Proceedings of the 2011 IEEE
International Conference on Mechatronics
April 13-15, 2011, Istanbul, Turkey

A Semantic-based Evolution of EIB Konnex Protocol Standard

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

***Abstract*—Domotics aims to improve features and capabilities of building systems and appliances. In spite of increasing self-management of electrical equipments, current solutions refer to static scenarios and require an explicit and elementary interaction with the user. Hence the real pervasiveness and context-awareness of enabled functions is still poor. The integration of knowledge representation and reasoning techniques (originally devised for the Semantic Web) in most common building automation standard protocols can allow to reach higher levels of autonomicity and controllability, so improving user comfort and building efficiency. This paper proposes a semantic-based evolution of EIB Konnex (EIB/KNX) protocol able to interface users (whose needs are expressed via their annotated profiles) with an ontology-based home appliance infrastructure in a fully automated fashion. The proposed approach has been simulated in a prototype in order to test feasibility and prove benefits. Main outcomes are reported.**

***Index Terms*—Building Automation; EIB Konnex; Pervasive Computing; Semantic Web; Resource Discovery**

I. Introduction

Home and Building Automation (HBA), also known as *domotics*, is a technological effort aiming at making houses and buildings more controllable, autonomous and comfortable. In latest years, research in the pervasive computing field has introduced remarkable innovations in the integration of computational capabilities into ordinary objects and everyday activities. A wide array of *Building Automation and Control Systems (BACS)* have been developed and implemented in commercial products and appliances to improve energy savings, minimizing waste and maintenance costs (reflecting guidelines of the European standard EN 15232 [1]). However, current HBA systems still require explicit interactions with the user and furthermore they are typically configured according to a static set of operational scenarios defined during the system design. As a consequence, a low adaptability and flexibility is permitted in both specification of user's needs and management of available resources.

The exploitation in most common BACS protocols of knowledge representation and reasoning techniques and technologies (currently adopted in the Semantic Web effort) is thought as a means to reach higher levels of pervasiveness and controllability, resulting in an improvement of user comfort and building effectiveness. This paper proposes an evolution of the application layer in one of the most widespread standards for HBA, *i.e.*, EIB Konnex (KNX) [2], supporting semantic-enhanced characterization of both user requirements and domestic devices/services/resources. A pervasive and context-aware computing framework is combined with decision support features enabled by logic-based inferences [3]. The integration of a semantic layer within the KNX protocol stack preserves a full backward compatibility, while introducing the following improvements: (i) the standard is enriched with an advanced service and resource discovery support; (ii) exploiting non-monotonic matchmaking, the user profile –annotated w.r.t. a shared ontology– can be compared with house configuration (which includes devices settings and appliances behavior) so identifying the home features best fitting user needs; (iii) advanced equipments (*e.g.*, media centers or meteorological detectors, whose profile is semantically annotated by means of data fusion algorithms which take into account surrounding and environmental conditions) can address requests for services and/or resources to the home government unit; (iv) the household equipments are autonomously able to reach the status better satisfying users activities or devices requirements. These aspects distinguish the proposal also from other ontology-based HBA approaches, which lack deep integration with standards, support for non-exact matches and device-initiated home self-configuration.

The proposed approach has been implemented in a prototypical application, in order to test the feasibility of theoretical framework and to evaluate benefits deriving from the exploitation of semantics in BACS. IEEE 802.11 and Bluetooth are the reference communication interfaces for user-building interaction, whereas IP and UDP are adopted in lower network layers. Furthermore, used ontological formalisms are based on Description Logics (DLs) [4]: particularly DIG [5] has been adopted, which is a more compact equivalent of OWL-DL [6] ontology language. In order to minimize communication latencies in bandwidth-constrained KNX networks, the framework includes a compression tool based on an efficient algorithm aiming at resizing document instances expressed in verbose ontological languages [7].

The rest of the paper is organized as follows. In Section II, most widespread domotics standards and relevant related work are briefly surveyed. Section III describes the proposed framework architecture, connoting the KNX standard extension. Section IV illustrates the proposed approach by means of a reference scenario, whereas Section V reports a system performance evaluation. Finally conclusion and future work are presented in Section VI.

