An adaptive formal metamodel for Semantic Complex Event Processing-driven Social Internet of Things network

Francesco Nocera and Angelo Parchitelli

Department of Electrical & Information Engineering, Polytechnic University of Bari
Via Orabona n.4, 70126, Bari,Italy

Abstract. Information, objects and people are the core innovation actors of human society progress. Their inner relations can be rebounded by the Internet, the Internet of Things (IoT) and social network, respectively. The integration of social networking concepts into the IoT solutions has led to the so called Social Internet of Things (SIoT) paradigm, according to the vision of a future world populated by intelligent objects that permeate the everyday life of human beings.
In this context we propose an adaptive formal model for a SIoT network driven by a Semantic Complex Event Processing where "things"are capable of establishing social relationships with respect to their owners, according to the monitoring of sensors value, changed behavioral properties, state and/or context variables and user's preference.

Keywords: Social Network, Internet of Things, Social IoT, Complex Event Processing

1 Introduction

The pervasive spread of physical devices, sensors and actuators, in general objects connected to the Internet is driving an exponential growth of Internet of Things (IoT) applications. The huge amount of data flowing though IoT networks poses a big issue related to the discovery of objects that are able to provide data by executing specific services. In this context, real world applications where streams of Big Event Data are the essence, include, but are not limited to, Smart City, Smart Home, Smart Transportation, Healthcare and Industry 4.0. Several approaches for "near"real-time search have been proposed.
The Smart Home is one of the focus area of the new IoT ecosystem era, both the centrality of the house in the life of every individual, with huge potential in terms of objects and distribution services, for both the bond with some of the leading sectors. Now, must be viewed as a dream where "things", particularly all home machines and more, are intelligible, locatable, addressable or controllable through the Internet.
The evolution of existing technologies and emerging standards is increasingly oriented to optimizing the use of mobile devices. This phenomenon is related to

a massive growing number of users accessing on Internet resources by a smartphone or tablet.
In parallel, the advent of the Internet and the mainstreaming of the world wide web in the early 1980s gave us the ideal technology to facilitate social networking. A social network is a community where the individuals are linked to each other by proximity, socially, by a common interest or goal. Examples of social networks include a family, a work friends and so on. When these groups connect for the purpose of communication and sharing, they are said to be "social networking".
When Social Networks meet the Internet of Things, the resulting paradigm is called *Social Internet of Things(SIoT)*. The idea to use social networking concepts in the IoT solutions to allow objects to autonomously establish social relationships is gaining popularity in the last years.
The IoT-generated data come in big amounts, are variable in terms of structure, often do not arrive at real-time, and could undermine the purpose of the services offered. This volume, speed and diversity make the storage and processing solution a very complex challenge. Traditional DBMSs, which need to store and index data before processing it, can hardly fulfill the concepts of timeliness and flow processing coming from such domains. Recently, new approach known as Complex Event Processing (CEP) emerged. CEP allows for efficient correlation, aggregation, and pattern matching of multiple distributed data streams on the fly.
In this paper we propose a Semantic CEP-driven SIoT network model to communicate and interact with smart things that humans use in daily life taking into account relevant aspects adaptation: context, users habits and profiles, information detached from external sources and sensors.
The proposed approach allows modelling and reasoning on complex adaptive architecture according to changed behavioural properties or context variables.
The rest of the paper is organized as follows: Section 2 provides a background and a review of related works, while Section 3 describes the proposed approach. Section 4 instantiates the model in Smart Home scenario and the last section concludes the paper and outlines future work.

2 Background and Related work

Social Internet of Things (SIoT). The idea of applying social networking concept in the IoT solutions to allow objects to autonomously fix social relationships has gained popularity in the last few years. The driving motivations are the following: *(i)* the SIoT structure can be shaped as required to guarantee the network navigability, so as that the discovery of objects and services is performed effectively and the scalability is guaranteed like in the human social networks; *(ii)* a level of trustworthiness can be established for leveraging the degree of interaction among things that are friends; *(iii)* models designed to study the social networks can be reused to address IoT related issues (intrinsically related to extensive networks of interconnected objects) [1].

