Semantic-based Social Intelligence through Multi-Agent Systems

Michele Ruta, Floriano Scioscia, Giuseppe Loseto, Filippo Gramegna, Agnese Pinto, Eugenio Di Sciascio
Polytechnic University of Bari - via E. Orabona 4, Bari (I-70125), Italy
{michele.ruta, floriano.scioscia, giuseppe.loseto, filippo.gramegna, agnese.pinto, eugenio.disciascio}@poliba.it

***Abstract*—Current technologies and market solutions are far from fulfilling the Ambient Intelligence (AmI) vision of simplified people-environment interactions. Even though Despite recent solutions based on Internet of Things (IoT) technologies provide the needed infrastructure, most approaches suffer from inadequate levels of intelligence and autonomy. This paper proposes a novel semantic-based Multi-Agent System (MAS) framework complying with the emerging Social Internet of Things paradigm devoted to improve both automation and adaptivity: device agents self-organize in social relationships, interacting autonomously and sharing information, cooperating and orchestrating ambient resources. A service-oriented architecture allows collaborative dissemination, discovery and composition of service/resource descriptions. Decision and choreography capabilities of software agents leverage Semantic Web languages at the knowledge representation layer and a mobile-oriented implementation of non-standard inferences for semantic matchmaking. Benefits of the proposal are highlighted through an AmI case study in the field of Home and Building Automation (HBA). A comparison with the state of the art is also provided.**

Index Terms*—Semantic Web of Things, Social Agents, Ambient Intelligence, Service Discovery

I. Introduction

The advent of Social Networking Services (SNSs) has had a deep impact on how people communicate and interact. Starting from personal user *profiles* containing general information, typical elements of SNSs include: the capability to engage asymmetrical (*e.g.*, *follower/followee*) or symmetrical (*e.g.*, *friendship*, *group*) relationships among users; a personal log (*wall*) to post text and/or multimedia items; the possibility to mark (*tag*) contacts to draw their attention to a given item, as well as to append *comments* and reactions (*e.g.*, *like*) to elements published by other users. These basic primitives can be combined to support several interaction models, granting users high flexibility in the way they share information, communicate, collaborate and search for resources of interest.

Endowing autonomous agents with social capabilities can transfer benefits of SNSs to Multi-Agent Systems (MASs), particularly to complex, dynamic and loosely coupled ones. This is the case of Internet of Things (IoT) contexts for Ambient Intelligence (AmI) [1], where networks of lightweight agents on highly heterogeneous mobile and embedded devices provide context-aware, adaptable, unobtrusive and intelligent support to users' activities [2]. In AmI, the environment should adapt to changes in external conditions as well as users' personal preferences and requirements, even anticipating needs and behaviors. Current solutions available for commercial and technical use cases are quite far from such levels of intelligence, automation and adaptivity. A limited flexibility is possible, as devices are logically associated at the application level by means of static profiles, defined during systems deployment. This is also the case of domotics –*i.e.*, Home and Building Automation (HBA)–, one of the widespread examples of AmI for environmental control. In most established HBA standards, solutions are centralized and proprietary: changing possible configurations or introducing new devices typically require the intervention of qualified practitioners. Even recent "smart home" platforms introduced by IT companies still depend heavily on manual configuration and provide only rudimentary levels of automation [3].

This paper presents a novel MAS paradigm at the convergence of the Semantic Web of Things (SWoT) [4] and Social Internet of Things (SIoT) [5] visions. IoT devices act as *socially intelligent agents*, capable of autonomous configuration, coordination and orchestration. Interaction patterns inspired by SNSs allow agents to establish relationships, share information, exchange requests and services, in a dynamic, decentralized and collaborative fashion. Agents exploit Knowledge Representation (KR) technologies borrowed from the Semantic Web to express and circulate knowledge about themselves and the context they are dipped in. In addition, the semantic-based matchmaking implemented in a resource-efficient mobile engine [6], on a moderately expressive fragment of the Web Ontology Language (OWL2) [7], supports the social intelligence through discovery, aggregation and ranking of available social entities. As agents acquire new knowledge about their context, both their configurations and the environmental services evolve: the MAS becomes a social network, where individual device interactions produce emergent behaviors toward high-level goals, without requiring explicit user commands. The paper reports on a case study in the field of HBA: current approaches are compared with the one proposed here in order to assess possible benefits and evidence the added value of the proposal.

