MAKING THINGS SOCIALIZE IN THE INTERNET – DOES IT HELP OUR LIVES?

Luigi Atzori, Antonio Iera, and Giacomo Morabito

University of Cagliari, Italy, l.atzori@diee.unica.it
University of Reggio Calabria, Italy, antonio.iera@unirc.it
University of Catania, Italy, giacomo.morabito@diit.unict.it

ABSTRACT

Current communication and computation technologies make it possible to embed intelligence and communication capabilities in most of the things *surrounding us; this leading to the* Internet of Things *(IoT) concept. To really exploit the potential of the IoT, objects and provided services should be easily discoverable and usable by humans and by other objects. Besides, trustworthiness of the billions of members of the IoT should be a key element in service selection. Existing solutions for service discovery in IoT do not scale with the number of nodes that is expected to be order of magnitude larger than in the current Internet. In this paper we propose to build a social network, that we name the Social Internet of Things (SIoT), that can be used to provide a navigable structure to the IoT. We also provide a framework that can be applied to socially tie things together and a preliminary architecture to be used as a baseline for the implementation of the SIoT. Our work demonstrates that standards should support establishment and management of federations of objects (ruled by social relationships) that represent "communities" of things in the SIoT.*

Keywords— Ubiquitous computing, Internet of Things, Social Networks.

1. INTRODUCTION

Current communication and computation technologies make it possible to embed intelligence and communication capabilities in most of the *things* surrounding us. This has led to the *Internet of Things* (IoT) concept. The IoT is *a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols* [1]. As such, the IoT can be seen as a *smart environment* paradigm and has the potential to radically change the way we interact with the environment and other people.

The IoT vision can be fully achieved only if objects are able to cooperate in an open way. Unfortunately, current implementations enable the cooperation among objects only if belonging to the same closed group. Sort of gateways are needed to allow specific groups of objects (aware of each others) to communicate and cooperate.

Necessary condition to reach the desired open cooperation is the ability of nodes to discover the available services they need. Solutions for *service discovery* can be classified as [2]

- **Centralized**: A server is deployed in the network where a *publish/subscribe* scheme is utilized. Obviously, such solution does not scale as the number of nodes in the IoT will be huge.

- **Request broadcast-based**: Service requests are broadcast throughout the network. Nodes containing the requested services will reply.

  Obviously such an approach is reliable, however, it is not scalable as it involves global broadcast of the requests which is highly inefficient when the network size increases. Furthermore, it involves processing even in nodes that do not have the required service, which causes a reduction in the overall system efficiency.

- **Advertisement-based**: Nodes that want to share a service broadcast information about it. Nodes that in the future might be interested in such a service store such *publish* messages in a cache. This approach has two problems. First: the size of the cache memory is expected to increase proportionally to the number of nodes in the IoT. Second: implementation of functionality that locate objects providing the required service is needed. Reliability of such a task becomes difficult to achieve if we introduce mobility into the picture.

In this paper we propose a paradigm for service discovery which makes use of social relationships between objects. We call this the *Social Internet of Things* (SIoT). More specifically, in the SIoT objects establish social relationships with each others. Objectives of such relationships are twofold:

- Give the IoT a structure that can be shaped as required to guarantee network navigability (see [3], for example) so that service discovery can be performed effectively while guaranteeing scalability.

- Create a level of trustworthiness which could be used to leverage the level of interaction between things that are friends (or friends of the friends, etc.).

The proposed approach is distributed and therefore is expected to guarantee higher scalability and better reaction to the frequent state changes.

Proliferation of embedded computing and communication devices in most objects surrounding humans is a necessary step towards the vision of a *fully networked human*. However, the risk exists that the number of such devices will soon become too large and scalability problems will emerge consequently. In this context, efficient cooperation between smart objects creating trusted, dynamic social-like communities contribute to solve the above scalability issues.

Furthermore, autonomous (without actions performed by human user) organization of objects in communities, cooperating towards the implementation of added value services, enables a paradigm shift to the vision of the fully networked human vision; it, in fact, supports the connection of human users with resources and services rather than nodes.

