Semantic-based Smart Homes: a Multi-Agent Approach

Giuseppe Loseto, Floriano Scioscia, Michele Ruta, Eugenio Di Sciascio
DEE - Politecnico di Bari
via Re David 200, I-70125, Bari, Italy
loseto@deemail.poliba.it, {f.scioscia, m.ruta, disciascio}@poliba.it

***Abstract*—*Ambient Intelligence* aims at autonomic coordination and control of appliances and subsystems located in a given environment. Home and Building Automation (HBA) complies with this paradigm but it is based on an explicit interaction with the user and a static set of operational scenarios. This paper proposes a more flexible multi-agent approach, leveraging semantic-based resource discovery and orchestration in HBA. Backward-compatible enhancements to EIB/KNX domotic standard allow to support the semantic characterization of user profiles and device functionalities, so enabling: (i) negotiation of the most suitable home services/functionalities according to implicit and explicit user needs, (ii) device-driven interaction for adapting the environment to context evolution. A power management problem in HBA is presented as a case study to better clarify the proposal and assess its effectiveness.**

***Index Terms*—Ambient Intelligence, Building Automation, EIB/KNX, Semantic Web, Multi-Agent Systems.**

I. INTRODUCTION

Ambient Intelligence (AmI) [1] refers to a research vision built upon advances in sensors networks, pervasive computing and artificial intelligence, that make a given environment capable of being sensitive and responsive by recognizing user needs and self-adapting accordingly. Devices communicate and interact autonomously, without direct human intervention, also making decisions based on multiple factors, including user presence and preferences. They are coordinated by intelligent systems acting as supervisors, devoted to manage available resources in order to meet assigned requirements.

Home and Building Automation (HBA) technologies should adhere to AmI fundamentals, but current systems are still far from that vision, being unable to grant such flexibility and autonomy. They now basically enable a static set of operational scenarios pre-defined during implementation and require explicit interaction with the user. On the contrary, an advanced and flexible management of information about users, devices and resources/services in a given ambient is needed. Really smart HBA infrastructures, autonomously controlling and adapting building appliances, could be conceived borrowing frameworks and approaches from artificial intelligence studies in pervasive computing field, also adapting theory and solutions of agent-based software design [2].

This paper proposes the exploitation of Knowledge Representation (KR) technologies and automated reasoning techniques, originally implementing the Semantic Web paradigm and properly adapted, to overcome restrictions of common domotic solutions. An enhancement to ISO/IEC 14543-3 standard, a.k.a. EIB/KNX (European Installation Bus/Konnex) [3], has been devised. Particularly, a context-aware multi-agent framework for building automation is proposed, which supports semantic annotation of both user profiles and device capabilities. The integration of a semantic micro-layer within KNX protocol stack enables novel decision support features in HBA, while preserving full backward compatibility.

Machine-understandable metadata characterize home environment, appliances and user profiles and preferences. Annotations are expressed in ontological formalisms derived from Description Logics (DLs) [4]: particularly DIG [5] has been adopted, being a more compact equivalent of OWL-DL[1]. As opposed to both static configuration approaches of standard HBA technologies and user-driven service selection of most research proposals, the framework we present here enables user-transparent and device-driven interaction. To this aim, the adopted multi-agent system allows requests coming from users and/or devices being collected by a *home mediator* which acts as a broker between users and home appliances. Each request is treated as a one-to-many negotiation among sender agent and various device agents. Such a complex process is divided in concurrent one-to-one negotiations between the home agent and each device agent. Services/resources so selected are used to cover sender requirements to the best possible extent.

The remainder of the paper is organized as follows. Section II outlines the proposed framework architecture and enhancements to KNX standard. A case study is reported in Section III, while relevant related work is discussed in Section IV. Final remarks are in Section V.

II. SEMANTIC-BASED HOME AUTOMATION

Multi-Agent Systems (MAS) are very helpful in HBA due to their ubiquitous and distributed nature. Hereafter the reference framework and the related infrastructure are reported.

A. Knowledge-based domotic and agent framework

As shown in Figure 1, the adopted MAS comprises a mediator as well as user and device agents referred to home appliances –including energy-providing systems (*e.g.*, photovoltaic collectors). The number of connected resources and agents may vary unpredictably –a new user, device or energy

[1]OWL Web Ontology Language, W3C Recommendation, February 10th 2004, available at http://www.w3.org/TR/owlfeatures/