The remainder of the paper is organized as reported in what follows: after related work discussion in Section II, Section III describes the proposed approach. An AmI case study is presented in Section IV to clarify the proposal, including a comparison with state-of-the-art technologies for IoT-oriented HBA. Conclusion in Section V closes the paper.

II. RELATED WORK

The study in [8] observed that the majority of SNS users access them regularly, as they find both enjoyment and usefulness. Higher numbers of connected users –and in particular *complementary* ones [8]– increase opportunities for finding needed information and services. These benefits can also apply to social networks of objects, which work as independent agents and interact for information and resource/service sharing.

One of the earliest examples of social object capabilities can be found in [9], a proposal aimed at distributed OWL Knowledge Base management and reasoning. Upon connection to the network, embedded devices proactively exchanged information in a handshake. "Requester" devices, endowed with reasoning capabilities, allowed users to execute queries, which were automatically distributed among requester's "known" devices. Unfortunately, reasoning capabilities were curbed by the restrictions of the adopted query language, limiting the practical interest of supported use cases.

The approach proposed in this paper is conceptually close to [5], where SIoT has been envisioned as a social evolution of the Internet of Things, with agentified objects capable of setting mutual relationships and exploiting them to exchange information and services, without requiring interactions with users or human-oriented SNSs. Conversely, earlier efforts such as [10] aimed to make objects aware of people's social context. Networks of socially intelligent objects were analyzed in [11], by defining key metrics about nodes and links, adapted from the literature on SNSs analysis. An ontology formalized the definitions, and social objects could use them to manage their policies, friends and reputation. A further step toward social agency is *object blogging* [12], *i.e.*, an object's ability to self-describe autonomously on the Web or in a local area network to support intelligent interactions. This was previously explored in RFID contexts [13] and constitutes an evolution of proposals requiring human intervention [14], [15]. The work in [16] identified smartphones as means to put people back in the loop of ubiquitous autonomous social MASs. Smartphones are ideal tools for learning about their owners and context, in order to work as their *digital counterparts*, exposing dynamic personalized profiles in the social agent choreography. Several works have already explored smartphones and wearable devices to model users' activities, preferences and contexts [17].

Semantic-based approaches are not uncommon in AmI and particularly in domotics. Building automation ontologies were used for system design and commissioning, device description, data modeling and access, ambient control [18]. The ontology-based system in [19] delivered context-aware customized information to different kinds of users. Queries matched device and user descriptions in OWL while rules implemented temporal and extra-logical constraints, achieving overall capabilities similar to Complex Event Processing (CEP) architectures. Nevertheless, integration appears as a serious limitation, because installing new devices required not only manual configuration, but also changes to the reference ontology. Also the proposals in [20], [21] relied on rule-based reasoning, where the system state should fully match rule conditions in order to trigger a rule. Unless only elementary rules are adopted, however, full matches are quite rare in realistic scenarios, where entities are described by heterogeneous and often contradictory annotations. Finally, in [22] a semantic Service-Oriented Architecture (SOA) enabled discovery and composition of semantic services. For greater autonomy and flexibility, in this paper the SOA paradigm has been coupled with a MAS of socially intelligent agents. The proposal extends the early conceptual and architectural elements introduced in [23].

TABLE I
NETWORK ENTITIES AND SOCIAL FEATURES

Technical feature	Social environment
Object / Device / Application	Social agent
Functional profile	Service
Object pairing	Social relationship (friend/follower)
Object communication	Social interaction
Object configuration update/adaptation	Distributed service discovery
Object log	Wall
Object command	Post
Object reply	Comment
Functionality activation/deactivation	Tag & Like

III. A NETWORK OF SEMANTIC AGENTS

The proposed approach aims at agent coordination in purposely infrastructured environments and particularly in domotics scenarios through interaction paradigms inspired by social networks. Devices are fully enabled in sharing resources/services, making decisions, disseminating requests and gaining responses through a distributed peer-to-peer protocol. Shared knowledge fragments about devices themselves, functional profiles and context are advertised via a decentralized service-oriented architecture. The social relationship and the discovery models outlined hereafter integrate in a unified social agent space both single-purpose physical objects and applications deployed on multi-purpose devices.

A. Framework and architecture for social agents

Table I highlights basic correspondences of entities and features in a generic AmI domain to the proposed social MAS environment. This applies particularly to domotics and HBA. Every object acts as a *social agent*: it exposes an individual profile describing its general features (*e.g.*, device type, location, hardware details) as well as the resources/services it can provide through possible configurations. An agent is able to become *friend* and/or *follower* of other agents. According to the different kinds of interactions described hereafter, it can write *posts* on either its wall or friends' walls when its settings or capabilities change, and also when it produces new or updated information after a context analysis. Each post contains perceptions and events observed by the social agent.