Observe that exploitation of social networks in the context of the IoT has been investigated in [4]. There, it was proposed to exploit (human) social network relationships to share the resources offered by a smart thing. More specifically, smart things support web services that can be used by friends of their owner. Online social networks can be used to authenticate and authorize *friends* of the thing's owner. In this context, also the idea of the possibility of associating social potentialities to smart objects, towards shaping a Social Internet of Things emerges from [5]. Notwithstanding, the authors' attention is more focused on envisaging a generic IoT architecture by integrating both RFID and smart object-based infrastructures than on defining social relationships among objects on which to base objects' interactions of a real social networks of smart objects.

Note that the approach we have in mind is different in two major ways:

- We are interested in establishing and then exploiting social relationships between things, not between their owners.
- We use social relationships so that things can crawl the IoT and discover services and resources.

The reasons and relative benefits of the foreseen SIoT are listed in Table 1 and compared with those that are recognized as the most important adavantages of the networks between humans.

With the institution of the *Internet of Things - Global Standards Initiative*[1] (IoT-GSI), ITU plays a crucial role in the standardization process of solutions for the IoT. Our contribution demonstrates that standards for the IoT should support the establishment and management of federations of objects (related by social relationships) that can interact in a more strict way and rely on each others for the execution of complex tasks. This, however, should not prevent the establishment of *loose* interactions between objects not related by social relationships.

Finally, we believe that standard guidelines for the interactions between different networks of social smart objects should be provided at the earliest steps of the deployment phase. In fact, in the context of online social networks the standardization process has begun late (compared to the deployment phase). Accordingly, today large part of the most popular social networks websites do not comply with standard-like guidelines such as those defined by the *OpenSocial*[2] initiative.

2. MAKING THE IOT SOCIAL

Basic idea of this work is the definition of a "social network of intelligent objects" – which we name the *Social Internet of Things* (SIoT) – in analogy with social networks of human beings, where the value of social relationships has been conceptualized in the so called *social capital* [6]. Bringing such a concept in the IoT would allow to successfully extend the use of models designed to study social networks [3] also to deal with IoT related issues (related to extensive networks of interconnected intelligent objects). The first issue to address is the definition of a kind of social behavior among objects. This may derive from observing typical information exchanges and possible interactions among smart objects, which are called to implement applications and services for the IoT.

Within the overall architecture we envision, the following tasks shall be fulfilled: (i) define a sort of notion of social relationship among objects, (ii) define the "degrees of social relationship" that can be established, (iii) study its evolution over time in a perspective of constant evolution and updating of the "Social Internet of Things", and (iv) investigate how this social relationship can be codified and supported by current technologies.

We address social relationships among objects in the IoT and the *degree* and dynamics of such relationships in Sections 2.1, 2.2, and 2.3, respectively. A preliminary study of the implementation of the above concepts through current technologies is provided in Section 3.

2.1. Social relationships between things

In defining the types of social relationships between objects we must consider that sociological studies have demonstrated that most value from social relationships can be gained when the structure of the social network is characterized by highly connected clusters which partially overlap with each others [7, 8]. Accordingly, as for human being, we first consider a "parental" form of socialization. In SIoT, what we define "Parental object relationship" is correlated to the membership of a set of objects to the same production batch and is established only among objects usually with the same nature and originated in the same period by the same manufacturer. Moreover, like humans do, objects can establish social

[1]See `http://www.itu.int/en/ITU-T/techwatch/Pages//internetofthings.aspx`.

[2]`http://www.opensocial.org`

Table 1. Reasons for which humans use and things may want to use social networks.