978-1-61284-985-0/11/$26.00 ©2011 IEEE

II. Basics

A. Building automation standards

BACS aim at automatically controlling and coordinating different appliances and utility subsystems in a building, simplifying activities that people normally do manually. Main classifications of BACS refer to architectural topology and infrastructure: (i) *centralized*, if a control unit supervises the whole system; (ii) *distributed*, if sensed information is locally processed by autonomous controllers, each supervising specific appliances and/or areas; (iii) *mixed*, *i.e.*, peripheral controllers are able to acquire and process information for groups of devices while a central building supervising unit acts as coordinator among local controllers.

Most important standards are Konnex, X-10, LonWorks and ZigBee. Their main features are summarized in Table I. Options for building the physical infrastructure include installing a separate wired network, using existing home wirings (power or phone lines) and adopting wireless technologies.

Technology	Transmission medium	Max bit rate	Cost
KNX	TP, PL, RF, IP	9600 bit/s	Low
X-10	PL	60 bit/s	Low
LonWorks	TP, PL, RF, IP, OF	1.28 Mbit/s	Medium
ZigBee	RF	250 kbit/s	Medium/High

TP: twisted pair, PL: power line, RF: radio frequency, IP: Ethernet, OF: optical fiber

TABLE I
HOME AUTOMATION TECHNOLOGIES COMPARISON

The framework proposed in this paper is based on the open standard by KNX Association, a noteworthy initiative aiming at global device interoperability ensuring high flexibility in the extension and modification of installations. KNX is supported by several key industry players, which created a large and competitive market of compatible low-cost solutions for residential buildings, whereas LonWorks is more focused on industrial automation and ZigBee is more expensive. KNX results from the convergence of three existing protocols for HBA: EIB (European Interconnection Bus), EHS (European Home Systems) and BatiBus. It mostly derives from the evolution of EIB, presenting a full backward compatibility. As shown in Table I, KNX supports several physical communication media: twisted pair, power-line, radio frequency and Ethernet (in this case it is also known as EIBnet/IP or KNXnet/IP). A KNX installation consists of several devices interconnected via a communication and control bus, realizing a fully distributed network. Devices are unambiguously identified and subsequently accessed within the network by their 16-bit individual address, reflecting the *area.line.device* topology. KNX also supports full multicast addressing mainly used for the runtime communication, so considerably reducing bandwidth requirements. The standard provides a 16-bit address space for device groups in the format *main.middle.little Group.*

B. Related work

Despite considerable benefits introduced in energy saving and building safety and security, current domotic systems present significant shortcomings. Only static and not-flexible architectures can be defined, with limited functionality and cooperation among different home subsystems. Furthermore, working scenarios, enabled during the setup phase of house equipments, are hardly modifiable and strongly based on user-driven interaction. They trivially allow to adjust multiple home devices with a single command. In order to improve system flexibility and to enable user-transparent interaction among devices, several suggestions are proposed in literature. However, most of them (see [8] as an exhaustive example) aim to enable a more sophisticated management using local or Internet-based remote control. Dynamic and autonomous interaction between user and environment appears as a very difficult goal, if neither attempts at enhancing the expressiveness of user and device profiles are given nor an improvement of resource discovery and decision support via the exploitation of articulated inferences are present.

In [9], a first approach was proposed towards an intelligent house management, including a decision support module. A domotic house gateway was developed to allow communication among different devices from heterogeneous domotic systems and appliances. The system also automates device cooperation using an embedded rule-based engine, which accommodates user needs. A further extension of that work was proposed in [10], introducing knowledge representation techniques to annotate household equipments and appliances. A technology-independent ontology, modeling house and related services, was designed to fit real world BACS and improve previous static rule-based approach. Nevertheless, the exploited engine was still based on trivial cause/effect rules, described adopting SWRL (Semantic Web Rule Language). THere, specific events are detected according to predefined patterns whereas implicit inferences are not possible. Therefore, though the proposal introduced relevant innovation aspects, several issues remain not solved yet. The presented framework was based on a centralized architecture managing the reference knowledge base including both ontology and devices and domestic environment descriptions. In this way, it is mandatory to create for each device an association with the related description in the system start-up phase. Moreover, rules-based matchmaking is not completely adequate in dynamic scenarios like the HBA ones, where full matches are quite unlikely. Finally, w.r.t. the framework proposed here, a complete and user-independent orchestration of elementary household functionalities, to build more complex services, is not enabled.

