

An Agent Framework for Knowledge-based Homes

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

ABSTRACT

Current technologies for Home and Building Automation (HBA) require an explicit interaction with the user and allow a static set of operational scenarios, defined during system implementation. More flexible HBA solutions are needed to enable the so-called *Ambient Intelligence*, permitting an automatic coordination and control of appliances and sub-systems located in a given environment. This paper proposes an agent-based approach leveraging semantic-based resource discovery and orchestration in HBA. Backward-compatible enhancements to one of the most widespread domotic standards, *i.e.*, EIB/KNX ISO/IEC 14543-3, have been devised, in order to support novel functionalities. The semantic annotation of user profiles and device capabilities is used to: (i) determine the most suitable home services/functionalities according to implicit and explicit user needs, (ii) allow device-driven interaction for autonomous adaptation of the environment to context modification. A case study about power management in HBA is presented to better clarify the proposal and assess its effectiveness.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence — *Intelligent agents, Multiagent systems*

General Terms

Algorithms; Theory

Keywords

Ambient intelligence; Building automation; EIB/KNX; Semantic Web; Multi-Agent Systems

1. INTRODUCTION

In latest years, advances in information and communication technology have opened the possibility to create smart buildings, aimed at increasing comfort and security, making management easier, reducing energy consumption and minimizing environmental impact. *Ambient Intelligence* (AmI) [28] refers to a research vision where people are surrounded by intelligent and unobtrusive micro-components, capable of being sensitive and responsive by recognizing user needs and self-adapting their behavior accordingly. In 2001 the Information Society Technologies Advisory Group (ISTAG) body of the European Commission envisioned the implementation of concrete AmI environments in real-life scenarios within 10 years approximately [9]. Devices had to communicate and

interact autonomously, without the need for direct user intervention, also making decisions based on multiple factors, including user presence and preferences. They had to be coordinated by intelligent systems acting as supervisors devoted to manage available resources in order to meet users' requirements. Human-Computer Interaction (HCI) theory evidences that humans have poor and basic knowledge of settings on devices which exploit [10], therefore AmI should enable a gradual shift from explicit to mainly implicit HCI paradigms.

Current systems and standard technologies developed for Home and Building Automation (HBA) are still far from that vision, being unsuitable for granting such an autonomy and flexibility. They still require explicit interaction with the user and are basically tied to a static set of operational scenarios defined during implementation. In order to setup novel and intelligent HBA infrastructures, able to autonomously adapt and control building appliances, smart environments have to be conceived according to results coming from pervasive computing studies, artificial intelligence theory and agent-based software design [18]. A flexible and powerful management of information about users, devices and resources/services in a given context is needed. Consequently, AmI research is closely related to studies for effective discovery and coordination in volatile and resource-constrained scenarios.

This paper proposes to overcome restrictions of common domotic appliances through the exploitation of Knowledge Representation (KR) technologies and automated reasoning techniques, originally conceived for the Semantic Web. An enhancement to ISO/IEC 14543-3 standard, a.k.a. EIB/KNX (European Installation Bus / Konnex) [16], has been devised in a knowledge-based and context-aware computing framework for building automation, supporting semantic annotation of both user profiles and device capabilities. The integration of a semantic micro-layer within KNX protocol stack enables novel resource discovery and decision support features in HBA, while preserving full backward compatibility. Machine-understandable metadata characterize both home environment and user profiles and preferences. Annotations are expressed in ontological formalisms derived from Description Logics (DLs) [3]: particularly DIG [4] has been adopted, being a more compact equivalent of OWL-DL¹. A distinguishing feature of the proposed framework is to enable a user-transparent and device-driven interaction as

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

opposed to current static configuration approaches. To the purpose, an overall multi-agent framework has been pointed out, where requests coming from users and/or devices are collected by a *home daemon* acting as mediator 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 several concurrent one-to-one negotiations between the home agent and each device agent. Services/resources so selected will be used in the subsequent covering phase.

The remainder of the paper is organized as follows. In Section 2 relevant related work is surveyed, while Section 3 outlines the proposed framework architecture and enhancements to KNX standard. A case study is reported in Section 4, and final remarks and future work are in Section 5.