Complex Event Processing (CEP). The concept of CEP was introduced by David Luckham in his seminal work [8] as a "defined set of tools and techniques for analyzing and controlling the complex series of interrelated events that drive modern distributed Information Systems (IS)."
As shown in Figure 1, CEP systems associate a precise semantics to the information items being processed: they are notifications of events happened in the external world and observed by sources. The CEP engine provides a rich set of concepts and operators for processing events, which include the CQL-like (Continuous Query Language), queries, primitive functions (aggregation, filtering, transformation, etc.), rules, showing derived events. A CEP workflow continually processes incoming events, explore and manipulates them, and outputs derived events that are delivered to sinks, which acts as event consumers. These output regularly represent notifications about detected interest situations. The handling of events are described by Event-Condition-Actions (ECA) -based CEP rules, that combine continuous query primitives with context operators (e.g. temporal, logical, quantifiers) on received events, checking for correlations among these events, and generating complex (or composite) events. [2]

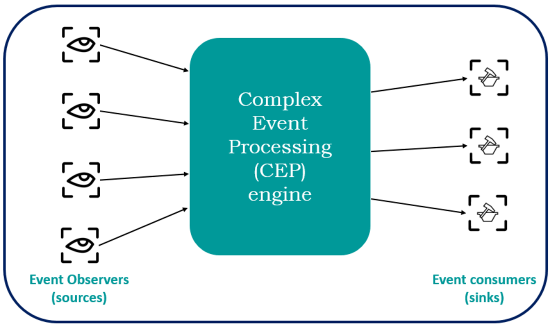

Fig. 1. The high-level view of a CEP system.

Related work. A first idea of socialization between objects has been introduced by Holmquist et al. in their work [5], where the focus was on solutions that enable smart wireless devices, mostly wireless sensors, to establish temporary relationships. The authors also analyse how the owners of the sensor nodes should control such a process and they propose *context proximity* for selective artefacts communication, using the concept of artefacts for matchmaking. In [3] the authors propose a a platform to cluster the Internet, the IoT and social network together. With the proposed platform, most macro elements can be tracked and summarized. The research activity in [9] introduces the idea of objects able to participate in conversations. The authors propose a balanced interaction between physical, social and virtual worlds, supported by the development of a data-centric architecture based on IP-driven opportunistic communications able to make useful data available to people when and where they really need it, augmenting their social and environmental awareness. An important step has been accomplished in [7]. In this work, the authors show how to empower physical objects to share pictures, comments, and sensor data via social networks. They also discuss about the implications of the so called "socio-technical networks" in the context of the IoT. The confluence of IoT and social networks has been considered in [4]. The proposed system enables individual to share the services offered by her/his smart objects with either her/his friends or their things. Kim et al. [6] propose relevant semantic metamodels for users, devices, locations and their relationships. Their proposed semantic metamodel enable interactions between users and devices based on the rules for relationship management, for basic automation, context generation, social gamification.

3 Proposed formal metamodel

In this section we describe the proposed formal approach to build a Social IoT network driven by a Semantic CEP engine to support the execution of domain-specific services. Modelling relies on behavioural/contextual changes and observable properties of the user's habits and profiles. The metamodel is made up of a inference level where incoming flows of information have to be processed to timely produce new flows as outputs (Sinks). The entities that create the information flows are called Information Sources. Sources and Sinks are interpreted in the sense of the CEP metamodel definition [8]. The events to be performed are derived from high-level properties, conditions about the state, context and sensor data.

Definition 1 (Event). *Event is a thing happening in a definite time and environment, that some social entity take part in and showing some action features. An event e can be defined as the following tuple:*

$$e ::= (A, A_c, T, E, A_{ss}, L_e),$$

where A is an agent, A_c is an action, T is a time, E is an environment, A_{ss} an assertions and L_e a language expression.

Intuitively, social entity (smart object or human agent) actions are the observations of sensor data, the publishing of a post and so on.

Definition 2 (SIoTN Ontology).

Social Internet of Things Network (SIoTN) Ontology formally specifies the shared and event classes. It can be defined as a quadruple formally.

$$SIoTNOntology =< Ec, Ei, R, e >$$

The elements in quadruples include the set of Entity classes, the set of Entity instances, the relationship $R =< Ec_i, Ei_j >$ (R includes parent-child, causal, follow and exclusion relations) and the correspondent event, respectively.

