ because of the symmetry between all agents in a clique.

This measurement per agent is also a rational number bounded in [0,1]. An agent participating in no cycle at all would have 0 conviviality, and all agents in a fully-connected dependence network would have a conviviality of 1.

Finally, the conviviality measurement for the whole dependence network defined in Section 5.2 can be deduced by calculating the average conviviality of all agents in the dependence network:

$$\mathrm{Conv}(DN) \quad = \quad \frac{\sum_{a \in A} (\mathrm{conv}_{\mathrm{DN}}(a))}{|A|} \tag{3}$$

5.4 Computation

We apply our computation on the dependence network of the running example illustrated in Figure 2. In this example, the set of all cycles is $C = \{(h, r), (r, l)\}$

The pairs participating in one cycle are $(h, r), (r, h), (l, r), (r, l)$ and there are no pairs participating in more than one cycle, thus the conviviality of the dependence network, according to Equation 1 is $\mathrm{Conv}(DN) = 4/\Omega$ with $\Omega = 980$ calculated over a clique of 5 nodes.

Now, to calculate the conviviality of each agent, we need to list the cycles containing that agent and applying Equation 2. We get:

$C_{DN}(h) = \{(h, r)\}$, $\mathrm{conv}(h) = 1/\omega$
$C_{DN}(r) = \{(h, r), (r, l)\}$, $\mathrm{conv}(r) = 2/\omega$
$C_{DN}(l) = \{(r, l)\}$, $\mathrm{conv}(l) = 1/\omega$
$C_{DN}(p) = \{\emptyset\}$, $\mathrm{conv}(p) = 0/\omega = 0$
$C_{DN}(d) = \{\emptyset\}$, $\mathrm{conv}(d) = 0/\omega = 0$
Where $\omega = \Omega/5 = 196$.

Note that, by taking the average of the convivialities of all the agents, we get $avg = (1/\omega + 2/\omega + 1/\omega + 0 + 0)/5 = 4/(5\omega) = 4/\Omega = \mathrm{Conv}(DN)$ as stated in Equation 3. Figure 4 shows our computation and the IT level of the objects of DN.

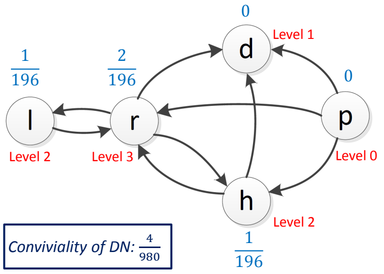


Figure 4. IT levels and conviviality measurements for the agents of DN.

As a conclusion, these measurements provide a way to compare agents to each other according to their social interactions and therefore they can be used to find potential improvements in the dependence network. For instance, in this example, we can deduce that agents d and p are the least convivial and can be seen as bottlenecks for the conviviality of DN.

6 Using Conviviality in IoT

6.1 Iterative Process

Improvement of the conviviality of a system is an iterative process. First, we identify the less participating agents in the network. Then, we try to involve them in more coalitions, which will increase their conviviality and consequently the conviviality of the system. If this solution is not applicable, then we upgrade these agents, when possible, to increase their participation. The overall conviviality of the system can thus be improved by iterating these steps.

6.2 Computation Examples

In the previous scenario, agents d and p are the least convivial and cannot do better because of their IT interaction levels of 0 and 1. We suggest as an alternate scenario S' to upgrade them by other IT objects (d' and p') having an IT interaction level of 2. But this is not enough. If the upgraded objects do not have the possibility to participate in more coalitions, the measurements will still remain the same, as the number of coalitions is unchanged. In the alternate scenario S', the smartphone p' (level 2) can have a more important role than just being a display device: it has a very good computation power and the ability to connect to the Internet to get updates and some information for example on potential solutions to a problem in the smart home context. In particular, the refrigerator and the heating system can potentially depend on its computation and connectivity capabilities. Figure 5 illustrates the dependence Network with the conviviality computation for scenario S'.

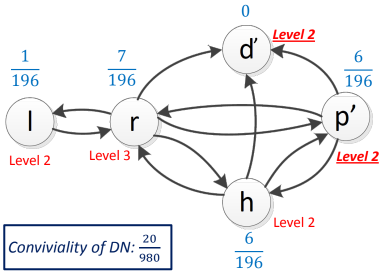


Figure 5. The alternate scenario S' with the new conviviality computation.

Comparing to Figure 4, we can deduce that conviviality of the refrigerator, heating system and the phone has improved. On the other hand, conviviality of the door and the log remain unchanged. Globally, the conviviality of the whole dependence network in S' has improved 5 times (4/980 to 20/980). Note that, the more dependencies we add to DN, the faster the conviviality increases, exponentially, to reach the 980/980 for a fully-connected DN because of the combinatorial nature of the measurements.