In the proposed SOA-based MAS, it is considered as a request for system reconfiguration through distributed semantic service discovery, which can be exploited by:

- sensor agents, such as a weather station, which can observe the environment and share data but don't have actuation capabilities;
- actuator agents, such as a lamp or a fan, which can react to environmental changes but have limited or no sensing facilities: by reading posts, they become aware of current conditions and activate/deactivate some services;
- smart agents, endowed with both sensors and actuators: if a smart agent does not have all the capabilities needed to comply with the perceived changes, a discovery process is started to find peers providing further suitable services, as described in Section IV-A.

Semantic annotations referred to ontologies in OWL2 [7] are used to express agent profiles, service descriptions and requests. Being formally grounded on Description Logics (DLs) semantics, they are both machine understandable and human readable. In particular, this paper refers to the OWL2 fragment corresponding to the $\mathcal{ALN}$ (Attributive Language with unqualified Number restrictions) DL, which supports standard and non-standard inferences with polynomial complexity [6].

Decision capabilities of social agents are enacted through a collaborative service/resource discovery. This process leverages semantic matchmaking, *i.e.*, the task aimed at retrieving and ranking the most relevant resources for a given request, where both requests and resources are satisfiable concept expressions w.r.t. a common ontology $\mathcal{T}$. Classic subsumption/satisfiability approach is extended here by means of the *Concept Abduction*, *Concept Contraction* and *Concept Covering* [6] non-standard inference tasks in $\mathcal{ALN}$:
- *Concept Contraction*: if annotations of a request R and a given resource S are not compatible (*i.e.*, an explicit clash arises from their logical conjunction), Contraction determines what part G (for *Give up*) of the request is conflicting with S. If one retracts G from R, a concept K (for *Keep*) remains, which is a contracted version of R compatible with S. G explains "why R and S are not compatible";
- *Concept Abduction*: if R and S are compatible, but S does not satisfy R completely, Abduction determines what should be hypothesized in S in order to obtain a full match. The solution H (for *Hypothesis*) to Abduction explains "what is requested in R and not specified in S". By computing *penalty* metrics linked to G and H [6], Contraction and Abduction further enable a logic-based relevance scoring of a set of resources w.r.t. a certain request;
- *Concept Covering*: in AmI scenarios such as domotics, it is often useful to compose multiple services/resources in order to satisfy a complex request. Given a request R and a set of resource instances $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, Covering finds out a pair $\langle \mathcal{S}_c, H \rangle$, where $\mathcal{S}_c \subseteq \mathcal{S}$ contains resources whose aggregation satisfies R as much as possible, while H is the (possible) remaining part of R not covered by concepts in $\mathcal{S}_c$.

The proposed MAS complies with a range of different scenarios and contexts, because it is inherently platform-independent and general-purpose. All social features reported in Table I can be modeled regardless of the particular application-layer communication protocol. Anyway, a first implementation has been proposed in [24] based on the Constrained Application Protocol (CoAP) [25], a lightweight Web of Things protocol for machine-to-machine interaction.

B. Semantic-based social network interaction

In the proposed framework, social agents are distinguished in two possible families: *full* ones, able to execute the inference tasks described above, and *basic* ones, endowed with low memory and low (or no) computing capabilities, which can only provide sensing/acting services, but cannot perform autonomous reasoning. A pair of agents can engage in two kinds of social relationships. Through the bidirectional *friendship* link, they can exchange both information and services. In particular, they became able to: (i) read and write on each other's wall; (ii) request the friend's service descriptions; (iii) activate or deactivate the friend's services. When becoming friend with a full agent, a basic agent can select it as *semantic facilitator*, *i.e.*, reasoning helper. Conversely, an agent can *follow* another one if interested only in receiving updates published on its wall, *i.e.*, becoming an observer through a unidirectional relationship.