2. RELATED WORK

The final goal of Ambient Intelligence is a greater friendliness of most common appliances surrounding a user in an environment, as well as a more efficient support to available services and resources, through more flexible and adaptive discovery and fruition techniques applied to pervasive and embedded systems. Among other research efforts, promising studies refer to Agent Systems, often seen as a characteristic element of AmI [21]. Particularly, Multi-Agent Systems (MAS) are suitable to model real-world social scenarios enabling concurrency and cooperation. According to the Agent-Oriented Software Engineering (AOSE) paradigm, agents can meaningfully represent and simulate entities (*e.g.*, devices), contexts or people emphasizing social capabilities (communication, cooperation, conflict resolution and negotiation), hence they can be usefully exploited in AmI. Several proposals can be found in literature for modeling HBA systems based on MAS. Tapia *et al.* [29] presented case studies referred to intelligent environments, demonstrating that mobile agents can be fruitfully adopted to build AmI-based systems. With specific reference to HBA, Morganti *et al.* [17] defined a Home Automation system as composed by a collection of *domotic objects* and *domotic agents* and a context as the environment where agents interact along with the interaction modes. Each agent in the environment declares itself, detects –and possibly recognizes– other agents and interacts with them according to given system rules. The agent behavior was modeled as a time-driven sequence of state transitions during operation modes. Agents were exploited to solve electrical power allocation problems in common homes, having a predefined power consumption threshold. The proposed solution also enabled the management of conflicts between competing agents. In [30] an AmI platform was proposed to allow easy adaptation and integration of multi-modal user interfaces to heterogeneous devices networks. The MAS was implemented using JADE² multi-agent development framework, whereas IP-based UPnP (Universal Plug and Play) was exploited as discovery protocol. An IP interface agent, a fuzzy inference agent, a decision tree agent and UPnP control point agent were implemented in a working prototype; furthermore, an XML-based language was developed as standard communication mean between user interfaces and the MAS. *DomoBuilder* [2] proposed a multi-agent architecture to integrate heterogeneous devices in the

same environment. Agents were used to expose resource features toward the overall system. Finally, [1] defined a MAS to control appliances and energy sources in buildings to determine the best energy assignment plan. Unlike the above-mentioned works, each agent identifies a single service performing a specific task.

In the last years, due to the growing interest in reducing energy consumption, several MAS have been proposed for building energy and comfort management. Rogers *et al.* [22] developed a home energy management agent working on behalf of the householder, and able to learn the thermal characteristics of the home predicting the additional factors that affect the cost and carbon emissions of heating use. The proposed approach was evaluated by taking into account environmental and consumption parameters and it shows an overall reduction of energy consumption cost and carbon emissions. In [12] a Multi-Agent Comfort and Energy System (MACES) was implemented to model a modern management and control of both building systems and occupants. MACES coordinates appliances and users through direct changes to occupant meeting schedules. Kwak *et al.* [32] proposed an innovative MAS, named *SAVES*, aiming at saving energy in commercial buildings. Agents communicate and negotiate with human occupants to save energy and optimal policies are generated considering multiple criteria (*e.g.*, energy and personal comfort), as well as uncertainty in occupant preferences.

Unfortunately, agent-based solutions either require direct user intervention or support only basic interactions between autonomous device agents, lacking advanced service/resource discovery and composition capabilities. On the contrary, an agent system approach based on logic reasoning was proposed in [5]. It was supported by a communication protocol where agents automatically discover services available in the environment and dynamically build composite services by exploiting View Design Language (VDL) rules. Wu *et al.* [31] defined a multi-agent service-oriented smart home architecture where each component is designed as an agent. Agents communicate by exchanging messages via publish/subscribe events through shared topics. The home system functionalities include interaction with inhabitants via smart interfaces, manipulation of devices around users and change of the environmental status. 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.

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 effectiveness. In [11] two ontologies for modeling agent-based applications in energy systems were presented and compared. The paper demonstrated that the intrinsic properties of an energy system could successfully be expressed by means of ontology-based approaches. Further recent researches about knowledge-based solutions for energy management were proposed in [7, 13]. Bonino *et*