Reason for Humans	**Reason for Things**
Become visible/increase popularity	Publish information/services
Find resources/find old friends	Find information/services
Obtain context information and get filtered information	Get environment characteristics
Discover new resources and find new friends	Find new services/updated information

relationships whenever they come into contact to share personal (e.g. cohabitation) or public (e.g. work) experiences, named "co-location object relationship" and "co-work object relationship", respectively. These relations are determined whenever objects (homogeneous or heterogeneous) are either used always in the same place or collaborate to provide a common IoT application. An example of co-location object relationship is the case of different objects (sensors, actuators, etc.) used in the same environment to implement either home or industrial automation applications. Examples of cowork object relationship, by contrast, involve objects that do not have constant co-location relationship but are used together and cooperate for applications such as emergency response (sensors of body area networks, environmental sensors, etc.) and telemedicine.

A further type of relationship among objects is a consequence of their belonging to the same user (e.g., mobile phones, music players, game consoles) and the resulting high probability of interaction and data exchange with each other. We name this "ownership object relationship".

The last type of relationship is established when objects come into contact, sporadically or continuously, for reasons purely related to relations of *friendship* among their owners, which are in touch during their lives (e.g., devices and sensors belonging to friends who attend each other, classmates, travel companions, colleagues). We name this "social object relationship".

2.2. Degrees of social relationship

A classification of the "degree" and the "structure" of social relationships among objects is a prerequisite for the definition of adequate models of interaction, based on the nature of their relationships. This is also the basis for defining the type of information exchange among objects belonging to each structure. Again, we can draw inspiration from typical studies in the fields of Sociology, Anthropology, Social Psychology, or Cognition. Several activities in these fields start from a widely accepted classification of social relations proposed by Alan Fiske in his "relational models" theory [9] and studied, among others, by Nick Haslam in [10]. From these theories, four basic relational frames or structures derive from the four elementary models of the Friske's theory. In *Communal sharing* relationships, equivalence and collectivity membership emerge against any form of individual distinctiveness. *Equality matching* is based on egalitarian relationships characterized by in-kind reciprocity and balanced exchange. *Authority ranking* relationships are asymmetrical, based on precedence, hierarchy, status, command, and deference. *Market pricing* relationships, finally, are based on proportionality, with interactions organized with reference to a common scale of ratio values.

Next step is to relate these patterns of interaction among human beings to possible relational modes of smart IoT objects. Communal sharing can be definitely associated with behaviors of objects, not relevant individually but with only a collective relevance. For example, this type of relation is associated to objects forming a "swarm", according to which it is not important the service offered by the single object but the service that the entire swarm can provide to users. Equality matching may represent all forms of information exchange among objects that operate as equals in the perspective of providing IoT services to users while maintaining their individuality. While with objects in communal sharing relationship the service is associated to the whole group, in the second case every object has associated a service that it advertises. Authority ranking is a type of relationship established among objects of different complexity and hierarchical levels (such as, RFID reader and Tags, master and slave terminals in Bluetooth) exchanging information in a highly asymmetric fashion. In this case, the service advertised is usually associated to the whole group of objects (e.g., the whole coalition composed of master and slaves) or to the object of highest rank, which then coordinates those of lesser rank to provide the service. The last type of relationship, Market pricing, can be associated to objects have to work together in the view of achieving mutual benefit. In many IoT applications, this implies that the participation in this relationship is considered only when it is worth the while to do so. Table 2 relates different types of object relationships, relational models and object interactions between each other. Also exemplary families of applications belonging to each category are shown.

2.3. How Things socialize?

At this point, it is necessary to understand if occasions actually exist to establish the defined relationships among objects and when the established relationship likely changes.

A "parental object relationship" is easy to implement, because such a tight link can be created, for example, among objects belonging to the same production batch, directly during the item production. Surely this inter-object link will not change over time and is only updated by events related to the disruption/obsolescence of a given device. Suitable proce-