Following/friendship criteria are automatically verified by means of a matchmaking process involving the device profiles. Two agents are good candidates for friendship if one or more of the following conditions are met: (i) strong *co-location*, *i.e.*, devices are placed in the same room/area; (ii) *parental* or *co-ownership*, *i.e.*, they are from the same manufacturer or belong to the same owner; (iii) *co-working*, *i.e.*, they are able to cooperate closely as they share annotations referred to the same ontology and provide functionalities related to the same activity (*e.g.*, room lighting) or observed parameter (*e.g.*, indoor temperature). On the other hand, a follower request is more appropriate in case of: (i) weak or sporadic co-location, such that information produced by an agent can still be useful to other ones to characterize their own context, but at the same time they need/prefer to start independent discovery requests; (ii) no co-ownership; (iii) weak co-working relationship, *i.e.*, direct interactions would have low usefulness, because *e.g.*, the two agent profiles are incompatible w.r.t. a common reference ontology, *i.e.*, they are significantly different (note that even a follower relationship is inappropriate in case of profiles referring to separate ontologies, as that implies agents belong to totally different domains, *e.g.*, HBA and healthcare).

For a broader range of interaction patterns, the framework also permits being both a friend and a follower of the same agent: this is useful in highly heterogeneous scenarios. In any case, a friendship/follow request can be rejected if the above conditions are not verified or the maximum number of friends/followers has been reached w.r.t. processing and memory limits of the invited agent. In practice, however, by enlarging its social network an agent increases opportunities for useful cooperation, hence rejections should be infrequent.

Like in human-oriented SNSs, agents' walls are the main knowledge sharing medium. The proposed framework supports both *push* and *pull* models, exploiting the above relationships:

- push: if agent A_i wants to receive updates from peer A_j automatically, it will ask to become a follower. If accepted, follower A_i will be able to start a distributed discovery session when it receives a notification of a new post or comment on the wall of the followed agent A_j;
- pull: if A_i wants to access A_j's wall on demand, it will ask to become a friend. By doing so, A_i will also automatically grant A_j access to its own wall. Then A_i will perform semantic matchmaking if A_j writes a post on A_i's wall, during a collaborative covering as reported in Figure 1.

Each agent will choose a model –or even both– based on its goals and strategies. These elements are relevant and conform to the general behavior policy of the MAS, anyway they are outside the scope of the paper.

When an agent detects an event (*e.g.*, a change in internal or environmental parameters) and conditions require adaptation –*i.e.*, modification to the functional configuration of itself and/or of nearby devices– it will write a post on its wall. As these trigger mechanisms are fundamentally domain-dependent and application-oriented, the framework does not prescribe specific solutions. In any case, the written post P will consist of a pair $\langle R, L\rangle$, where R is the request issued by the node –expressed as a semantic annotation w.r.t. a reference ontology– and L is the *like* value. The *like* reaction to a post has been mutuated from human-oriented SNSs, but in the proposed approach it is a real value in $[0, 1]$ instead of a Boolean value. It represents the coverage ratio of request R, as resulting from Concept Covering in the collaborative service discovery process. Specifically, if U is the uncovered part returned by the Concept Covering of R with a set of available services, the associated like value is computed as $L = 1 - \frac{norm(U)}{norm(R)}$ using the norm on concept expressions described in [6]. An example of the whole process is in Figure 1, composed of the following steps:

1) When an agent A_i detects a reconfiguration is needed, it writes a post P_i on its own wall. L_i is initialized to 0.

2) If A_i is a basic device, go to step 3. Otherwise, A_i executes the Concept Covering task on the local set of service annotations $\mathcal{S}$ (Section III-A). A_i activates the selected services and adds a *comment* C_i to P_i as a pair $\langle U_i, T_i\rangle$, where U_i is the uncovered part of R_i and T_i *tags* the selected local services/resources. Moreover, the value of L_i is updated as per the above formula.

3) If R_i is not completely covered, A_i selects a friend A_j and writes a post P_j=$\langle R_j, L_j\rangle$ on its wall. Particularly, if A_i has executed step 2, R_j is set to the uncovered part U_i, otherwise R_j is equal to R_i and L_j is 0. Writing P_j on the friend's wall automatically implies that A_i must be notified when a comment is added to the post. A_j recursively executes the steps 2) - 3).

4) When A_i receives the notification of P_j, it reads the comment from the friend's wall and appends it to P_i in order to update the status of the request. Finally, A_i updates the *like* value accordingly.