Finally, having the maximum conviviality is not always the best option for an IT system, because it might have other requirements and other constraints like security, privacy, efficiency, power management requirements, costs, etc. For instance, in a secured location, a smartphone might not be allowed to connect to the Internet for security reasons. In a camping context, a smartphone might not be able to do a lot of computations due to power saving measures. A good trade-off between conviviality and other requirements is the key to have a better system. Smart IT objects should be capable to adapt to different situations and contexts, selecting between different trade-offs accordingly in order to optimize their utility.

6.3 Conviviality as an Incentive

Conviviality can be used in agent theory to satisfy requirements on user-friendly systems and ensure that considerations such as the usability of a system get the same importance as the functionality.

In this section we discuss how conviviality measures can be used by agents as an incentive for cooperation, using a game-theoretic framework [17]. IT objects with a social interaction level of 3, as defined in Section 4, are not programmed explicitly to interact with specific objects. Depending on their needs, they have the ability to cooperate with any other object that will help them, or that needs to be helped. In order to decide on the form of cooperation, Level 3 objects have first to find out from which coalition they will benefit more, or, to which coalition they can contribute more.

The conviviality measures, introduced in Section 5, can be used to calculate the payoff of each agent participating in a coalition. Thus, agents have a formal way to calculate the gain that their participation in a coalition infers and therefore, decide which coalition to join.

A *co-operative game* is determined by a set Ag of agents wherein each subset of Ag is called a coalition, and a characteristic function V, assigning each coalition its maximum gain, the expected total income of the coalition (the so-called coalition value). The *payoff distribution*, P, assigns each agent its utility out of the value of the coalition it is member of in a given coalition structure [13]. In other words, P is the gain of the agent and V is the gain of the coalition.

The main idea behind these notions is to find out if the agents have an incentive to form coalitions. If the agents are not motivated to form coalitions, or if they find another, better coalition for them, then the current coalition is at risk; it is unstable. If the payoff of the agents is greater when they are in a specific coalition, than what they would gain otherwise, then that makes this coalition stable. We propose conviviality measures as a way to quantify this gain.

In our example, let's consider that the Level 3 refrigerator r is not yet a part of a network and it is trying to decide which of the networks, Figure 4 or Figure 5, to join. Then r can calculate what its conviviality would be if it joined each of these networks. It finds out that in the network of Figure 4 its conviviality would be $2/196$, whereas in the one of Figure 5 its conviviality would be $7/196$. Therefore, it decides to join the second network.

6.4 Computational Challenges

In our vision of the IoT, each IT object has the ability to act autonomously, in the sense of decision making. This means that IT objects can perform computations before joining a coalition, like the refrigerator in the previous paragraph. This is different from what usually exists today; a centralized system that makes all the computations. In today's systems, devices are usually programmed to interact only with the central computer and get these computational results, or request an available service.

The problem with our Level 3 objects is that smaller IT objects usually have a limited capability of processing. The computational complexity of our conviviality measurements is prohibiting such small devices to perform this calculation, especially as the number of agents in the network increases. This also limits the potential size of coalitions that can be created, since for larger coalitions it is harder to compute the conviviality in a reasonable time.

One possible solution is to revise these measurements and make them computationally easier for such devices. The new measurements should also meet the requirements introduced in [4]. Another approach would be to consider new definitions of conviviality.

7 Related Works

Many measures exist in graph theory domain that can be used to reflect the "social importance" of a node and the "structural importance" of a graph [10, 11, 16]. Some of the most relevant measures for a node are: *clustering coefficient* of the node which is the ratio of existing links connecting the node's neighbors to each other, to the maximum possible number of such links, *closeness centrality* of the node which is the reciprocal of the sum of distances to all other nodes in the graph. For a graph, we have the *clustering coefficient of the graph* which is the average of the clustering coefficients of all the nodes. However, these measurements do not take into consideration *cycles* in the graph. Our conviviality measures are based on the number and size of cycles in the graph which reflect interdependence.

In [7], Castelfranchi et al. use dependence networks to represent trust among agents. They build a socio-cognitive model of trust and present measurements for the degree of trust and trust capital.

The i-dud property [2] is a reciprocity property, saying that "an agent sees to a goal of another agent only if this enables it to obtain, directly or indirectly, the satisfaction of one of its own goals". This is also a desired approach for our goal-directed agents, similar to what we refer to as interdependency, or conviviality.

The notion, issues, and challenges of dynamic coalition formation (DCF) among rational software agents are introduced in [13]. For the formation of a dynamic coalition, the coalitions are represented by a coalition leader, who continuously attempts to improve the value of its coalition, by building re-configurations and suggesting them to its coalition members. This is different to what we vision for the future IoT, as discussed in 6.4, where there is no coalition leader. However, this approach could solve the computational issues that are discussed in the same section.