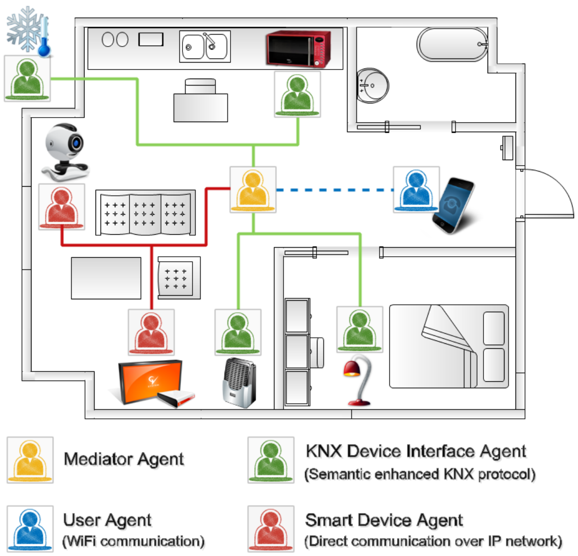

Fig. 1. Agent-based framework

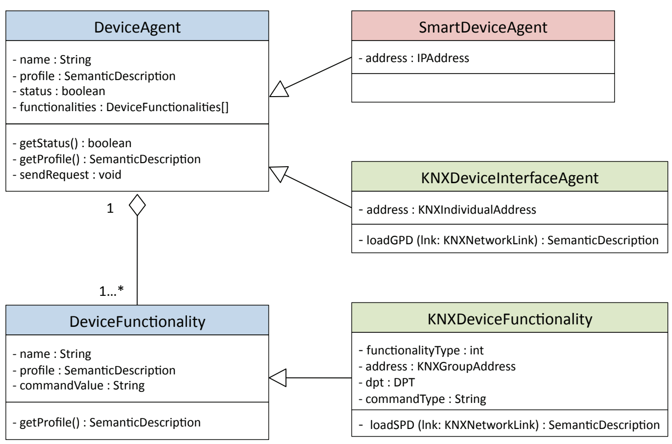

Fig. 2. Agent modeling

source can be connected or disconnected at any time– without redefining the communication and negotiation framework.

The user agent, running on a mobile client, addresses requests toward the home manager, describing the needs and preferences of the user. Each device agent can expose one or more services (*i.e.*, functional profiles). The agent-based architecture and EIB/KNX protocol enhancements allow also device agents to issue requests to the home system in order to supports automatic reconfiguration and adaptation to changing conditions. Figure 2 refers to the agent modeling scheme.

Smart Device Agents are thought to be embedded within advanced devices (*i.e.*, home appliances with some computational capabilities and memory availability). They encapsulate device status and properties in a semantic annotation to be provided during discovery operated by other agents or to be issued in semantic-based requests toward the home agent. Such requests are generated after a sensor data gathering phase or when the internal status changes. The goal is to negotiate a home configuration better fitting a possible new situation.

KNX Device Interface Agents support semantic-based enhancements in case of legacy or elementary appliances (*e.g.*, switches, lamps, and so on). In such cases, if semantic annotations are asked, the request will be replied by the agent. Conversely, if the home agent refers to standard KNX device properties, the request will be simply forwarded by the agent to the device.

The *Home (Mediator) Agent* has the responsibility of making the domotic environment *a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources* [6]. In particular: (i) it coordinates the explicit characterization of available services, described w.r.t. a reference ontology modeling conceptual knowledge for the building automation problem domain; (ii) when a request is received, it acts as a mediator in a negotiation round between the sender agent and each available device agent, in order to discover the (set of) elementary services that cover (part of) the request, maximizing the overall *utility*. It employs a logic-based bilateral negotiation protocol, originally conceived for marketplace scenarios [7] and fully revised to apply to HBA. There, agents are able to: (i) negotiate on available home services; (ii) reveal conflicting information between request and provided resources; (iii) support non-expert users in selecting home configurations ranked w.r.t. utility. In case of requests coming from the user, utility is the relevance of each wanted feature, while in request originating from devices, utility values are associated to service properties in order to minimize or maximize a given aspect (*e.g.*, costs, efficiency, comfort). In the case study reported afterwards, utility is exploited to minimize the consumption of external energy sources (electricity, gas) favoring the usage of homemade energy, *i.e.*, produced by equipment installed in the household, *e.g.*, photovoltaic systems.

Formally, a request (as well as each available home service/resource) is expressed as a set of formulas $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_n\}$ ($\mathcal{S}_i = \{\sigma_{i,1}, \sigma_{i,2}, \ldots, \sigma_{i,m}\}$ for the i-th service, respectively) in Description Logics. $\mathcal{ALN}$ (Attributive Language with unqualified Number restrictions) was adopted as reference language in the current system prototype and case study. Each formula represents a preference, to which a utility value is assigned by means of a function $u_\beta : \mathcal{B} \rightarrow Q^+$ such that $\sum_h u_\beta(\beta_h) = 1$ ($u_\sigma : \mathcal{S} \rightarrow Q^+$ s.t. $\sum_k u_\sigma(\sigma_k) = 1$, respectively), *i.e.*, utility values are normalized. Besides, each agent sets a disagreement threshold t, that is the minimum utility required to pursue a deal. The bilateral negotiation protocol is of *alternating offers with minimum concession* type: if some preferences in $\mathcal{B}$ and $\mathcal{S}_i$ are in contrast, provider and requester take turns in issuing counter-offers, each relaxing at every step the preference with the lowest utility value. The process is repeated until either an agreement is found (*i.e.*, remaining elements in $\mathcal{B}$ and $\mathcal{S}_i$ are not in contrast) or the negotiation fails because the residual utility of one of the agents has gone below its disagreement threshold. The *overall utility* of the agreement is then computed as the product of individual