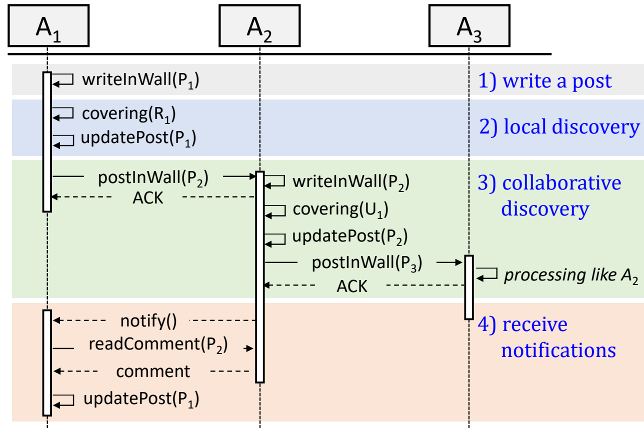

Fig. 1. Sequence diagram for a distributed reconfiguration

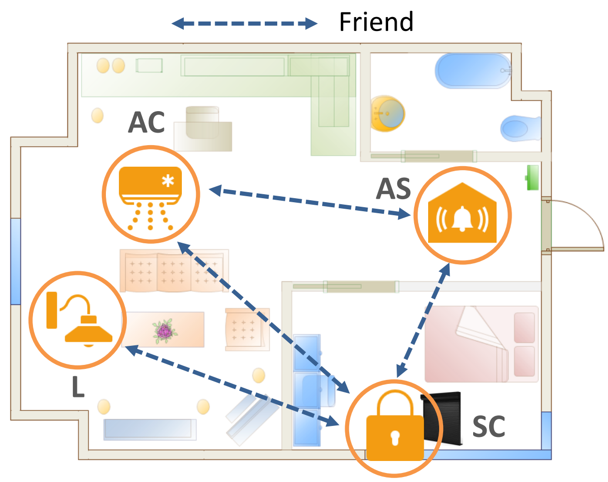

Fig. 2. Case study scenario

The choice of friend(s) to call in the above step 3 basically depends on heuristic preference criteria, such as the number and type of services exposed by the friend (known at friendship establishment time), network latency or friend's computational resources.

IV. Case Study: From Objects to Agents for Ambient Intelligence

The case study presented here would clarify the social and collaborative potentialities of the proposed MAS framework. To this aim, a specific scenario is targeted: the self-orchestration capability of agentified home devices allows to evidence the AmI capabilities of the above approach.

A. Illustrative example

Figure 2 depicts the reference testbed recalling the case study; a house contains a social network of semantic-enabled

AS_Request: (detectsOutdoorLuminosityCondition **some**) **and** (detectsOutdoorLuminosityCondition **only** LowLuminosityCondition) **and** (detectsIntrusionEvent **some**) **and** (detectsIntrusionEvent **only** IntrusionEvent) **and** (detectsOccupancyCondition **some**) **and** (detectsOccupancyCondition **only** (**not** OccupantPresence))

Fig. 3. Request posted by the alarm system AS

Full_Close: (detectsPrecipitationCondition **some**) **and** (detectsPrecipitationCondition **only** Rain) **and** (detectsWindCondition **some**) **and** (detectsWindCondition **only** StrongWind) **and** (detectsIntrusionEvent **some**) **and** (detectsIntrusionEvent **only** IntrusionEventForShutter) **and** (detectsOccupancyCondition **some**) **and** (detectsOccupancyCondition (**not** OccupantPresence))

Half_Close: (detectsPrecipitationCondition **some**) **and** (detectsPrecipitationCondition **only** (**not** Rain)) **and** (detectsWindCondition **some**) **and** (detectsWindCondition **only** ModerateWind)

Open: (detectsPrecipitationCondition **some**) **and** (detectsPrecipitationCondition **only** (**not** Rain)) **and** (detectsWindCondition **some**) **and** (detectsWindCondition **only** LightBreeze) **and** (detectsOutdoorLuminosityCondition **some**) **and** (detectsOutdoorLuminosityCondition **only** HighLuminosityCondition)

Fig. 4. Shutter controller SC service annotations

agents embedded in the following devices: an alarm system (AS), a rolling shutter controller (SC), an air conditioner (AC) and a dimmer lamp (L). The blue arrows in Figure 2 specify the existing friendship relations between the above agents.

According to the criteria suggested in Section III-B, the agents set friendship relations because they are in the same location and share functionalities useful to improve comfort or security in the house. Not all agent pairs are friends: in particular, Figure 2 shows L befriends SC only. Besides, each agent has embedded sensing and/or actuating capabilities and exposes a set of functional profiles to its friends.

Let us suppose *it is evening and AS detects an intrusion while there is nobody in the house*. AS writes a new post on its wall, representing what it has sensed as an OWL2 annotation. Figure 3 shows a formalization of the post in OWL2 Manchester syntax [26]. Service requests and descriptions are expressed w.r.t. the reference ontology (not reported due to space constraints), by specifying the context conditions suitable for the activation of a given service. Then AS starts a *Concept Covering* process using the content of the post as request, while services are taken from AS's cache of available functionalities exposed by all its direct friends, *i.e.*, SC and AC. Freshness of cache entries is checked via preliminary conditional requests: a service annotation will be retrieved again only if it has been updated, otherwise AS can directly use the cached copy. This procedure guarantees the covering task is performed using the latest descriptions of all available services.