8 Conclusion

In this paper, we extend the social concept of conviviality as a new paradigm for IoT IT objects in two ways. First, convivial relations among IT objects and human users support the latter in fulfilling their needs for social interactions, and second, conviviality among IT objects facilitates their cooperation.

We first introduce a hierarchy for IT objects *social interactions*, from low-level one-way interactions to high-level complex interactions. Second, we propose a mapping of our hierarchy levels into dependence networks-based conviviality classes. In particular, low levels without cooperation among objects, are mapped to lower conviviality classes, and high levels with complex cooperative IT objects are mapped to higher conviviality classes. Third, we define new measures, since conviviality measures introduced in [4] are over the whole network, and do not differentiate among objects.

Fourth, in order to increase the conviviality of the system, we establish an iterative process through which the least cooperative IT objects are identified, then, upgrades for the identified objects are proposed to allow more cooperations among them, by increasing their inclusions into a greater number of coalitions. The process iterates to satisfy the system requirements, in which the tradeoffs among potentially conflicting requirements have been set, for example between conviviality, efficiency, privacy and security.

In future works, we plan to define the requirements needed for communications and negotiations among level three objects. We also want to provide a first set of tools for the next generation of smart devices and IT objects. More specifically, we plan to endow such objects with the capability of making more accurate decisions, for example, when adapting to their surroundings, and while evolving. Furthermore, we will focus on the capability for smart devices and IT objects to choose the community of devices and objects they may connect to. This choice may be guided by maximizing both their own benefit as well as their communities', i.e., the coalitions they belong to. Finally, we plan to enable devices and objects with the possibility to improve their social involvement through new skill sets acquisition and the adoption of new goals.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, 'The internet of things: A survey', *Computer Networks*, **54**(15), 2787–2805, (2010).

[2] Guido Boella, Luigi Sauro, and Leendert van der Torre, 'Algorithms for finding coalitions exploiting a new reciprocity condition', *Logic Journal of the IGPL*, **17**(3), 273–297, (2009).

[3] P. Caire, S. Villata, G. Boella, and L. van der Torre, 'Conviviality masks in multiagent systems', in *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12–16, 2008, Volume 3*, pp. 1265–1268, (2008).

[4] Patrice Caire, Baptiste Alcade, Leendert van der Torre, and Chattrakul Sombattheera, 'Conviviality measures', in *10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2–6, 2011*, (2011).

[5] Patrice Caire and Leendert van der Torre, 'Convivial ambient technologies: Requirements, ontology and design', *The Computer Journal*, **3**, (2009).

[6] C. Castelfranchi, 'The micro-macro constitution of power', *Protosociology*, **18**, 208–269, (2003).

[7] Cristiano Castelfranchi, Rino Falcone, and Francesca Marzo, 'Being trusted in a social network: Trust as relational capital', in *iTrust*, pp. 19–32, (2006).

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, The MIT Press, 2nd edn., 2001.

[9] *The Internet of Things - 20th Tyrrhenian Workshop on Digital Communications*, eds., Daniel Giusto, Antonio Iera, Giacomo Morabito, and Luigi Atzori, Springer, 2010.

[10] D. Gómez, 'Centrality and power in social networks: a game theoretic approach', *Mathematical Social Sciences*, **46**(1), 27–54, (August 2003).

[11] Robert A. Hanneman and Mark Riddle, *Introduction to social network methods*, 2005.

[12] Ivan Illich, *Tools for Conviviality*, Marion Boyars Publishers, London, August 1974.

[13] Matthias Klusch and Andreas Gerber, 'Dynamic coalition formation among rational agents', *IEEE Intelligent Systems*, **17**, 42–47, (2002).

[14] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, 'Smart objects as building blocks for the internet of things', *Internet Computing, IEEE*, **14**(1), 44–51, (2010).

[15] Friedemann Mattern, 'From smart devices to smart everyday objects (extended abstract)', in *Proceedings of sOc'2003 (Smart Objects Conference)*, pp. 15–16, Grenoble, France, (may 2003).

[16] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee, 'Measurement and analysis of online social networks', in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pp. 29–42, New York, NY, USA, (2007).

[17] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, July 1994.

[18] Timothy I Salsbury, 'A survey of control technologies in the building automation industry timothy i. salsbury', *Proc IFAC World Congress*, (2005).

[19] Jaime Simão Sichman and Rosaria Conte, 'Multi-agent dependence by dependence graphs', in *Procs. of The First Int. Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*, pp. 483–490. ACM, (2002).

[20] Daniel Steinberg and Stuart Cheshire, *Zero Configuration Networking: The Definitive Guide*, O'Reilly Media, Inc., 2005.

[21] International Telecommunication Union, 'The internet of things', ITU Internet Reports 7, International Telecommunication Union, (2005).

[22] Mark Weiser, 'Some computer science issues in ubiquitous computing', *Commun. ACM*, **36**(7), 74–84, (1993).