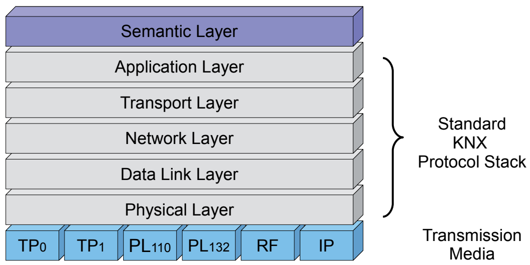

Fig. 3. Enhanced KNX protocol stack

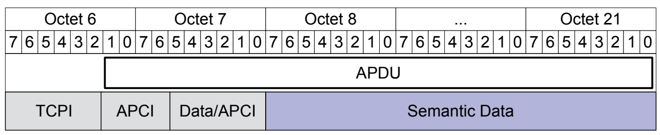

Fig. 4. Semantic-enabled Application layer Protocol Data Unit format

utilities. Due to space constraints, the reader is referred to [7] for a more comprehensive discussion of the negotiation protocol and its computational and game-theoretic properties. The above agent-based collaborative framework leverages the KNX knowledge-oriented evolution presented in [8], which implements a semantic micro-layer on the top of protocol stack as pictured in Figure 3. Novel services and functions have been introduced while keeping a full backward-compatibility with current protocol and devices. Semantic enhancements allow to fully describe device features by means of annotations expressed via logic languages such as RDF[2], OWL or DIG. The domotic knowledge domain has been conceptualized in a shared ontological vocabulary enabling a throughout characterization of home services and appliances. A preliminary study of KNX standard highlighted the inadequacy of the raw protocol to manage semantic metadata, requiring the definition of specific application layer services. Particularly, two service primitives have been introduced, allowing devices to autonomously exchange semantic annotations through the standard *Application layer Protocol Data Unit (APDU)*:
- *A_SEMANTIC_REQUEST*: used to send a semantic description of needed home functionalities;
- *A_SEMANTIC_RESPONSE*: contains descriptions of selected device functionalities covering the request.

Figure 4 shows how semantic annotations are carried on by the related KNX frame. In order to minimize sending data and communication time, semantic annotations are compressed by means of an algorithm specifically devoted to compact XML-based ontological languages [9]. Since descriptions can still exceed the maximum APDU data field size (14 bytes), the extended KNX frames have been used (up to 255 bytes, 249 of them reserved for data). However, if semantic annotations result even larger than APDU maximum limits, descriptions are split in more different APDUs including in the PDU the total packet number.

[2]RDF (Resource Description Framework) Primer, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/rdf-primer/

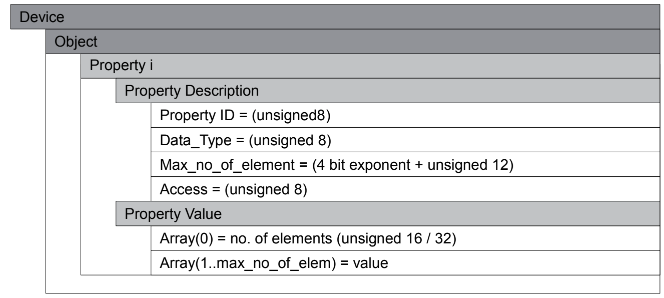

Fig. 5. Interface Object Structure

Two *Interface Objects* –data structures KNX used to set device properties– have been defined to manage structured and machine-understandable semantic descriptions. To maintain a full compatibility with original protocol and applications, new objects are compliant with structural specification in [10]. To describe generic device features, *i.e.*, manufacturer or model, a ***Generic Profile of Device (GPD)*** object has been introduced while ***Specific Profile of Device (SPD)*** objects store the semantic annotations of device functionalities. If a device provides different available services or operating modes, an SPD will be defined for each one. Both introduced interface objects adhere to the scheme reported in Figure 5.

According to this classification, GPD and SPD objects are featured by properties with following identifiers:
- `PID_OBJ_TYPE` = 1 ($0x01_h$): 16-bit mandatory field indicating the object type;
- `PID_OUUID` = 77 ($0x4D_h$): 16-bit Ontology Universally Unique Identifier (OUUID) marking the reference ontology the device semantic annotation refers to [11];
- `PID_OUUIDs` = 100 ($0x64_h$): OUUID set, useful when more ontologies are used to describe device functionalities. This field is present only in GPD properties;
- `PID_SEMANTIC_HEADER` = 150 ($0x96_h$): the header of compressed semantic annotation (variable-length string);
- `PID_SEMANTIC_BODY` = 151 ($0x97_h$): the body of compressed semantic annotation (variable-length string).