²Java Agent Development Framework (JADE), <http://jade.tilab.com/>

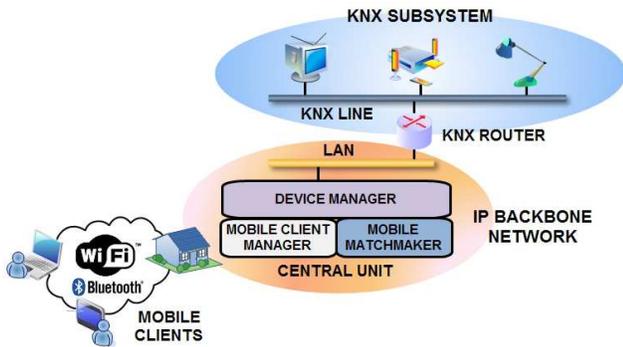


Figure 1: Proposed framework architecture

al. [6] acknowledged the relevance of knowledge-based approaches and proposed a self-contained prototype for intelligent domotic environments based on technologies derived from the Semantic Web. The proposed architecture included a reasoning module 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. A possible solution can be found in context-aware computing frameworks enabling decision support features (allowed by non-standard logic-based inferences) possibly including non-exact matches calculation. In [26] an early building automation framework was introduced, combining ontology-based annotation of both user profiles and device capabilities with a semantic-based enhancement of EIB/KNX standard for resource discovery. In the present paper, an agent-based evolution of that framework is presented, including a formal protocol for automated negotiation between user and device agents.

3. SEMANTIC-BASED HOME AUTOMATION

3.1 Reference architecture

IP is the most important standard protocol for local and wide area computer networks. Nowadays it is increasingly used in automation systems and particularly HBA. The communication architecture underlying the proposed approach integrates an EIB/KNX bus with an IP network used as fast backbone. This hybrid home network interconnects several KNX/IP routers and enables the communication among different KNX lines via the IP backbone. In this way, devices send and receive KNX group telegrams via multicast IP frames compliant with the EIBnet/IP routing protocol.

The overall framework architecture is shown in Figure 1 and it consists of four main functional components:

- *Central Unit*: represents the system core and embeds a device manager and a mobile client manager. The central unit also includes a micro matchmaker, based on the one presented in [25] and exploiting DL-based standard and non-standard inference services described in [8];
- *KNX Router*: 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;

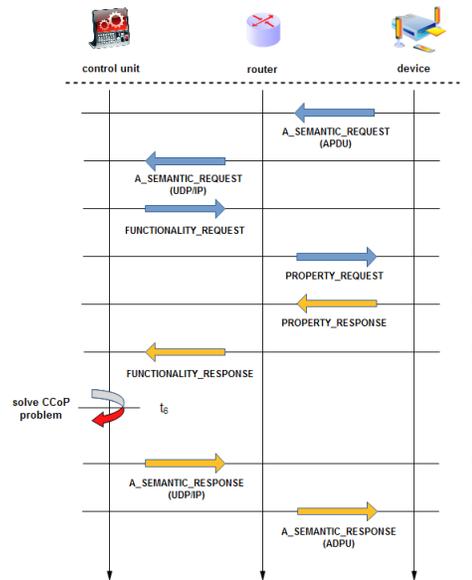


Figure 2: System interaction

- *Semantic-based devices*: KNX devices implementing the enhancements to the standard presented in the following subsection;

- *Mobile clients*: mobile devices, such as notebooks, smartphones or PDAs, equipped with a tool able to send and receive semantic annotations properly encapsulated in KNX PDUs. Clients can also simply activate elementary functionalities for environmental control and remote devices check-up. Communication between clients and home system is based on IEEE 802.11 and Bluetooth protocols.

Particularly, the agent running on the central unit allows to: (i) discover and orchestrate suitable *in-home* device functionalities compatible with users or system requirements via semantic-based inferences; (ii) arrange in relevance order the best services/resources to be activated in order to fulfill received requests; (iii) find possible inconsistencies between home current status and selected services or resources; (iv) explain the matchmaking outcomes evidencing possible open issues and negotiation options. During start-up phase, the central unit also takes care of system configuration. By sending specific messages, 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.