According to the semantic service descriptions

Lamp_On: (detectsOutdoorLuminosityCondition **some**) **and** (detectsOutdoorLuminosityCondition **only** LowLuminosityCondition) **and** (detectsIntrusionEvent **some**) **and** (detectsIntrusionEvent **only** IntrusionEventForLamp)

Lamp_Medium: (detectsOutdoorLuminosityCondition **some**) **and** (detectsOutdoorLuminosityCondition **only** MediumLuminosityCondition) **and** (detectsOccupancyCondition **some**) **and** (detectsOccupancyCondition **only** OccupantPresence)

Lamp_Off: (detectsOutdoorLuminosityCondition **some**) **and** (detectsOutdoorLuminosityCondition **only** HighLuminosityCondition) **and** (detectsOccupancyCondition **some**) **and** (detectsOccupancyCondition **only** (**not** OccupantPresence))

Fig. 5. Dimmer lamp L service annotations

listed in Figure 4, Concept Covering selects only the *Full_Close* service, provided by SC: this is basically due to commonality with the request of concepts (detectsOccupancyCondition **some**) and (detectsOccupancyCondition (**not**OccupantPresence)) (service descriptions provided by the air conditioner are not reported because it does not offer any useful feature). AS comments its post including both a tag to the *Full_Close* shutter service and the uncovered part of the request. In order to further cover the post, AS can select one of its friends and forward the uncovered part. Since SC has provided the highest contribution in the first covering step, AS posts on SC's wall the OWL2 annotation of the uncovered part, reported in Figure 6. When SC receives the message, it recursively starts a covering process, which involves the services exposed by its friend L (Figure 5). The Covering inference task selects the *Lamp_On* service, which completely covers the remaining part of the initial request. SC therefore comments the post on its wall by tagging the activated service and updating the like value to 1. Further agents do not need to be involved, as the request is fully satisfied. Finally, AS receives a notification of the comment to its post on SC's wall, it reads the comment and sees the initial request has been completely fulfilled. As a consequence, it updates the like value of the post on its wall and the discovery process stops. The house has changed its configuration by closing shutters and switching the lamp on reacting to the intrusion alert.

AS_Req_Uncovered: (detectsOutdoorLuminosityCondition **some**) **and** (detectsOutdoorLuminosityCondition **only** LowLuminosityCondition) **and** (detectsIntrusionEvent **some**) **and** (detectsIntrusionEvent **only** IntrusionEventForLamp)

Fig. 6. OWL2 annotation of the uncovered part of *AS_Request*

It is useful to point out that the Intrusion class was defined as more specific than both IntrusionForLamp and IntrusionForShutter, *i.e.*, it should require services both from a lamp and a shutter controller. Such a modeling pattern allows activating functionalities (*Full_Close* and

TABLE II
COMPARISON WITH CURRENT IOT-ORIENTED FRAMEWORKS FOR HBA

Features	**KNX IoT**	**IzoT Platform**	**Dog Gateway**	**Eclipse SmartHome**	**Proposed Approach**
Home Area Network reference protocol	EIB/KNX	LonTalk	multi-protocol over HTTP	multi-protocol over HTTP	multi-protocol over CoAP
Network architecture	centralized	centralized	centralized	centralized	full P2P
Network/devices configuration	via ETS software	via LonBuilder software or XML configuration files	XML configuration files	Domain-Specific Language (DSL) configuration files	autonomous social agents configuration
Add/remove devices	edit network/devices configuration	edit network/devices configuration	edit network/devices configuration	edit network/devices configuration	agents self-configuration
Multi-protocol communication	KNX gateway	IzoT gateway	Dog gateway	node acting as gateway	smart agents acting as gateways
Device binding	static, defined during network configuration	static, defined during network configuration	dynamic, based on device profile	static, defined during device configuration	dynamic, based on friendship relationships
Scenarios configuration	static, defined during network configuration	static, defined during network configuration	dynamic, exploiting rule-based reasoning	static, based on an ECA rule engine	dynamic, exploiting non-standard inferences
Service composition	no	no	no	no	yes, through distributed covering
Message data format	proprietary (KNX specs.)	proprietary, XML-based (LonTalk specs.)	OWL 2	XML	OWL 2
Standardized framework interface	KNX IoT Web Services	HTTP RESTful API	WebSocket and HTTP RESTful API	HTTP RESTful API	CoAP RESTful interface

Lamp_On) of different devices that are fired when the same event is detected.