Table 2. Object relationships, relational models, and interactions

Category of “object relationship”	**Applicable relational model**	**Type of object interaction**	**Application examples**
Parental object relationship	Communal sharing Equality matching Market pricing	Swarm Balanced Cooperative	Best practice sharing
Co-location object relationship	Communal sharing Equality matching Authority ranking	Swarm Balanced Unbalanced	Environmental monitoring Building automation Industrial automation Data fusion Automatic identification of goods in storing area
Co-work object relationship	Communal sharing Equality matching Authority ranking	Swarn Balanced Unbalanced	Emergency and first responder deployments Data distribution Telemedicine Military applications Logistics
Ownership object relationship	Equality matching Authority ranking	Balanced Unbalanced	Remote control of devices Personal data storing and distribution Multimedia content fruition Infomobility and positioning
Social object relationship	Equality matching Authority ranking Market pricing	Balanced Unbalanced Cooperative	Personal data exchange Cooperative sharing and downloading Distribute gaming Cooperative and hybrid positioning

dures of relationship refresh, based on the periodical check of the existence and the functioning of a given friend object are required.

Also “co-location” and “co-work object” relationships are easily implementable, as the establishment of the social relationship among objects become part of the initialization/implementation of either a “location based application” profile or a “situation-based application” profile. Changes in this kind of relationships are more frequent. Surely, it may be dynamically updated based on the evaluation of parameters such as: time duration of either the co-location or the co-working, frequency of the interaction events, object reputation gained during last interactions, etc. All these parameters shall be monitored through suitable policies whose decision are based on flexible and rich object profile descriptions, and on digital reputation management policies. Maybe, the way to establish an “ownership object relationship” is the most natural to envisage. Associating one another all devices owned by the same user is in fact a common procedure performed by anybody to allow them to exchange data. A ownership object relationship is the logical generalization of this concept through a more complex device profile. Variations happen following either the natural obsolescence of owned objects or any change in the ownership.

The implementation of a “social object relationship”, may for example naturally follow the social interaction of human beings. Similarly to people exchanging their contacts (phone numbers, e-mail addresses, etc.), it is easy to implement ad-hoc procedures for the exchange of profiles of the devices they own. Actually, also the case in which objects episodically come into proximity may fall into this category of object relationship. In this latter case objects are transported by their owner in a given area and discover nearby devices with a profile of any interest: functional similarity, complementarities, same trademark, etc. The device, if properly authorized, may decide to establish a relationship with other objects, even transparently to the user, by exchanging the social profile. The driving idea is that a device with similar functional behavior may become a best practice to follow to solve a problem that could raise. The duration of these social relationships is ruled by policies exploiting ad-hoc defined metrics to measure the opportunity of maintaining a given link.

3. AN ARCHITECTURE FOR THE SOCIAL INTERNET OF THINGS

In Section 3.1 we first identify the components that can be distinguished in current *Social Network Services* (SNSs) used by humans. Then, we analyze how such components should change to implement the SIoT in Section 3.2. Finally, in Section 3.3 we provide an overview of a preliminary architecture implementing such components.

3.1. Components of SNSs

The definition of an architecture for SIoT should start from the analysis of the solutions currently adopted by the SNSs used by humans.

Unfortunately, we found that there is not a common reference architecture. Indeed, the number of the components and relevant functions may differ significantly from one implementation to another. A partial analysis in this direction is provided in [11] and [12]. On the left hand side of Figure 1, we provide a sketch of the resulting logical architecture.

Central part of the architecture is the *Profiling* of the member, who is asked at the beginning to describe himself/herself and will be adding further information and updates about his/her personalities during all the virtual/digital life.

Then, the uploaded profiles must be visible within the system (and externally) by a *Social graph* module that publicly or semi-publicly displays the connections between the member and their "friends".

Strictly linked to this module is the *Social presence* one that provides the users with the functionality to traverse the connections (e.g., to view profiles associated with the list of "friends").

Participation tools are then used to allow the users to keep in touch with the other members, such as: e-mail, instant messaging, chat rooms, blogs, message boards, telephony, videoconferencing, and others.

Each member is usually provided with tools for controlling his/her own visibility (search, profile viewing) and how he/she prefers to interact or be contacted by other entities. This is the *Relation control* tool.