Finally, a new *DataPoint Type (DPT)* was defined to store the 16-bit ontology OUUID.

B. Reference architecture

The communication architecture infrastructuring the above framework integrates an EIB/KNX bus and an IP network used as fast backbone. Nowadays IP is increasingly adopted in automation systems and particularly in HBA. Such a hybrid home network interconnects several KNX/IP routers and enables the communication among different KNX lines via IP. In this way, devices send and receive KNX group telegrams through multicast IP frames compliant with the EIBnet/IP routing protocol.

As depicted in Figure 6, the overall framework architecture consists of four main functional components:
- *Central Unit*: which represents the system core and embeds

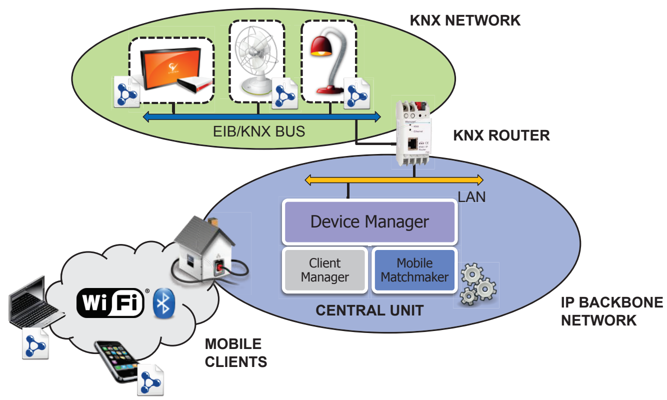

Fig. 6. Proposed framework reference architecture

a mobile client manager, a device manager and a micro matchmaker (based on the one presented in [12] and exploiting DL-based standard and non-standard inference services described in [13]). It runs on a laptop PC equipped with Intel Core 2 Duo T7700 CPU (2.4 GHz clock frequency), 4 GB DDR2 RAM and Ubuntu 10.04 operating system with Java Virtual Machine 1.6.0 17;
- *KNX Router*: which converts the EIB/KNX telegrams into IP frames and vice-versa according to EIBnet/IP standard. Besides, it filters telegrams to keep the bus load low;
- *Semantic-based devices*: *i.e.*, KNX devices implementing the protocol enhancements presented in the previous subsection;
- *Mobile clients*: *i.e.*, mobile devices, such as notebooks, smartphones or PDAs, able to send and receive semantic annotations properly encapsulated in KNX PDUs. Communication between clients and home system is based on IEEE 802.11 and Bluetooth protocols. A smartphone having a S5PC111 CPU (1 GHz clock frequency), 512MB RAM and Android 2.1 operating system has been used for the test.

Particularly, the agent running on the central unit allows to: (i) discover and orchestrate suitable home device functionalities compatible with users or context requirements via semantic-based inferences; (ii) rank in relevance order w.r.t. received requests the best services/resources to be activated; (iii) find possible inconsistencies between home current status and selected services or resources; (iv) inform about the matchmaking outcomes evidencing possible open issues and negotiation options. During start-up phase, the central unit also takes care of system configuration. It finds out all KNX routers connected to the home LAN through a discovery procedure defined in EIBnet/IP standard. For each router, a new bidirectional tunneling channel is established and the system is ready to accept further semantic requests.

Figure 7 shows a typical system interaction. Along with requests issued by a User Agent toward the Home Agent running in the central unit, the proposed system also allows a Device Agent to perform queries. In such case, devices exploit the previously described novel application layer service for conveying semantic requests in one or more KNX frames. Routers then forward them to the central unit over the IP

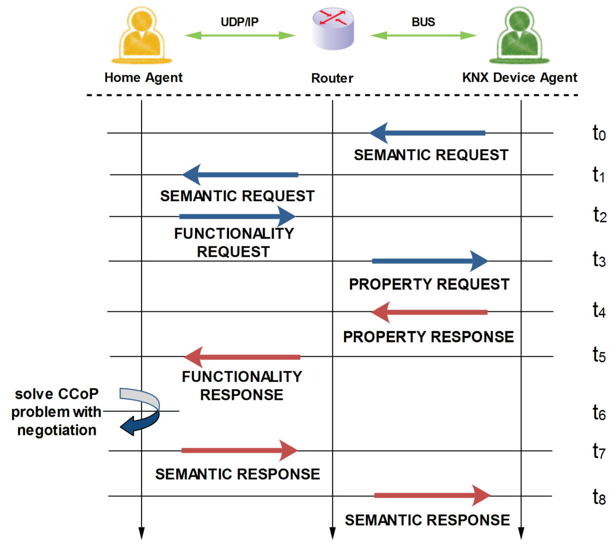

Fig. 7. System interaction

network. Instead, if the request comes from a mobile client, the central unit directly receives it via Bluetooth/Wi-Fi and processing starts at time t_2 of the diagram in Figure 7. In either case, the Home Agent aims to find a set of suitable home functionalities for performing a semantic-based *covering* process. Given a request and several available services –*i.e.*, home appliances– the covering allows to compose services in order to satisfy the request to the best possible extent.