Thanks to the defined relationship between Entity instances it is possible to infer conditions between the entities, allowing the activation of a certain action.

Definition 3 (Context-aware SIoT Metamodel).
A Context-aware SIoTN Metamodel is a tuple

$$CaSIoTNM =$$

$$< S_c, CEP, SIoTNOntology, S_k, Actions, C_x, U_m >,$$

where S_c *and* S_K *are respectively the* sources *and* sinks *according to CEP definition; CEP is an instantiation of a Complex Event Processing (CEP) engine,* C_x *is the Context and* U_m *the user's model. The Contexts model every condition of the entity adaptation can have no effect on, e.g., instance, sensors data, geographical location, nearest street address, date/time, etc. Performed Actions are based on continual query and pattern matching verifying ontological conditions.*

Figure 2 shows the metamodel of the proposed approach.

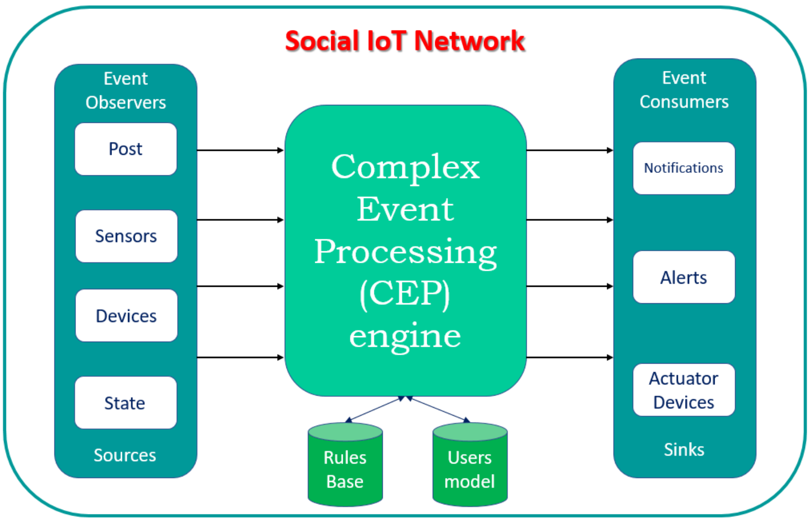

Fig. 2. The proposed metamodel.

3.1 Personalized action selection: Fuzzy *ECA* rules

In this section we present a set of definitions useful for characterize the selection of actions to be performed at runtime according to *(i)* changes in requirements, *(ii)* changes in state-context/environment, and *(iii)* user's habits and needs.
In most cases, a perfect match between the actual state and context and the

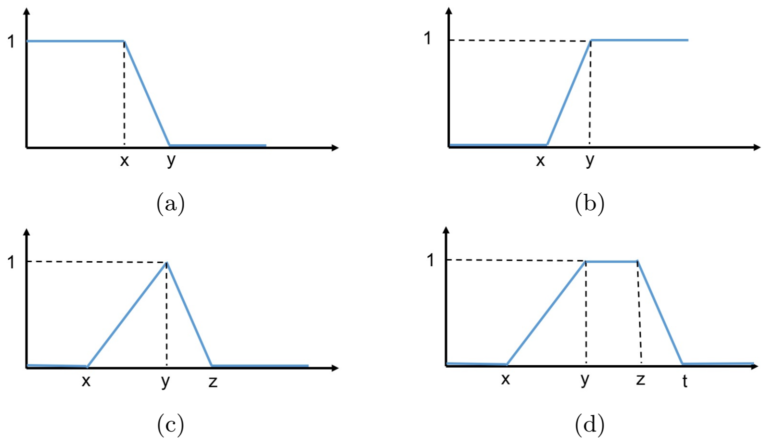

Fig. 3. Fuzzy Membership functions. (a) Left Shoulder function $ls(x, y)$, (b) Right Shoulder function $rs(x, y)$, (c) Triangular function $tri(x, y, z)$, and (d) Trapezoidal function $tra(x, y, z, t)$.

ones required in the condition is not to be expected. Given an ECA rule and a state-context $\tilde{S}$, *(i)* we need to evaluate if the state-context is "similar"enough to the one specified in the Condition (C part of ECA rule); *(ii)* we want to execute the Action (A part of ECA rule) whose condition C are more similar to $\tilde{S}$. Moreover, it would be advisable that the selection procedure by the Rule Manager behaves in a personalized way. That is, given $\tilde{S}$, the selection of the action to be executed may change depending on the user. Hence, when evaluating the condition C_i of an action A_i, most comparisons in C_i are evaluated as *fuzzy conditions* in Fuzzy Logic [10], an approach to computing based on "degree of truth "rather then the usual "true or false "boolean logic values.