Typical system interaction is shown in Figure 2. Along with requests issued by the user agent toward the home agent running in the central unit, the proposed system also allows a device agent to perform queries requiring to activate/deactivate home appliances. In such case, devices exploit a novel application layer service (described in detail in subsection 3.2) which encapsulates a semantic request in one or more KNX frames. Routers then forward these packets to the central unit over the IP 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 in Figure 2. In both cases, the central unit aims to find a set of suitable home functionalities for performing a semantic-

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 , $a_i \in A$, $i = 1, 2, \dots, n$ and $na_j \in NA$, $j = 1, 2, \dots, m$ concept expressions of active/not active functionalities in \mathcal{L} satisfiable in \mathcal{T} .**Ensure:** $G = \{G_1, G_2, \dots, G_k\}$ set of functionalities to activate; $K = \{K_1, K_2, \dots, K_h\}$ set of functionalities to deactivate; H request uncovered part.

```
1:  $G := \emptyset$ 
2:  $K := \emptyset$ 
3: for all  $a_i \in A$  do
4:   if  $(a_i \sqcap R)$  is not satisfiable in  $\mathcal{T}$  then
5:      $K := K \cup \{a_i\}$ 
6:      $A := A \setminus a_i$ 
7:   end if
8: end for
9:  $H := \text{solveCAP}(\langle \mathcal{L}, A, R, \mathcal{T} \rangle)$ 
10: if  $(H \neq \emptyset)$  then
11:    $\langle G, H \rangle := \text{solveCCoP}(\langle \mathcal{L}, NA, H, \mathcal{T} \rangle)$ 
12: end if
13: return  $G, K, H$ 
```

based *covering* process. Given a request and several available services –*i.e.*, home appliances– the covering procedure allows to compose services with an increasing level of complexity in order to satisfy the request to the best possible extent.

Such orchestration process can be formalized as follows:

1. A *FUNCTIONALITY_REQUEST* message is sent to KNX Router to discover home appliances;
2. For each device connected to the KNX line, the router sends a *PROPERTY_REQUEST* message to obtain semantic annotations of exposed services. They are encoded through a specific compression algorithm (presented in [24]) in order to reduce latencies and network overload;
3. Data received from devices are then forwarded to the central unit, decoded and temporarily stored in local memory;
4. Algorithm 1 solving the *Concept Covering Problem (CCoP)* [19] is then applied to both request and annotations of available home appliances. In the first step, a compatibility check is performed in order to find active services and resources contrasting the request. They will be deactivated subsequently. Then a Concept Abduction Problem (CAP) [8] is solved between request and compatible active functionalities to verify if the user request is already completely covered without activation of further services. Just notice that Concept Abduction allows to determine what should be hypothesized, *i.e.*, what is missing, in functionalities in order to completely satisfy the request. If there is such an uncovered part of the request, a CCoP is solved to select one or more deactivated functionalities whose orchestration fills requested features. Finally, the algorithm returns a set of services to enable or to turn off along with the uncovered part of the original query, if present;
5. Selected functionalities are activated and a *A_SEMANTIC_RESPONSE* message is sent to the home agent originating the interaction. Instead, if the request came from a mobile device agent, a reply is sent back to the user.

3.2 Agent framework

Figure 3 shows the proposed agent framework. Agent systems result very helpful in HBA due to the ubiquitous and distributed nature of such applications. Agents are embedded in both energy-providing systems (*e.g.*, photovoltaic appliances) and home equipments, and allow to implement

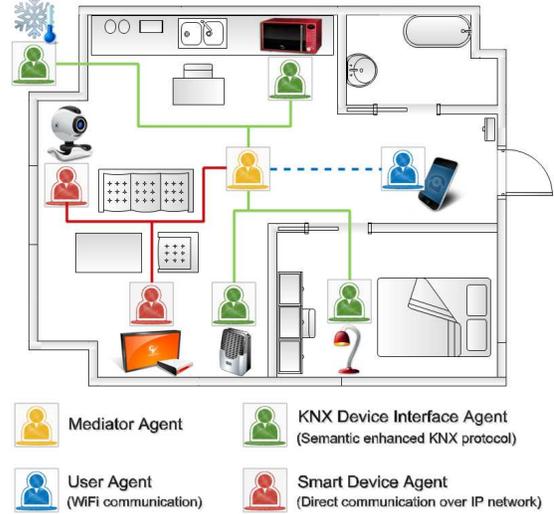


Figure 3: Agent-based framework

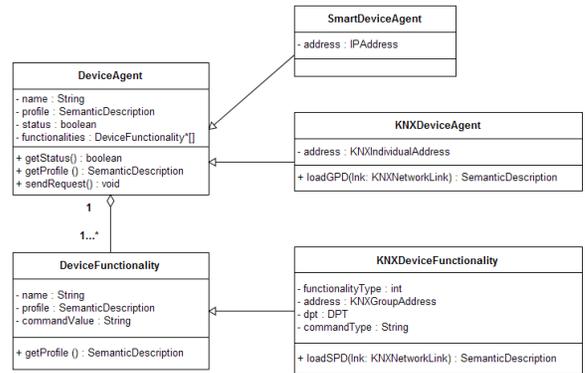


Figure 4: Agent modeling

independent and flexible control systems. The number of connected resources and agents may vary –a new user, device or energy source can be connected or disconnected at any time– without redefining the communication and negotiation framework.