The orchestration process can be formalized as in what follows:
1. A *FUNCTIONALITY_REQUEST* message is sent to KNX Router to discover available home appliances.
2. For each on-line device, the router sends a *PROPERTY_REQUEST* message to retrieve compressed semantic annotations of exposed services/resources.
3. Data received from devices are then forwarded to the central unit, decoded and temporarily stored in local memory.
4. Algorithm 1 is applied to request and service annotations. An early compatibility check is performed in order to find any active service/resource in conflict with the request. They will be deactivated subsequently. Then a *Concept Abduction Problem (CAP)* [13] is solved between request and compatible active functionalities to verify if the user request is already completely covered without activation of further services. Concept Abduction allows to determine what functionalities should be hypothesized, *i.e.*, what is missing, in order to completely satisfy the request. If there is such an uncovered part of the request, a *Concept Covering Problem (CCoP)* [14] is solved to select one or more deactivated functionalities whose orchestration fills needed features. Finally, the algorithm returns a set of services to turn on or off, along with the uncovered request, if present.
5. Selected functionalities are activated and a *A_SEMANTIC_RESPONSE* message is sent to the device agent originating the interaction. Instead, if the request came

Algorithm 1 Algorithm for request covering

Algorithm: $requestCovering\,(\langle R, A, NA, \mathcal{L}, \mathcal{T}\rangle)$

Require: $\mathcal{L}$ Description Logic, acyclic $\mathcal{T}$, request R, $a_i \in A, i = 1, 2, \ldots n$ and $na_j \in NA, j = 1, 2, \ldots m$ concept expressions of active/not active functionalities in $\mathcal{L}$ satisfiable in $\mathcal{T}$.

Ensure: $G = \{G_1, G_2, \ldots G_k\}$ set of functionalities to activate; $K = \{K_1, K_2, \ldots K_h\}$ set of functionalities to deactivate; H request uncovered part.

$G := \emptyset$
$K := \emptyset$
$H := R$
for all $a_i \in A$ **do**
  if $(a_i \sqcap R)$ is not satisfiable in $\mathcal{T}$ **then**
    $K := K \cup \{a_i\}$
    $A := A \setminus \{a_i\}$
  else
    $H :=$ *solveCAP* $(\langle \mathcal{L}, a_i, H, \mathcal{T}\rangle)$
  end if
end for
if $(H \neq \top)$ **then**
  $\langle G, H\rangle :=$ *solveCCoP* $(\langle \mathcal{L}, NA, H, \mathcal{T}\rangle)$
end if
return G, K, H

TABLE I
UTILITY VALUES FOR SCENARIO A

i	$\beta_{A,i}$	$u(\beta_{A,i})$
1	$isSuggestedForSensation.Cold$	0.6
2	$= 4\ available\_kWh$	0.2
3	$= 10\ outsideTemperature$	0.2
	$t_{\mathcal{B}_A}$	0.8

from a mobile device agent, a reply is sent back to the user.

III. CASE STUDY

A case study referred to power management in home automation was developed to make evident the capabilities of the proposed agent-based framework. The home agent exploits a bilateral negotiation process whose final aim is to obtain a logic-based ranking of available services and resources according to current status of user, devices and home environment, seeking to maximize both comfort and energy efficiency.

Let us consider the following example scenarios taken from our case study. *EIB/KNX-compatible equipment in Bob's house includes: an air-source heat pump, an electrical heater, photovoltaic collectors with battery accumulator and a weather station measuring outside temperature. Bob comes home from work and feels cold.* He inputs this information to the user agent on his smartphone, which issues a request to the home agent in order to provide heating. The home agent collects environmental information from device agents and associates it to the user profile, so that the request to be satisfied takes both user preferences and home status into account. *Weather station reports that outside temperature is* $10°C$*, while photovoltaic accumulator reports that* $4kWh$ *energy is available in scenario A and* $0kWh$ *in scenario B.* The proposed example can be formalized as follows with respect to an HBA ontology (expressly defined for the case study and not reported here due to lack of space).

Requests in scenarios A and B, named $\mathcal{B}_A$ and $\mathcal{B}_B$ respectively, are reported in Table I and Table II. They combine the user agent requirement for a service suggested in case of cold feeling and contextual information about temperature and energy availability, provided by the weather station and energy manager agents, respectively. Higher utility is assigned to the user preference, because user satisfaction is the primary goal of the related agent. Device agents make three service profiles available for activation: heat pump, electrical heater at half power, electrical heater at full power. They are named $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ and their descriptions are reported in Table III, Table IV and Table V, respectively. Utility values in Table III model the fact that the heat pump is more beneficial when no self-produced electric power is available and for higher external temperatures (due to thermodynamics, coefficient of performance is higher when working at a lower temperature differential). Values in Table IV model the fact that the electrical heater is more beneficial when self-produced electric power is available, while those in Table V model the fact that using the heater at full power requires more electricity, but is more efficient at lower temperatures.