This approach offer remarkable advantages when the set of conditions is huge. In fact, we can use this logic, based on the idea that the elements of the set are defined by degrees of membership in the set of values. The basic idea is that a physical quantity can assume not only boolean values, such as true or false, but a set of values indicating the degree of truth of a certain expression.

Definition 4 (Crisp Set). *A crisp set A is a collection of objects O_i taken into a universal set U.*

Definition 5 (Characteristic Function). *A characteristic function is a function that declares what elements of U belong to the set and which are not according to the following expression:*

$$\mu_A : x \in A \rightarrow \{ \textit{ 0,1 } \}$$

$$\textit{if } x \in A \rightarrow \mu_A = 1$$
$$\textit{if } x \notin A \rightarrow \mu_A = 0$$

Give a fuzzy set A, the proposition "x is a member of A"is not necessarily true or false but if it can be only partially true. The proposition is true with a degree of certainty which ranges between 0 and 1.

Definition 6 (Fuzzy Set). *Let U be a space elements of x. A fuzzy set A is defined as the set of pairs:*

$$A = \{(x, \mu_A(x)) | x \in U, \mu_A(x) : U[0,1]\}$$

A fuzzy logic has the aim to shape the real world no more as a binary world but in a more nuanced logic. The fuzzy logic is an alternative logic than to Aristotelica. In fact this logic rejects the Aristotelian principle of the excluded middle: if A is A, then A can not be not-A. The principle of excluded drastically defines the boundaries between opposites: high and low, good and bad, black and white, etc... In practice, the reality is less clear: not everything is high and not everything is low. For example, with the classic definition, once defined the hot temperature to above 18° C, the temperature of 17.9° C is defined as cold and not hot.
In literature exist several membership functions as depicted in Figure 3: the Left Shoulder function $ls(x, y)$, the Right Shoulder function $rs(x, y)$, the Triangular function $tri(x, y, z)$ and the Trapezoidal function $tra(x, y, z, t)$. The choice of the appropriate function is associated to a fuzzy set.
In this way, we can represents a general Fuzzy condition C based on *state* and *Context* information according to the following definitions.

Definition 7 (States). *Given a set of* State Variables $SVars = \{v_1, \ldots, v_n\}$, *and a set of corresponding domain values* $\{V_1, \ldots, V_n\}$, *a* State *is an assignment* $s : v_i \mapsto d \in V_i$ *that for each* $i = 1, \ldots, n$ *maps each state variable* v_i *to its current value* $s(v_i) \in V_i$.

An *Action* can change a state s to a state $\tilde{s}$, by changing the value of any number of state variables.
For example, a state variable *gps* of a mobile device may record in a boolean value whether the GPS is ON or OFF, another may record whether or not the present value of $Thermostat$, etc. Actions model the adaptations of the devices, e.g. turn OFF the GPS and reduce the temperature of autonomous heating. State variables not affected by an action keep their values.

Contexts are defined in the same way, but with the crucial difference that context variables cannot be changed by actions; the device environment changes them exogenously.

Definition 8 (Contexts). *Given a set of* Context Variables $CVars = \{w_1, \ldots, w_m\}$, *and a set of corresponding domain values* $\{W_1, \ldots, W_m\}$, *a* Context c_x *is an assignment* $c_x : w_i \mapsto d \in W_i$ *that for each* $i = 1, \ldots, m$ *maps each context variable* w_i *to its current value* $c(w_i) \in W_i$.