The above example has been kept simple for the sake of clarity, with relatively short service annotations and purely reactive MAS behavior. Notwithstanding, the adopted inferences allow managing more articulated specifications with detailed constraints. Moreover, the proposed approach fully supports proactive agents, which can fire periodic or sporadic internal events to trigger collaborative service discovery and MAS configuration updates. Finally, the small MAS described in the example can be federated with other MASs in nearby zones (*e.g.*, of adjacent houses) by means of social interaction capabilities, ensuing from the possibility to establish friendship or follower relationships between agents across zones. This allows taking advantage of sensing/acting capabilities of a larger agent pool, as well as compensating possible deficits of individual agents and zones reaching a concrete ambient intelligence in real-life significant scenarios.

B. Evaluation

In order to assess both peculiarities and capabilities of the proposed semantic-based social MAS, a systematic comparison with existing IoT-oriented AmI approaches has been carried out. Particularly, HBA platforms have been selected as reference systems. In more detail, the following solutions have been considered: KNX IoT[1]; IzoT Platform[2], originally developed by Echelon Corporation for the Industrial IoT but also exploited for HBA applications; Dog Gateway[3] [20]; Eclipse SmartHome[4].

Table II highlights most relevant elements: it emerges that, to the best of our knowledge, only the approach proposed here fully complies with resource-constrained scenarios (by supporting a P2P architecture and lightweight protocols such as CoAP). Another distinguishing feature is a certain expressiveness in the possibility of device description and modeling (by adopting semantically rich formalisms as OWL 2). Finally, noteworthy is the support for an articulated discovery through both exact and approximated matches formally grounded on service/resource composition.

Quantitative performance results of the proposed approach are not provided here, but the semantic service discovery and orchestration based on Concept Covering is arguably the most computationally demanding task, while social relationship management is not resource-intensive. Results obtained in [23] for an earlier version of this framework allow optimistic expectations about feasibility on IoT device networks and compatibility with performance requirements of HBA and AmI scenarios.

V. CONCLUSION

The paper proposed a novel semantic-based social MAS framework. Though presented in a HBA scenario, features and approach are general-purpose and target several possible Ambient Intelligence records. The application domains are basically inherited from ontologies modeling the reference implementation.

The proposed approach enables autonomic agent interaction and a semantic-enhanced service/resource discovery grounded on the formal annotation of devices, environment and phenomena. A case study and a comparison with state- of-the-art techniques help highlighting peculiarities of the proposal.

Future work will include further investigation and extension of the social presence capabilities of agents, as well as novel interaction patterns. A full prototypical implementation is expected to evidence possible optimization directions and scalability concerns. Finally, graphical visualizations of devices' walls are being implemented.

[1] http://www.knx.org/knx-en/Landing-Pages/KNX-IoT
[2] http://www.echelon.com/izot-platform
[3] http://dog-gateway.github.io/
[4] http://www.eclipse.org/smarthome/index.html

Acknowledgements

This work was supported in part by Italian PON project ERHA (Enhanced Radioteraphy with HAdrons).

References

[1] C. Savaglio, G. Fortino, M. Ganzha, M. Paprzycki, C. Bădică, and M. Ivanović, "Agent-Based Computing in the Internet of Things: A Survey," in *International Symposium on Intelligent and Distributed Computing*. Springer, 2017, pp. 307–320.

[2] F. Sadri, "Ambient intelligence: A survey," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, p. 36, 2011.

[3] W. Kastner, L. Krammer, and A. Fernbach, "State of the Art in Smart Homes and Buildings," in *Industrial Communication Technology Handbook*, 2nd ed., R. Zurawski, Ed. CRC Press/Taylor & Francis, 2014, ch. 55, pp. 1415–1434.

[4] M. Ruta, F. Scioscia, and E. Di Sciascio, "Enabling the Semantic Web of Things: Framework and Architecture," in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 345–347.

[5] L. Atzori, A. Iera, and G. Morabito, "From "Smart Objects" to "Social Objects": The next evolutionary step of the Internet of Things," *Communications Magazine, IEEE*, vol. 52, no. 1, pp. 97–105, 2014.