Let us consider scenario A. As explained in Section II, the home agent (i) receives the functionality request from the user agent, (ii) collects available service descriptions from device agents, (iii) checks compatibility between active functionalities and the request, (iv) solves the Concept Covering Problem in order to find functionalities that are suitable to cover (part of) the request, mediating negotiation to select the ones with

TABLE II
UTILITY VALUES FOR SCENARIO B

i	$\beta_{B,i}$	$u(\beta_{B,i})$
1	$isSuggestedForSensation.Cold$	0.6
2	$= 0\ available\_kWh$	0.2
3	$= 10\ outsideTemperature$	0.2
	$t_{\mathcal{B}_B}$	0.8

TABLE III
UTILITY VALUES FOR HEAT PUMP

i	$\sigma_{1,i}$	$u(\sigma_{1,i})$
1	$isSuggestedForSensation.Cold$	0.5
2	$= 0\ available\_kWh$	0.1
3	$\geq 12\ outsideTemperature$	0.2
4	$\geq 8\ outsideTemperature$	0.2
	$t_{\mathcal{S}_1}$	0.6

TABLE IV
UTILITY VALUES FOR HEATER AT HALF POWER

i	$\sigma_{2,i}$	$u(\sigma_{2,i})$
1	$isSuggestedForSensation.Cold$	0.4
2	$\geq 3\ available\_kWh$	0.3
3	$\leq 8\ outsideTemperature$	0.3
	$t_{\mathcal{S}_2}$	0.6

TABLE V
UTILITY VALUES FOR HEATER AT FULL POWER

i	$\sigma_{3,i}$	$u(\sigma_{3,i})$
1	$isSuggestedForSensation.Cold$	0.6
2	$\geq 6\ available\_kWh$	0.2
3	$\leq 2\ outsideTemperature$	0.2
	$t_{\mathcal{S}_3}$	0.6

TABLE VI
CASE STUDY: NEGOTIATION RESULTS

Request	A	B
Heat pump	0.64	0.8
Heater half power	0.8	0.56
Heater full power	0.64	0.64

highest utility, and (v) activates selected functionalities. In our example, no service is active at the request time, so step (iii) has no effect. The first negotiation round occurs between user agent with request $\mathcal{B}_A$ and the first device agent, heat pump, with service $\mathcal{S}_1$. It can be noticed that constraints $\beta_{A,3}$ and $\sigma_{1,3}$ about temperature are in conflict, as well as constraints $\beta_{A,2}$ and $\sigma_{1,2}$ about available energy. Therefore negotiation is carried out as in what follows:
1. User agent discards $\beta_{A,2}$ ($u_\beta = 0.8, u_\sigma = 1$).
2. Device agent discards $\sigma_{1,3}$ ($u_\beta = 0.8, u_\sigma = 0.8$).
No more conflicts exist and utility of both agents is above their thresholds, so an agreement is reached with overall utility $u = u_\beta u_\sigma = 0.64$. Discarding environmental constraints can appear as inappropriate, since they model matters of fact, not modifiable preferences. This kind of situations can be taken into account by dividing every request and service profile in two sets of constraints, *strict* and a *negotiable* ones, with violation of any strict constraint immediately leading to a missed deal. In the current system prototype strict constraints are not implemented (although it is trivial to do so, with the framework already in place), but similar effects can be obtained by properly setting utility values and disagreement thresholds.

Negotiation is executed in the same way in all the other cases. Utility outcomes are summarized in Table VI. It can be noticed that, when solar power is available (scenario A), the heater is globally more beneficial than the heat pump, because no external resources are consumed, even though the heat pump is more efficient than the electrical heather from a thermodynamic standpoint. Conversely, when no self-produced power is available (scenario B), the heat pump is preferred. Nevertheless, it is useful to notice that for colder outside temperatures the utility of the heat pump would decrease and the heater might become the best option again.

The presented example is purposely simplified in order to make presentation of the proposed approach clear and short. In real scenarios, more articulated requests and service descriptions can be used. Benefits of the framework (enabling logic-based matchmaking and negotiation with support to approximate matches and service ranking) become even larger w.r.t. both standard home automation technologies, characterized by static profiles, and other state-of-the-art ontology-based agent infrastructures, which support only rule-based inferences and exact matches.