Another important module is used in the current SNSs and is gaining more and more importance. It is the *Service API* component, which represents the interface that allow either third-party services to be included in the SNS (so that the user can benefits of additional services) or external sites to incorporate content into their services provided by the SNSs (e.g., Facebook and OpenSocial).

All the activities carried out through these modules are stored in a layer of metadata where the use of ontology and semantic web has become mandatory [13], with major initiatives already carried out, such as: the Friend-of-a-Friend (FOAF; www.foaf-project.org) project and the Simple Knowledge Organization Systems (SKOS) model.

On top of the components for SNSs shown in Figure 1, applications are developed, spanning from the creation of events to the online gaming, from the management of virtual farms to the collaborative creation of multimedia content (music, movies, logos).

3.2. Components of SIoT

We envision a system for making things socialize on the Internet which takes some major components of existing SNSs for humans as described in the previous subsection. However, some major differences must be introduced which are mainly due to the limited computing capabilities of smart things and the different objectives of SIoTs. On the right hand side of Figure 1 we show the SIoT main components. Underlined bold fonts have been used to highlight the differences with respect to SNSs.

At the center of the system three basic components can be envisioned:

- *ID management*: assigning an ID that universally identifies all the possible categories of objects in the real world is not an easy task. To maintain the current object identification schemes, we foresee the deployment of a system where existing mechanisms can interoperate. This can be made possible by adopting a simple XML-based protocol that allows for specifying the ID mechanism adopted other than the ID itself. This system should include at least: IPv6 addresses, Universal Product Code (UPC), Electronic Product Code (EPC), Ubiquitous code (Ucode), OpenID, URI.

- *Object profiling*: it includes static as well as dynamic information about the object. Objects should be organized in classes, where each class is defined on the basis of the main object features. In this context, appropriate classification strategies should be identified. A viable option could be to distinguish objects based on their computing and communication capabilities. Alternatives could be based on either the type of services they offer or the type of interfaces they implement.

- *Owner control*: the owner has to be able to define activities that can be performed by the object, the information that can be shared (and which other object is involved in the sharing), as well as the type of relationships to set up. Accordingly, specific policies need to be defined for every possible operation that can be performed by/on each member. To this purpose, different security and access control policy definition languages are already available and can be used [14]. Owner control includes the functionality of the Relation control component in the SNSs.

As to the other satellite components, in Figure 1 we report the most important[3]:

- **Service discovery**: this component replaces the Social presence. Service discovery is a fundamental component which is aimed at finding which objects can provide the required service in the same way humans seek for friendships and information in the SNSs.

- **Relationship management**: this is fundamental in the network of objects since these have not the intelligence of humans in selecting the friendships so that this intelligence needs to be incorporated in the SIoT. Objective

[3] Note that *Social graph* is not a major component of the SIoT functionality. Indeed, a social graph tool may still be implemented to allow humans to visualize their (and not only) objects relationships; this is, however, a minor functionality.