In particular the user agent, running on a mobile client, addresses a request toward the home environment, describing needs and preferences of the user. Each device agent can provide 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, so that automatic home environment reconfiguration and adaptation to changing conditions is supported. The adopted agent modeling scheme is reported in Figure 4.

Smart Device Agents encapsulate their status and properties in a semantic annotation and send semantic-based requests –*e.g.*, generated after a sensor data gathering phase or when their internal status changes– to the home agent for negotiating an environmental profile better fitting the new situation. *KNX Device Interface Agents* are thought to support semantic-based enhancements in case of legacy or elementary appliances (such as switches, lamps, and so on). In such cases, if the home agent requires standard KNX de-

vice properties, the request will be simply forwarded by the agent to the given device. Conversely, in case semantic annotations are needed, the request will be directly replied by the agent.

Exploiting the logic-based bilateral negotiation protocol defined in [20], the home agent acts as a *mediator* in a negotiation round between the user agent and each available device agent, in order to discover the (set of) elementary services that maximize the overall utility and cover (part of) the request. The employed negotiation protocol –originally devised for marketplace scenarios– has been fully revised to apply to buildings energy systems. There agents are able to: (i) negotiate on available home and energy resources through a user-transparent and device-driven interaction; (ii) reveal conflicting information between request and resources on energy constraints; (iii) support non-expert users in selecting home configurations ranked w.r.t. utility.

From the user standpoint, utility is a measure of the relevance of each requested feature. From the device standpoint, utility values associated to service properties can be used to minimize operating costs or maximize efficiency. In the case study reported in the following section, utility is exploited to minimize the consumption of external energy sources (electricity, gas) favoring the usage of energy produced by equipment installed in the household, *e.g.*, photovoltaic systems. Current building automation employ generic and rigid profiles, which cannot adapt resource usage patterns according to dynamic conditions.

Formally, the request (as well as each available service description) is expressed as a set of formulas $\mathcal{B} = \{\beta_1, \beta_2, \dots, \beta_n\}$ ($\mathcal{S}_i = \{\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,m}\}$ for the *i*-th service, respectively) in Description Logics, w.r.t. a suitable common reference ontology, modeling conceptual knowledge for the building automation problem domain. \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. Furthermore, 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} are in contrast, supplier 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} are not in contrast) or the negotiation fails because the residual utility of one of the agents has gone below its disagreement threshold. Finally, the *overall utility* of the agreement is computed as the product of individual utilities. Due to space constraints, the reader is referred to [20] for a more comprehensive discussion of the negotiation protocol and its computational and game-theoretic properties.

To support this agent-based collaborative framework, the proposed approach leverages a knowledge-based evolution of KNX and particularly it implements a semantic micro-layer on the top of KNX protocol stack as pictured in Figure 5. Novel services and functions have been introduced while keeping a full backward-compatibility with current protocol and devices. Thanks to such semantic enhancement to the standard, device features can be fully described

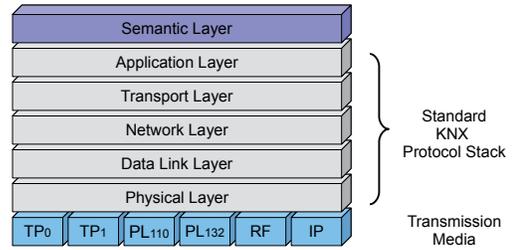


Figure 5: Enhanced KNX protocol stack

by means of annotations expressed via logic languages such as RDF³, OWL or DIG. Exploiting knowledge representation technologies, the domotic domain has been conceptualized in a shared ontological vocabulary so permitting a complete characterization of home services and appliances. A preliminary study of KNX standard highlighted the inadequacy of the raw protocol to manage semantic metadata. Hence, two new Interface Objects have been defined in order to manage structured and machine-understandable semantic descriptions. Notice that Interface Objects are data structures used to specify device properties. To maintain a full compatibility with KNX protocol and applications, new objects are compliant with structural specification in [14]. Each interface object consists of:

- **Object Name**;
- **Object Type**: a 16-bit unique identifier. KNX standard defines different value ranges for this code according to specific object usage;
- **Object Properties**: each property is composed by a property description and a property value. Property description consists of a property ID, a data type, a value indicating the maximum number of elements (**max_no_of_elem**) and a field concerning property access rights. The property value is instead an array with **max_no_of_elem**+1 elements. The array element “0” contains the current number of valid elements.

To describe generic device features, *i.e.*, manufacturer or model, a **Generic Profile of Device (GPD)** object has been introduced. For complying the above constraints, the unused code 1200 has been assigned to this object. The 1205 value is used instead to identify a **Specific Profile of Device (SPD)** object, storing the semantic annotations of device functionalities. If a device provides different available services, an SPD will be defined for each service. Both introduced interface objects present similar properties, and reflect the scheme in Figure 6.

In order to define novel properties, also specific **Property Identifiers (PID)** –in addition to object type ID– have been introduced. For application interface objects, KNX specification provides three ID categories:

- *Standardized object independent ID* [0-50];
- *Standardized object dependent ID* [51-154];
- *Not standardized ID* [155-255].

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 indi-

³RDF (Resource Description Framework) Primer, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/rdf-primer/>

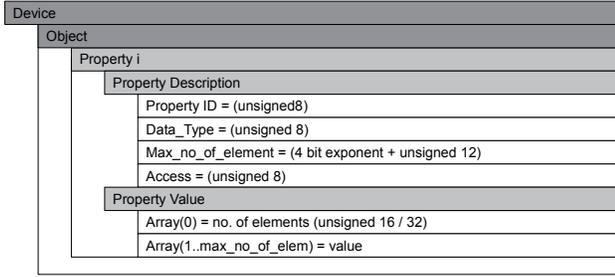


Figure 6: Interface Object Structure

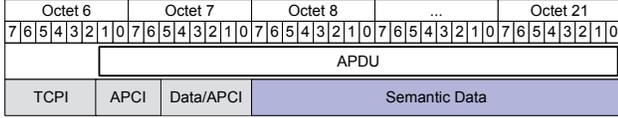


Figure 7: Semantic-enabled Application layer Protocol Data Unit format

cating the object type;

- PID_OUUUID = 77 (0x4D_h): 16-bit Ontology Universally Unique Identifier (OUUID) marking the reference ontology the device semantic annotation refers to [23];
- PID_OUUUIDs = 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. It is stored in the object as a variable length string;
- PID_SEMANTIC_BODY = 151 (0x97_h): the body of compressed semantic annotation. Also this property is defined as a string.

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

Introduced modifications to the KNX protocol required the definition of specific application layer services to have complete and user-independent device interactions. Particularly two new service primitives have been implemented allowing devices to autonomously exchange semantic meta-data through the standard *Application layer Protocol Data Unit (APDU)*. The resulting KNX frame is shown in Figure 7. Each primitive is identified by a specific code included in *Application layer Protocol Control Information (APCI)* field. According to code assignment in [15], the first two free APCI codes reserved for user messages are used to define:

- *A_SEMANTIC_REQUEST* service: used to send a semantic description of needed home functionalities;
- *A_SEMANTIC_RESPONSE* service: containing description of selected device functionalities covering the semantic-based request.

As said, to minimize sending data and communication time, semantic annotations are encoded in a compact way by means of an algorithm aimed at efficient compression of XML-based ontological languages [24]. Nevertheless descriptions can exceed maximum APDU data field size (14 bytes). Hence the extended KNX frames have been used (up to 255 bytes), 249 of them reserved for data. However in case semantic annotations result larger than APDU max-

Table 1: Utility values for scenario A

i	$\beta_{A,i}$	$u(\beta_{A,i})$
1	<i>isSuggestedForSensation.Cold</i>	0.6
2	= 20 <i>available_kWh</i>	0.2
3	= 10 <i>outsideTemperature</i>	0.2
	t_{B_A}	0.8

Table 2: Utility values for scenario B

i	$\beta_{B,i}$	$u(\beta_{B,i})$
1	<i>isSuggestedForSensation.Cold</i>	0.6
2	= 0 <i>available_kWh</i>	0.2
3	= 10 <i>outsideTemperature</i>	0.2
	t_{B_B}	0.8

imum limits, descriptions are split in more different APDUs. The total number of packets is set into *number of elements* variable of the related interface object.