IV. RELATED WORK

Ambient Intelligence aims to increase comfort of living/working environments by efficiently exploiting available services and resources. Flexible and adaptive discovery and fruition of pervasive and embedded systems must be leveraged for that. Therefore, mobile context-aware middlewares [15] and agent systems [16], [17] are often seen as pivotal elements of AmI [18]. Particularly, MAS are suitable to model real-world social scenarios enabling concurrency and cooperation. According to Agent-Oriented Software Engineering (AOSE), agents can meaningfully represent and simulate entities (*e.g.*, devices), contexts or people emphasizing social capabilities (communication, cooperation, conflict resolution and negotiation). Several proposals can be found in literature for modeling HBA systems through MAS. Case studies presented in [19] evidence that mobile agents can be fruitfully adopted to build AmI-based systems. With specific reference to HBA, Morganti *et al.* in [20] defined a Home Automation system as composed by a collection of *domotic objects* and *domotic agents*. Each agent in the environment declares itself, detects –and possibly recognizes– other agents and interacts with them to solve electrical power allocation problems in common homes. The proposed solution also enabled the management of conflicts between competing agents. *DomoBuilder* [21] was based on a multi-agent architecture to integrate heterogeneous devices in the same environment. Agents were used to expose resource features toward the overall system. Furthermore, in latest years, due to the growing interest in reducing energy consumption, several MASs have been proposed specifically for energy management [22], [23], [24].

Unfortunately, the above agent-based solutions either require direct user intervention or support only elementary agent behaviors and basic interactions, lacking advanced service/resource characterization, discovery and composition. The exploitation of knowledge representation and reasoning techniques and technologies is thought as a means to reach higher levels of accuracy and controllability w.r.t. the above approaches, resulting in an improvement of user comfort and building efficiency. An agent system approach based on logic reasoning was proposed in [25]. A butler mediator recognizes the user context, based on interaction with sensor agents, in order to infer possible user's goals and select the most suitable workflow among a set of available candidates. It was supported by a communication protocol where agents automatically discover services available in the environment and dynamically compose them by exploiting View Design Language (VDL) rules. Wu *et al.* [26] defined a service-oriented smart home architecture where each component is designed as an agent communicating by exchanging messages via publish/subscribe events. In particular, when the smart home is going to perform a service for a user, it will compare service requirements with the environment situation to find out spaces whose status and resources are already available for activating a given service. Similarly, in [27] the use of intelligent agents, designed according to the BDI (Belief-Desire-Intention) model, was proposed to automate service composition tasks, so providing transparency from the user standpoint, although the approach lacks adequate expressiveness for user, device and service profiles description. Bonino *et al.* [28] developed a complete prototype for ontology-based HBA. The proposed architecture included a reasoner exploiting rule-based inferences, whose well-known limits make the system not completely suitable for a widespread usage in dynamic AmI contexts: in order to trigger a rule, the system state should fully match rule conditions. Nevertheless full matches are quite unlikely in real-life scenarios, where objects, subjects and events are featured by different heterogeneous descriptions, often partially in conflict among them. In [29] two ontologies for modeling agent-based applications in energy systems were compared demonstrating that the intrinsic properties of an energy system could successfully be expressed by means of semantic-based approaches.

V. Conclusion

The paper presented a distributed multi-agent framework for home and building automation, based on a semantic enhancement of EIB/KNX standard exploiting knowledge representation and reasoning technologies. The proposed approach allows advanced, fine-grained resource/service discovery grounded on the formal annotation of user characteristics and device capabilities and leveraging logic-based negotiation. The devised framework has been realized in a prototypical testbed in order to verify both feasibility and effectiveness.

Acknowledgments

The authors acknowledge partial support of national project ERMES (Enhance Risk Management through Extended Sensors) - PON (2011-2014).

References

[1] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277 – 298, 2009.

[2] H. Nakashima, H. Aghajan, and J. C. Augusto, Eds., *Handbook of Ambient Intelligence and Smart Environments*. New York: Springer, 2010.

[3] Konnex Association, *KNX Handbook for Home and Building Control*, 5th ed., 2006.

[4] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook*. Cambridge University Press, 2002.

[5] S. Bechhofer, R. Möller, and P. Crowther, "The DIG Description Logic Interface," in *Proceedings of the 16th Int. Workshop on Description Logics (DL'03)*, ser. CEUR Workshop Proceedings, vol. 81, Rome, Italy, September 2003.

[6] D. Weyns, A. Omicini, and J. Odell, "Environment as a first class abstraction in multiagent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 5–30, Feb. 2007.

[7] A. Ragone, M. Ruta, E. Di Sciascio, and F. Donini, "Bargaining Agents in Wireless Contexts: An Alternating-Offers Protocol for Multi-issue Bilateral Negotiation in Mobile Marketplaces," *E-Commerce and Web Technologies*, pp. 14–25, 2009.

[8] M. Ruta, F. Scioscia, E. Di Sciascio, and G. Loseto, "Semantic-based Enhancement of ISO/IEC 14543-3 EIB/KNX Standard for Building Automation," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 731–739, 2011.

[9] M. Ruta, T. Di Noia, E. Di Sciascio, and F. Scioscia, "An efficient data compression algorithm for semantic-based ubiquitous computing applications," in *Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM07)*. IARIA, 2007, pp. 177–182.

[10] Konnex Association, *Handbook for Home and Building Control: Application Interface Layer*, 1st ed., 2001, vol. 3, ch. 4.1, pp. 12–16.