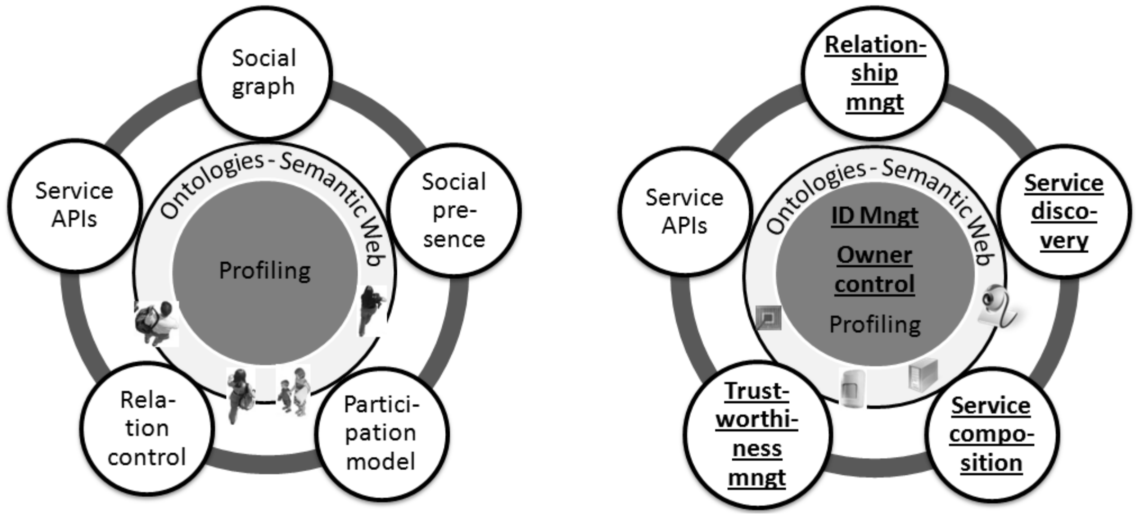

Figure 1. Basic components of social networks platforms for humans (on the left) and for objects (on the right)

of this component is to allow objects to start, update, and terminate relationships with other objects. The selection of which friendship to accept is based on the human control settings previously described. It is then driven by a set of rules that are defined on the basis of the way the objects get into contact with new objects in the physical and virtual worlds.

- **Service composition**: this component enables interaction between objects and replaces the Participation model. The interaction most of the times is related to an object that wishes either to retrieve information about the real world or to find a specific service provided by another object. Indeed, the main potentialities we see in deploying SIoT is its capability to foster the retrieval of information about the real world and services provided by other objects. Leveraging on the object relationships, the Service discovery provides with the way to find the desired service, which is then activated by means of this component. In general, when the requested service corresponds to the provisioning of information about the physical world, the service composition process can be performed according to either a *reactive* or a *proactive* approach. In the *reactive* approach, one of the applications developed on top of the SIoT triggers the request for an object providing a specific data. The service discovery components drive the discovery of the potential sources and then a reactive composition is performed to get the requested information. This is done by making objects interact by means of the available technologies for service composition (mainly according to either a RESTful or a SOA approach).

  In the *proactive* approach the source of information about the real world can directly expose the generated data (or metadata) on its own social network showcase so that every other member (or a subset according to the authorization policies) can directly acquire the information when needed.

  This component will also include the functionality of crowd information processing. This is aimed at processing the information obtained from different objects so as to obtain the most reliable answer to the information query on the basis of the different visions, similarly to what has been proposed in [15].

- **Trustworthiness management**: this is aimed at understanding how the information provided by the other members has to be processed. Reliability is built on the basis of the behavior of the object and is then strictly related to the relationship management module. Trustworthiness can be estimated by using notions well known in the literature such as *centrality* and *prestige*, which are crucial in the study of social networks [16].

- **Service APIs**: This component is analogous to the one required in SNSs.

To support the deployment of the model, a specific ontology is needed to record and represent the objects profiles, their friendships, as well as the relevant relationships. This has to be designed taking into account the objective of managing the relationships but also considering that the same ontology is used in the other components, especially for service discovery and trustworthiness management.

3.3. SIoT architecture

In this section we briefly introduce a possible preliminary architecture which is aimed to implement the SIoT components described in Section 3.2. We propose a system made of three main layers at the server side, as shown on the right hand side of Figure 2.

The *Base layer* encompasses the database for the storage and management of the data and relevant descriptors, the ontologies database, the semantic engines and the communications.