Definition 9 (Conditions). *We call* $\mathcal{C}$ *the language containing all possible conditions. For every action* a*, the* condition C *(part of ECA rule) is a formula* $C_a \in \mathcal{C}$ *which is a Boolean combination of*

- *comparisons of state variables* $(v_i \text{ op } d)$*, where* $v_i \in SVars$*,* $d \in D_i$*, and* op $\in \{=, \neq, <, >, \geq, \leq\}$ *is a comparison operator;*
- *comparisons of context variables* $(c_i \text{ op } d)$*, where* $c_i \in CVars$*,* $d \in W_i$*, and* op *is as above.*

Given a condition $C \in \mathcal{C}$, represented as a Boolean combination of *state* and *Context* variables comparison we now define how to evaluate its truth value.

Definition 10 (Interpretation). *An* ***interpretation*** $\mathcal{I}$ *for* $\mathcal{C}$ *is a function* $\cdot^{\mathcal{I}}$ *that maps each comparison of state variables* $(v_i \text{ op } d)$ *occurring in* P_a *to a truth value* $(v_i \text{ op } d)^{\mathcal{I}} = f(d)$ *and, analogously, each comparison of context variables* $(c_{xi} \text{ op } d)$ *to a truth value* $(c_{xi} \text{ op } d)^{\mathcal{I}} = f(d)$ *with* f *being a fuzzy membership function. Given* $C_a, C'_a \in \mathcal{C}$ *we recursively define the interpretation of a formula as:*

- $(\neg C_a)^{\mathcal{I}} = 1 - C_a^{\mathcal{I}}$
- $(C_a \wedge C'_a)^{\mathcal{I}} = min(C_a^{\mathcal{I}}, {C'_a}^{\mathcal{I}})$
- $(C_a \vee C'_a)^{\mathcal{I}} = max(C_a^{\mathcal{I}}, {C'_a}^{\mathcal{I}})$

With reference to the above definition, given a set of conditions $\hat{\mathcal{C}} \subseteq \mathcal{C}$ we can compute a total order among its elements by means of the interpretation functions. Indeed, given $C_a, C_b \in \hat{\mathcal{C}}$ we can always evaluate whether $C_a^{\mathcal{I}} \geq C_b^{\mathcal{I}}$ or $C_b^{\mathcal{I}} \geq C_a^{\mathcal{I}}$. Actually, an order among conditions can be easily reverted to a ranking among the corresponding actions. In other words, if $C_a^{\mathcal{I}} \geq C_b^{\mathcal{I}}$ we assume a is more likely to be executed than b.

Definition 11 (Executable Rule). *Let* $ECARR = \{\langle a, C_a\rangle, \langle b, C_b\rangle, \ldots\}$ *be an ECA Rules Repository, and* $t \in (0, 1)$ *be a threshold value. We say* a *is an* ***executable rule*** *iff both there is no rule* b *such that* $C_b^{\mathcal{I}} > C_a^{\mathcal{I}}$ *and* $C_a^{\mathcal{I}} \geq t$*.*

We may have more than one executable rule $a, a', a'', \ldots$. Indeed, it may occur that $C_a^{\mathcal{I}} = C_{a'}^{\mathcal{I}} = C_{a''}^{\mathcal{I}} = \ldots$. We see that as we do not have any order among $a, a', a'', \ldots$ we may execute any of them randomly. The reason why we introduce the threshold t is to avoid situations where the executed action has a low truth value (which corresponds to a high untruth value). Given a state-context $\tilde{C}$, in case there is no executable action, the CEP engine does nothing until the next change in $\tilde{C}$.

With respect to the metamodel presented in this section, we can encode preferences within the fuzzy membership functions. In fact, looking at Figure 3, we see they are defined in terms of a set of parameters x, y, z, t. By changing these values, we modify the shape of the functions. Let us go back to our example ($time = 13:30$) and suppose we define the *fuzzy set* associated to time by means of a triangular function with $y = 13:30$. We may distinguish between an "always on time" user and a "more relaxed" one by setting, for instance, in the former case $x = 12:25$ and $z = 13:35$ while in the latter case $x = 13:00$ and $z = 14:00$. Hence, the truth value associated to C_a may change and then the possible selection of a as executable action. It is noteworthy that x, y, z, t can be either be set manually or be automatically learned by collecting information about the user's behavior.