[6] F. Scioscia, M. Ruta, G. Loseto, F. Gramegna, S. Ieva, A. Pinto, and E. Di Sciascio, "Mini-ME matchmaker and reasoner for the Semantic Web of Things," in *Innovations, Developments, and Applications of Semantic Web and Information Systems*. IGI Global, 2018, pp. 262–294.

[7] B. Parsia, S. Rudolph, M. Krötzsch, P. Patel-Schneider, and P. Hitzler, "OWL 2 Web Ontology Language Primer (Second Edition)," W3C, W3C Recommendation, Dec. 2012, http://www.w3.org/TR/owl2-primer.

[8] K.-Y. Lin and H.-P. Lu, "Why people use social networking sites: An empirical study integrating network externalities and motivation theory," *Computers in Human Behavior*, vol. 27, no. 3, pp. 1152–1161, 2011.

[9] B. Ramis Ferrer, S. Iarovyi, L. Gonzalez, A. Lobov, and J. L. Martinez Lastra, "Management of distributed knowledge encapsulated in embedded devices," *International Journal of Production Research*, pp. 1–18, 2015.

[10] G. Biamino, "A semantic model for socially aware objects," *Advances in Internet of Things*, vol. 2012, 2012.

[11] O. Voutyras, P. Bourelos, S. Gogouvitis, D. Kyriazis, and T. Varvarigou, "Social monitoring and social analysis in internet of things virtual networks," in *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*. IEEE, 2015, pp. 244–251.

[12] G. Capurso, M. Ruta, F. Scioscia, and E. Di Sciascio, "Object (B)logging: Semantic Self-Description for Cyber-Physical Systems," in *Proceedings of the Doctoral Consortium, Challenge, Industry Track, Tutorials and Posters @ RuleML+RR 2017*, ser. CEUR Workshop Proceedings, July 2017, vol. 1875.

[13] R. De Virgilio, E. Di Sciascio, M. Ruta, F. Scioscia, and R. Torlone, "Semantic-based RFID Data Management," in *Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks*. Springer, 2011, pp. 111–142.

[14] J. Bleecker, "A Manifesto for Networked Objects Cohabiting with Pigeons, Arphids and Aibos in the Internet of Things," in *Human-Computer Interaction with Mobile Devices and Services, International Conference on*, 2006.

[15] S. Benford, A. Hazzard, A. Chamberlain, and L. Xu, "Augmenting a guitar with its digital footprint," in *New Interfaces for Musical Expression, 15th International Conference on*, 2015.

[16] J. Miranda, N. Makitalo, J. Garcia-Alonso, J. Berrocal, T. Mikkonen, C. Canal, and J. M. Murillo, "From the Internet of Things to the Internet of People," *Internet Computing, IEEE*, vol. 19, no. 2, pp. 40–47, 2015.

[17] V. Pejovic and M. Musolesi, "Anticipatory mobile computing: A survey of the state of the art and research challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 47, 2015.

[18] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, "BASont – A modular, adaptive building automation system ontology," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2012, pp. 4827–4833.

[19] Y. Evchina, J. Puttonen, A. Dvoryanchikova, and J. L. M. Lastra, "Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems," *Engineering Applications of Artificial Intelligence*, vol. 43, pp. 111–126, 2015.

[20] D. Bonino, E. Castellina, and F. Corno, "The DOG gateway: enabling ontology-based intelligent domotic environments," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1656–1664, 2008.

[21] L. Mainetti, V. Mighali, L. Patrono, and P. Rametta, "A novel rule-based semantic architecture for iot building automation systems," in *Software, Telecommunications and Computer Networks (SoftCOM), 2015 23rd International Conference on*. IEEE, 2015, pp. 124–131.

[22] S. N. Han, G. M. Lee, and N. Crespi, "Semantic context-aware service composition for building automation system," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 752–761, 2014.

[23] M. Ruta, F. Scioscia, G. Loseto, and E. Di Sciascio, "A semantic-enabled social network of devices for building automation," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3379–3388, dec 2017.

[24] M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, S. Ieva, A. Pinto, and E. Di Sciascio, "Social Internet of Things for Domotics: a Knowledge-based Approach over LDP-CoAP," *Semantic Web*, 2018, in press.

[25] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *Internet Computing, IEEE*, vol. 16, no. 2, pp. 62–67, 2012.

[26] M. Horridge and P. F. Patel-Schneider, "OWL 2 Web Ontology Language Manchester Syntax (Second Edition)," W3C, W3C Working Group Note, Dec. 2012, https://www.w3.org/TR/owl2-manchester-syntax/.