4. CASE STUDY

In this section, capabilities of the proposed agent-based framework are clarified by means of a case study referred to power management in home automation. As the subsequent example will show, by means of semantic-based negotiation our framework allows to perform logic ranking of services according to current status of user, devices and home environment, seeking to maximize user satisfaction as well as 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 modules 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 modules accumulator reports that 20kWh are available in scenario A and 0kWh in scenario B.* The proposed example can be formalized as follows with respect to the HBA ontology defined for the case study (not reported due to lack of space).

Requests in scenarios A and B, named \mathcal{B}_A and \mathcal{B}_B respectively, are reported in Table 1 and Table 2. They combine the user agent requirement for a service suggested in case of cold feeling and environmental information about temperature and energy availability, provided by the weather station and accumulator agents, respectively. Higher relevance 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 3, Table 4 and Table 5, respectively. Utility values in Table 3 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 of a heat pump is higher when working at a lower temperature differential). Utility val-

Table 3: Utility values for Heat Pump

i	$\sigma_{1,i}$	$u(\sigma_{1,i})$
1	<i>isSuggestedForSensation.Cold</i>	0.5
2	$= 0$ <i>available_kWh</i>	0.1
3	≥ 12 <i>outsideTemperature</i>	0.2
4	≥ 8 <i>outsideTemperature</i>	0.2
	t_{S_1}	0.6

Table 4: Utility values for Heater at half power

i	$\sigma_{2,i}$	$u(\sigma_{2,i})$
1	<i>isSuggestedForSensation.Cold</i>	0.4
2	≥ 3 <i>available_kWh</i>	0.3
3	≤ 8 <i>outsideTemperature</i>	0.3
	t_{S_2}	0.6

ues in Table 4 model the fact that the electrical heater is more beneficial when self-produced electric power is available, while those in Table 5 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 3, 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 highest utility, and (v) activates selected functionalities. In our example, no service is active at the time of request, 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 $\sigma_{1,3}$ on temperature are in conflict, as well as constraints $\beta_{A,2}$ and $\sigma_{1,2}$ on available energy. Therefore negotiation is carried out as 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 $u = u_\beta u_\sigma = 0.64$.

Negotiation is executed in the same way in all the other cases. Utility outcomes are summarized in Table 6. It can be noticed that in scenario A (when solar power is available) 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 heater from a thermodynamic standpoint. Conversely, in scenario B (when no self-produced power is available) the heat pump is preferred. Nevertheless, it is useful to note 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

Table 5: Utility values for Heater at full power

i	$\sigma_{3,i}$	$u(\sigma_{3,i})$
1	<i>isSuggestedForSensation.Cold</i>	0.6
2	≥ 6 <i>available_kWh</i>	0.2
3	≤ 2 <i>outsideTemperature</i>	0.2
	t_{S_3}	0.6

Table 6: Case study negotiation results

Request	A	B
Heat pump	0.64	0.8
Heater half power	0.8	0.56
Heater full power	0.9	0.64

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 ontology-based agent infrastructures in the literature, which support only rule-based inferences and exact matches.

5. CONCLUSION AND FUTURE WORK

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

Besides protocol optimizations, future work will include the design of a structured crawler running as part of the user mobile agent, able to automatically build user profiles examining phone PIM (Personal Information Manager), SMS and call lists. That will increase context-awareness of the user agent and enable proactive decision-making on behalf of the user.