The history of the user's behavior is stored in an ontological model through the values of the context and state variables describing the actions generally performed by the user and the related preferences. A triangular function elicits the variables values to describe the user's behavior: for example, the history of the places she usually visits (in this case, the state variable is *position*), or the times he usually returns at home or he's going to another place. The CEP engine will choose among the pool of rules identified in the *ECA Rules Repository* the rule that verifies the constraint with the threshold t with respect to the triangular function of variable in the condition. In case of multiple properties in the condition expressed with a fuzzy variables the minimum or maximum operator as specified in the corresponding fuzzy interpretation.

To make the formulas fuzzy, we can express the conditions using intervals, that is, the condition is not true for only one value of the formula, but for the values in these ranges. For example, the choice of points to be displayed on a map will not be shown only for an exact value of the radius of the area, but depending on user habits, this value can be included in a interval. For example, y is the current position, while the interval $[x, z]$ defines the length of the circumference of the diameter to be displayed (centered at y).

4 Prototype instantiation of the metamodel

4.1 Smart Home

The metamodel proposed in Section 3 was instantiated in the domain of smart home. Living in a smart home brings many benefits, especially at the level of comfort, safety, energy saving and so on. Systems integration, with its endless customization possibilities, significantly improves the domestic liveability. In smart home scenario it is in fact possible to control each device with the touch of a button or in a completely automated way, and have full control of our home. The practicality of an automated system is also expressed in the possibility to program specific functions at predetermined times or store scenarios, namely a series of commands that are activated simultaneously, designed to meet the needs of a particular condition based on state and context variables.
Nowadays, also the proximity environment domain is being considered as an added value of most applications, especially in social environments. This phenomenon is widely observed spreading in the social sphere, thanks to the enormous spread of smartphones with GPS. Using the GPS connection, we can model as context variable the *social network user position.* For example, depending on the context (location, time) points and user's habits, when one user is returning home or is wakes up in the morning, devices such as autonomous heating or rolling shutter, can interact with the user activating the change of device

state, turn ON the autonomous heating system and pull up the rolling shutters, respectively.

4.2 Context-aware SIoT metamodel instantiation

The instantiated metamodel exploits the *goals*, the objective that the user expects. Table 1 summarizes, in this exemplifying version, how the elements in the tuple $CaSIoTNM$ are instantiated. Table 2 presents an example of the

Table 1. Instantiation of the elements in the tuple $CaSIoTNM$ for Smart Home scenario.

Element	Instantiation
S_c	Entities that create the information flows entering the CEP engine: GPS device, smart home objects, human agent post and so on.
CEP	tool that operates according to a set of *processing rules.*
SIoTNOntology	A formal naming and definition of the types, properties, and interrelationships of the entities that really exist according to Definition 2.
S_k	Recipient of output(s): Human agents, devices and actuators.
Actions	Performed actions based on the combination of continuous query primitives with context operators on received events, checking for correlations among these events.
C_x	Sensors in the external environments
U_m	User habits (time lunch, sleep), the places already visited, how many times they were visited, preferences about smart devices states, type of medium supply, etc.

Table 2. Example of (Fuzzy) ECA rule for Smart Home Scenario.

EVENT: Fire-alarm CONDITION: (*temperatureValue* >*thresholdofHeat*) and ()*duration* >*thresholdofDuration*) ACTION: Send Fire-alarm Post on social Network
EVENT: Recommend most used app at home CONDITION: ($gps = true$) and ($gps.precision < x_1$) and ($location = y_2$) and ($time > 19:00$) ACTION: Displays facebook, youtube, netflix, meteo apps
EVENT: Automatic ignition autonomous heating system CONDITION: ($gps = true$) and ($season = winter$) and ($radius > x_2$) and ($radius < z_2 - x_2$) and ($time = 18:50$) ACTION: Turn ON rolling shutter actuator

instantiated (fuzzy) ECA rules. State and context variables are sensors in the external environments, context variables available on the user mobile device, events extracted from the sensors. In this instantiation we constructed an OWL 2 ontology[1] to represent all the described knowledge related to the defined meta-model in Section 3 by using *Protégé Version 5.1.0*[2] editor. Figure 4 shows the class hierarchy graph of the instantiated ontology. CEP engine is responsible for observation, filtering, and pattern matching from data sources, based on the defined *ECA* rules, combining such notifications to sinks.