The semantic-based evolution of EIB/KNX protocol standard proposed here allows to solve the above issues, so increasing the autonomicity of BACS. The increased complexity of applications and interaction is fully justified by the introduced benefits. Similar approaches were proposed also for most widespread ad-hoc networks (*e.g.*, Bluetooth piconets) and for a wide range Internet of Things exploitation based on RFID [11]. In the overall cases, introduced advances resulted very useful and effective: prototypical implementations proved feasibility and added-value.

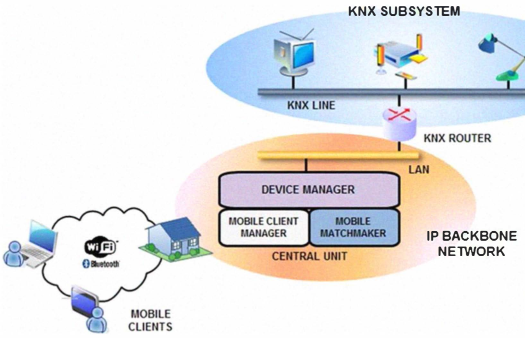

Fig. 1. Proposed framework architecture

III. Framework

A. Framework architecture

The approach proposed here is founded on an architecture which integrates existing HBA systems, based on EIB/KNX bus, with an IP network used as fast backbone. This hybrid home network, compliant with the EIBnet/IP routing protocol, enables communication among different KNX lines via IP. The framework architecture is shown in Figure 1. It consists of four main functional blocks: (i) ***Central Unit***: represents the system core unit and includes a *device manager* and a *mobile client manager*. It also includes the *mobile matchmaker* in [12], exploiting standard and non-standard inference services in Description Logics (DL), to carry out a semantic matchmaking process (see [13] for theoretical background) between a semantic request and device annotations; (ii) ***KNX Router***: converts the KNX/EIB telegrams into IP frames and vice-versa according to the EIBnet/IP standard; (iii) ***Semantic-based devices***: KNX home devices implementing the enhancements described in Section III-B; (iv) ***Mobile clients***: mobile devices, such as notebooks, smartphones or PDAs, equipped with a software agent able to send semantic-based requests (grounded on specific user profiles) and to receive replies. Communication between clients and the home system may occur through either IEEE 802.11 or Bluetooth standards.

Particularly, the central unit allows to: (i) retrieve device *service* (*i.e.*, functional profile) descriptions, expressed as annotations in standard logic-based knowledge representation languages; (ii) rank best services that must be activated to fulfill users' or system requirements, by computing a relevance score w.r.t. to the received request; (iii) find possible inconsistencies between system status and selected services; (iv) explain the matchmaking outcome, showing possible issues and negotiation options to the user.

Main novelty introduced by the approach is the management of requests coming not only from user agents, but also from devices. In this way, home appliances can embed device agents, able to provide services and autonomously issue requests. This is a unique feature w.r.t. current HBA standards, enabling more dynamic environment adaptation through decentralized device cooperation, whereas the central control unit basically acts as service broker.

With reference to the sequence diagram in Figure 2, let us

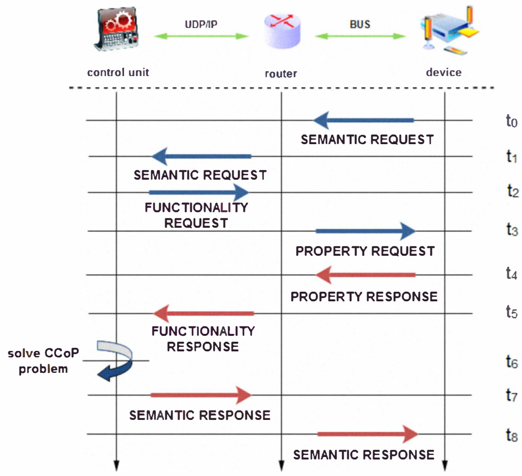

Fig. 2. Device semantic request

analyze the tasks performed by the system when a request comes from a device. At t_0, a device encapsulates a semantic request in one or more KNX frames, forwarded toward the central unit over the IP network. If requests are originated by a mobile client, the central unit directly receives a semantic-based query generated by the user, based on her psychophysical profile, and processing starts from time t_2. In either case, the central unit will perform an orchestration of suitable services via a semantic-based covering process formalized as in what follows:

1. A *service request* message is sent to KNX router in order to retrieve all services from suitable devices.

2. For each device, the router sends a specific *property request* message to obtain service descriptions. In order to save memory and bandwidth, annotations are compressed by means of the algorithm in [7].