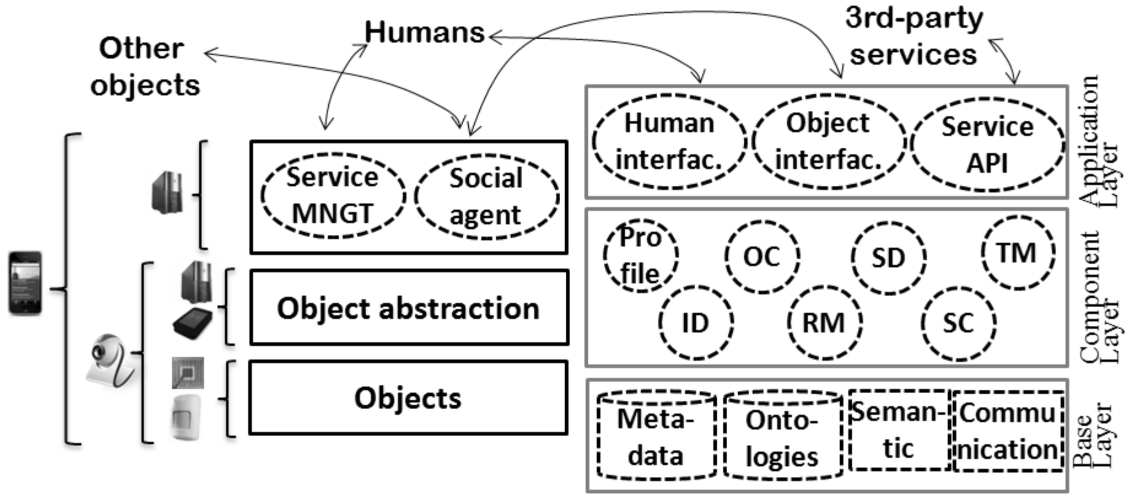

Figure 2. Architecture for the SIoT: client side (left) and server side (right). Acronyms stand for: OW - owner control, RM - relationship management, SD - service discovery, SC - service composition, TM - trustworthiness management

The *Component layer* includes the tools for the implementation of the functionalities described in the previous subsection: profiling, ID management, owner control, relationship management, service discovery, service composition, and trustworthiness management.

The *Application layer* is where the interfaces to the objects, the humans and the services developed by third-party are located.

This high-level architecture sketch may be mapped in a single site, deployed in a federated way by different sites or deployed in a cloud.

At the object side (see left side of Figure 2), the first layer of the architecture – named the *Object layer* – is where the physical objects are located and are reached through their specific communication interfaces.

Due to the fact that a large and heterogeneous set of objects can take part of the network, each one providing specific functions accessible though its own dialect, an *Object abstraction layer* is needed to harmonize the communication of the different devices through common language and procedure. Accordingly, there is the need to introduce a wrapping layer, consisting of two main sub-layers: the interface (upper) and the communication (lower) sub-layers. The first one provides an interface exposing the methods available through an appropriate interface. It is responsible for the management of all the incoming/outcoming (from /to the upper layer) messaging operations involved in the communication with the object. The second sub-layer implements the logic behind each service methods and translates these methods into a set of device-specific commands to communicate with the real-world objects. Some objects may be very elementary, such as an RFID-tagged object, while others may be equipped with an embedded TCP/IP stack, like TinyTCP, mIP or IwIP, which provide a socket like interface for embedded applications. In the first case a gateway is required to implement such abstraction layer, while in the second case this layer can be implemented in the object itself.

In the third layer, the *Social agent* is devoted to the communication with the SIoT servers to update its profile and friendships, as well as to discover and request services from the social network. It also implements the methods to communicate directly with other objects when they are close geographically or when the service composition needs direct communications between objects. Finally, the *Service management* represents the interfaces with the humans that can control the behavior of the object when communicating with the social network. The social management module and the social agent are usually implemented in an external server but sometimes can be located in the device itself when equipped with enough processing capabilities, as in the case of the smartphone.

4. NEEDS FOR STANDARDS

In our vision there will be more than one SIoT platform working in parallel, which should be interoperable not to limit the potentialities of a network of billions of things and not to require each thing to duplicate accounts, as it is currently happening to human users of social network services. At the first access of each new member, the owner interacts with the platform servers to create the account, insert the object profile data, set the control parameters. The thing then will start crawling the network to look for friends among the platform members (parental and ownership relationships) and managing the relationships when encountering other members (social and co-working/location relationships). The object will also make available its own services (e.g., information from the physical world) to the rest of the network. During these processes all the information related to the object profile, activity and relationships are then stored in the SIoT platform. Whenever an object wishes to move to another platform this information should be transferable in the new systems and this can be done only if a standard representation has been adopted, otherwise the object history is lost (*blocked* in the original platform). Additionally, an object can encounter another potential friend that is a member of a different platform. For the friendship to be created it is required that each platform exposes to the external world well-known functionalities to retrieve object identity, profile, and services.