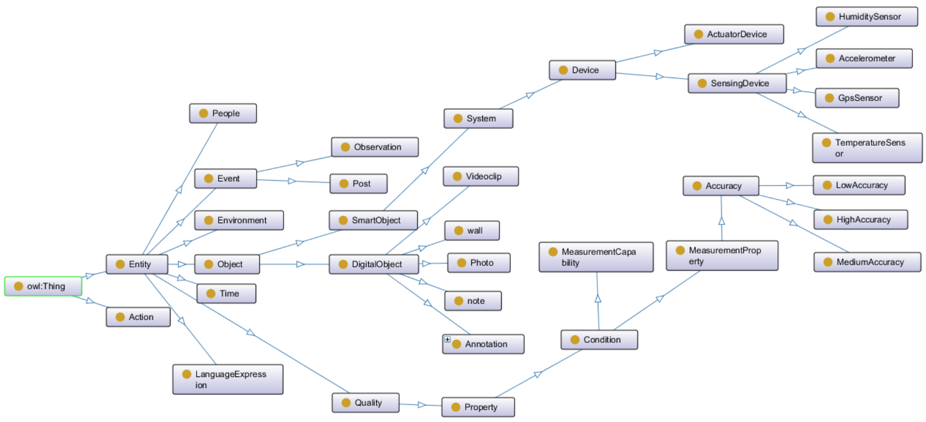

Fig. 4. Class hierarchy graph of the instantiated *SIoTNOntology*.

The proposed solution allows the communication between humans and smart devices to autonomously establish social relationships, providing automatic functionalities related to the home living, status change of smart home devices, include recommendations and so on.

5 Conclusion and future work

In the Big Data era, the huge amount of data flowing though IoT networks poses a big issue related to the discovery of objects that are able to provide data by executing specific services. The evolution of existing technologies and emerging standards is increasingly oriented to optimizing the use of mobile devices. The idea to use social networking concepts in the IoT solutions to allow objects to autonomously establish social relationships is gaining popularity in the last years. When Social Networks meet the Internet of Things, the resulting paradigm is called *Social Internet of Things(SIoT)*. In this paper we propose a formal Semantic CEP-driven SIoT network model to communicate and interact

[1] `https://www.w3.org/TR/owl2-overview/`
[2] `http://protege.stanford.edu/`

with smart things that humans use in daily life taking into account relevant aspects adaptation: context, users habits and profiles, information detached from external sources and sensors.
The proposed approach allows modelling and reasoning on complex adaptive architecture according to changed behavioural properties or context variables. We are currently implementing the prototype version by detecting all possible ECA rule in Smart Home Scenario. We see the presented model as a research vehicle to analyze the implications of socio-technical networks in the context of IoT, especially concerning the perception of these system in the eyes of human users. We guess that IoT, as it becomes part of our social lives, can help non-technical relations we continuously establish in a more technical advanced society and make our dependencies on systems more intuitive. We plan to additionally include wearable sensing system to monitor personal health data.

References

1. Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608, 2012.
2. Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):15, 2012.
3. Lianhong Ding, Peng Shi, and Bingwu Liu. The clustering of internet, internet of things and social network. In *Knowledge Acquisition and Modeling (KAM), 2010 3rd International Symposium on*, pages 417–420. IEEE, 2010.
4. Dominique Guinard, Mathias Fischer, and Vlad Trifa. Sharing using social networks in a composable web of things. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 702–707. IEEE, 2010.
5. Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *international conference on Ubiquitous Computing*, pages 116–122. Springer, 2001.
6. Ji Eun Kim, Adriano Maron, and Daniel Mosse. Socialite: A flexible framework for social internet of things. In *Mobile Data Management (MDM), 2015 16th IEEE International Conference on*, volume 1, pages 94–103. IEEE, 2015.
7. Matthias Kranz, Luis Roalter, and Florian Michahelles. Things that twitter: social networks and the internet of things. In *What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, pages 1–10, 2010.
8. David Luckham. *The power of events*, volume 204. Addison-Wesley Reading, 2002.
9. Paulo Mendes. Social-driven internet of connected objects. In *Proc. of the Interconn. Smart Objects with the Internet Workshop*, 2011.
10. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.