[11] M. Ruta, T. Di Noia, E. Di Sciascio, and F. Donini, "Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications," *Int. Journal of Web and Grid Services*, vol. 2, no. 4, pp. 424–452, 2006.

[12] M. Ruta, F. Scioscia, T. Di Noia, and E. Di Sciascio, "Reasoning in Pervasive Environments: an Implementation of Concept Abduction with Mobile OODBMS," in *2009 IEEE/WIC/ACM Int. Conf. on Web Intelligence*. IEEE, 2009, pp. 145–148.

[13] S. Colucci, T. Di Noia, A. Pinto, A. Ragone, M. Ruta, and E. Tinelli, "A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces," *Int. Journal of Electronic Commerce*, vol. 12, no. 2, pp. 127–154, 2007.

[14] A. Ragone, T. Di Noia, E. Di Sciascio, F. M. Donini, S. Colucci, and F. Colasuonno, "Fully automated web services discovery and composition through concept covering and concept abduction," *Int. Journal of Web Services Research (JWSR)*, vol. 4, no. 3, pp. 85–112, 2007.

[15] M. Mamei and F. Zambonelli, "Programming pervasive and mobile computing applications: The TOTA approach," *ACM Trans. Softw. Eng. Methodol.*, vol. 18, no. 4, pp. 15:1–15:56, Jul. 2009.

[16] M. P. Locatelli and G. Vizzari, "Awareness in collaborative ubiquitous environments: The multilayered multi-agent situated system approach," *ACM Trans. Auton. Adapt. Syst.*, vol. 2, no. 4, Nov. 2007.

[17] K. I. K. Wang, W. H. Abdulla, and Z. Salcic, "Ambient intelligence platform using multi-agent system and mobile ubiquitous hardware," *Pervasive Mob. Comput.*, vol. 5, pp. 558–573, October 2009.

[18] C. Ramos, J. C. Augusto, and D. Shapiro, "Ambient Intelligence - the Next Step for Artificial Intelligence," *IEEE Intelligent Systems*, vol. 23, pp. 15–18, March 2008.

[19] D. Tapia, A. Abraham, J. Corchado, and R. Alonso, "Agents and ambient intelligence: case studies," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, pp. 85–93, 2010.

[20] G. Morganti, A. M. Perdon, G. Conte, and D. Scaradozzi, "Multi-agent system theory for modelling a home automation system," in *Proceedings of the 10th Int. Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence*, ser. IWANN '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 585–593.

[21] A. Addis and G. Armano, "DomoBuilder: A MultiAgent Architecture for Home Automation," in *Proceedings of the 11th Workshop on Objects and Agents (WOA 2010)*, ser. CEUR Workshop Proceedings, vol. 621, September 2010.

[22] S. Abras, S. Pesty, S. Ploix, and M. Jacomino, "Advantages of MAS for the Resolution of a Power Management Problem in Smart Homes," in *Advances in Practical Applications of Agents and Multiagent Systems*, ser. Advances in Intelligent and Soft Computing, Y. Demazeau, F. Dignum, J. Corchado, and J. Prez, Eds. Springer Berlin / Heidelberg, 2010, vol. 70, pp. 269–278.

[23] L. Klein, J. Kwak, G. Kavulya, F. Jazizadeh, B. Becerik-Gerber, P. Varakantham, and M. Tambe, "Coordinating occupant behavior for building energy and comfort management using multi-agent systems," *Automation in Construction*, vol. 22, pp. 525 – 536, 2012.

[24] J. Kwak, P. Varakantham, R. Maheswaran, M. Tambe, F. Jazizadeh, G. Kavulya, L. Klein, B. Becerik-Gerber, T. Hayes, and W. Wood, "SAVES: A Sustainable Multiagent Application to Conserve Building Energy Considering Occupants," in *Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

[25] D. Cavone, B. De Carolis, S. Ferilli, and N. Novielli, "An agent-based approach for adapting the behavior of a smart home environment," in *Proceedings of the 12th Workshop on Objects and Agents (WOA2011)*, ser. CEUR Workshop Proceedings, vol. 741, 2011, pp. 105–111.

[26] C.-L. Wu and L.-C. Fu, "Design and Realization of a Framework for Human-System Interaction in Smart Homes," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 42, no. 1, pp. 15 –31, jan. 2012.

[27] M. Santofimia, F. Moya, F. Villanueva, D. Villa, and J. Lopez, "Intelligent Agents for Automatic Service Composition in Ambient Intelligence," in *Web Intelligence and Intelligent Agents*, Z. Usmani, Ed. InTech, 2010, pp. 411–428.

[28] 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, november 2008.

[29] J. Keirstead and K. H. van Dam, "A comparison of two ontologies for agent-based modelling of energy systems," in *Proceedings of the First Int. Workshop on Agent Technologies for Energy Systems (ATES 2010)*, 11 May 2010, pp. 21–28.