These issues are already encountered in the social networks between humans, where the Facebook user profile and history cannot be exported entirely and the interaction with other platforms is limited. Lately with respect to the impressive market success of well-know platforms, some initiatives are going on towards the definition of common languages. The open social is a set of common application programming interfaces (APIs) for web-based social network applications, developed by Google along with MySpace and a number of other social networks. These APIs are intended to allow for accessing data and core functions on participating social networks. Other efforts are devoted to the definition of the syntax to describe people, the links between them and the things they create and do (FOAF; www.foaf-project.org). Similar efforts are needed in the SIoT context, and we hope that this time the standards will be available when the market will require SIoT services!

5. CONCLUSIONS

In this paper we have introduced the concept of the Social Internet of Things (SIoT) that can be exploited to implement scalable service discovery in the IoT. We have investigated how social relationships between objects can be established and managed. Finally, we have proposed a preliminary architecture which is able to implement the major components of the SIoT that have been identified by starting from the components that can be distinguished in current SNS used by humans.

The development of the SIoT concept requires a large research effort devoted to the study of the structure of networks of socializing things as well as to the detailed definition and assessment of the operations required to implement the components identified in Section 3.2.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, October 2010.

[2] D. Chakraborty, A. Joshi, Y. Yesha, and T. Finn, “Towards distributed service discovery in pervasive computing environments,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 97–112, 2006.

[3] J. Kleinberg, “The small-world phenomenon: an algorithmic perspective,” in *Proc. of ACM Symposium on Theory and Computing*, 2000.

[4] D. Guinard, M. Fischer, and V. Trifa, “Sharing using social networks in a composable web of things,” in *Proc. of IEEE PERCOM 2010*, March-April 2010.

[5] Anthony C. Boucouvalas Evangelos A. Kosmatos, Nikolaos D. Tselikas, “Integrating rfids and smart objects into a unified internet of things architecture,” *Advances in Internet of Things*, vol. 1, pp. 5–12, 2011.

[6] J. S. Coleman, “Social capital in the creation of human capital,” *American Journal of Sociology*, vol. 94, no. Supplement, 1988.

[7] R. S. Burt, *The Social Structure of Competition*, Cambridge, Massachusetts: First Harvard University Press, 1992.

[8] R. S. Burt, “Structural holes and good ideas,” *American Journal of Sociology*, vol. 110, no. 2, September 2004.

[9] A. P. Fiske, “The four elementary forms of sociality: framework for a unified theory of social relations,” *Psychological review*, vol. 99, pp. 689–723, 1992.

[10] Nick Haslam, “The four elementary forms of sociality: framework for a unified theory of social relations,” *Cognition*, vol. 53, pp. 59–90, 1994.

[11] D. M. Boyd and N. B. Ellison, “Social network sites: Definition, history, and scholarship,” *Journal of Computer-Mediated Communication*, vol. 1, no. 13, 2007.

[12] Mike Gotta, “Reference architecture for social network sites,” July 2008.

[13] J. Breslin and S. Decker, “The future of social networks on the internet,” *IEEE Internet Computing*, vol. 11, no. 6, pp. 86–90, 2007.

[14] D. Diaz-Sanchez, A. Marin, F. Almenarez, and A. Cortes, “Social applications in the home network,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 220–225, February 2010.

[15] A. Kansal, S. Nath, Jie Liu, and Feng Zhao, “Senseweb: An infrastructure for shared sensing,” *IEEE Multimedia*, vol. 14, no. 4, pp. 8–13, 2007.

[16] S. K. Bansal, A. Bansal, and M.B. Blake, “Trust-based dynamic web service composition using social network analysis,” in *Proc. of IEEE Workshop on Business Applications of Social Network Analysis*, 2010.