6. REFERENCES

- [1] S. Abras, S. Pesty, S. Ploix, and M. Jacomino. Advantages of MAS for the Resolution of a Power Management Problem in Smart Homes. In Y. Demazeau, F. Dignum, J. Corchado, and J. Pérez, editors, *Advances in Practical Applications of Agents and Multiagent Systems*, volume 70 of *Advances in Intelligent and Soft Computing*, pages 269–278. Springer Berlin / Heidelberg, 2010.
- [2] A. Addis and G. Armano. DomoBuilder: A MultiAgent Architecture for Home Automation. In *Proceedings of the 11th Workshop Dagli Oggetti Agli Agenti (WOA 2010)*, volume 621 of *CEUR Workshop Proceedings*, September 2010.
- [3] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2002.
- [4] S. Bechhofer, R. Möller, and P. Crowther. The DIG Description Logic Interface. In *Proceedings of the 16th Int. Workshop on Description Logics (DL'03)*, volume 81 of *CEUR Workshop Proceedings*, Rome, Italy, September 2003.
- [5] Y. Benazzouz, N. Sabouret, and B. Chikhaoui. Dynamic Service Composition in Ambient Intelligence Environment. In *IEEE Int. Conf. on Services Computing, 2009. SCC '09.*, pages 411–418, September 2009.

- [6] D. Bonino, E. Castellina, and F. Corno. The DOG gateway: enabling ontology-based intelligent domestic environments. *IEEE Transactions on Consumer Electronics*, 54(4):1656–1664, november 2008.
- [7] D. Bonino, F. Corno, and F. Razzak. Enabling machine understandable exchange of energy consumption information in intelligent domestic environments. *Energy and Buildings*, 43(6):1392–1402, 2011.
- [8] 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*, 12(2):127–154, 2007.
- [9] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman. Scenarios for ambient intelligence in 2010. IST Programme Advisory Group (ISTAG), 2001.
- [10] J. Froehlich, L. Findlater, and J. Landay. The design of eco-feedback technology. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1999–2008, New York, NY, USA, 2010. ACM.
- [11] 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)*, pages 21–28, 11 May 2010.
- [12] L. Klein, J. young 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*, 22(0):525–536, 2012.
- [13] M. J. Kofler, C. Reinisch, and W. Kastner. A semantic representation of energy-related information in future smart homes. *Energy and Buildings*, 47(0):169–179, 2012.
- [14] Konnex Association. *Handbook for Home and Building Control: Application Interface Layer*, volume 3, chapter 4.1, pages 12–16. 1.0 edition, 2001.
- [15] Konnex Association. *Handbook for Home and Building Control: Application Layer*, volume 3, chapter 3.7, page 8. 1.0 edition, 2001.
- [16] Konnex Association. *KNX Handbook for Home and Building Control*. 5.0 edition, 2006.
- [17] 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*, IWANN '09, pages 585–593, Berlin, Heidelberg, 2009. Springer-Verlag.
- [18] H. Nakashima, H. Aghajan, and J. C. Augusto, editors. *Handbook of Ambient Intelligence and Smart Environments*. Springer, New York, 2010.
- [19] 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)*, 4(3):85–112, 2007.
- [20] 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*, pages 14–25, 2009.
- [21] C. Ramos, J. C. Augusto, and D. Shapiro. Ambient Intelligence - the Next Step for Artificial Intelligence. *IEEE Intelligent Systems*, 23:15–18, March 2008.
- [22] A. Rogers, S. Maleki, S. Ghosh, and J. Nicholas R. Adaptive home heating control through gaussian process prediction and mathematical programming. In *Second Int. Workshop on Agent Technology for Energy Systems (ATES 2011)*, pages 71–78, May 2011.
- [23] 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*, 2(4):424–452, 2006.
- [24] 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)*, pages 177–182. IARIA, 2007.
- [25] 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*, pages 145–148. IEEE, 2009.
- [26] 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*, 7(4):731–739, 2011.
- [27] M. Santofimia, F. Moya, F. Villanueva, D. Villa, and J. Lopez. Intelligent Agents for Automatic Service Composition in Ambient Intelligence. In Z. Usmani, editor, *Web Intelligence and Intelligent Agents*, pages 411–428. InTech, 2010.
- [28] N. Shadbolt. Ambient intelligence. *Intelligent Systems, IEEE*, 18:2–3, July 2003.
- [29] D. Tapia, A. Abraham, J. Corchado, and R. Alonso. Agents and ambient intelligence: case studies. *Journal of Ambient Intelligence and Humanized Computing*, 1:85–93, 2010.
- [30] K. I. K. Wang, W. H. Abdulla, and Z. Salcic. Ambient intelligence platform using multi-agent system and mobile ubiquitous hardware. *Pervasive Mob. Comput.*, 5:558–573, October 2009.
- [31] 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*, 42(1):15–31, jan. 2012.
- [32] J. young 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.