3. Data received from devices are then forwarded to the central unit, decoded and temporarily stored in the local memory.

4. Using semantic annotations related to both received request and services, the mobile matchmaker executes the *covering* Algorithm 1, based on the solving of *Concept Covering Problem (CCoP)* [13], a non-standard DL-based inference service. In the first part (lines 3-8), a compatibility check is performed to find active services conflicting with the request, deactivated subsequently. Then in line 9 a Concept Abduction inference task [3] is performed in order to verify whether request is already completely covered or the activation of further services is needed. In the latter case, CCoP is solved (line 16) to select one or more inactive functionalities, whose combination can cover missing features; a preliminary check is performed to skip inactive services contrasting with currently active ones (lines 11-15). Finally, the algorithm returns the set of services to be activated, the (possibly empty) set of ones to be disabled and a description of uncovered request part, if present.

5. Selected services are activated through standard EIB/KNX telegrams and a *semantic response* message is sent to the requesting device. Instead, if request came from a mobile client, a message is sent to the user reporting outcomes.

Algorithm 1 *requestCovering* $(R, A, NA, \mathcal{L}, \mathcal{T})$

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 (resp. not active) functionalities in $\mathcal{L}$ satisfiable in $\mathcal{T}$.
Ensure: $G = \{G_1, G_2, \ldots G_k\}$ set of services to activate; $K = \{K_1, K_2, \ldots K_h\}$ set of services to deactivate; H uncovered part of the request.

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

B. EIB KNX semantic extension

The proposed approach defines a new semantic layer on top of the EIB/KNX protocol stack. New elements and capabilities are introduced, while keeping a backward compatibility. Features of interconnected devices can be now fully described by means of encoded annotations in semantic-based languages such as OWL or DIG.

A preliminary study of KNX standard highlighted protocol inadequacy to manage semantically annotated information. Hence two new Interface Objects have been defined for annotation storage, containing a structured and machine-understandable device description. New object types are compliant with the original structural specifications reported in [14]. Interface Objects are particular data structures used in KNX to specify device properties. Basically they group several ***Object Properties***, each composed of a *property description* and a *property value*. Property descriptions consist of: (i) a Property IDentifier (PID) code, (ii) a *Property Data Type* (PDT) code, (iii) a value indicating the maximum number of elements *(max_no_of_elem)* and (iv) a field related to property access rights. The property value is instead an array with *max_no_of_elem*+1 elements. The component at index *0* contains the current number of valid elements. One Object Property is mandatory for all Interface Objects, the ***Object Type***. It is a 16-bit unique identifier. KNX standard defines different value ranges for this code according to the specific object purpose.

The first newly introduced Interface Object type is called ***Generic Profile of the Device (GPD)***. It describes generic device features *e.g.*, manufacturer or model. Due to the above constraints, GPDs have Object Type field set to the unused value *1200*. The Object Type value code *1205* is used instead to identify a ***Specific Profile of the Device (SPD)*** object, which contains a semantic-based annotation of a service (*i.e.*, functional profile or operating mode) provided by a device, expressed w.r.t. a reference ontology that models knowledge for the domain the service belongs to. If a device exposes different available services, for each one of them a distinct SDP will be defined. In order to define semantic properties, specific PIDs are introduced in addition to the Object Type field. KNX specifications separate PIDs in three categories: (i) *Standardized object independent ID* $[0-50]$; (ii) *Standardized object dependent ID* $[51-154]$; (iii) *Not standardized ID* $[155-255]$. According to this classification, GPD and SPD objects include properties with the following identifiers:
- *PID_OBJ_TYPE* = 1 ($0x01_h$): a 16-bit mandatory field indicating the Object Type;
- *PID_OUUID* = 77 ($0x4D_h$): 16-bit Ontology Universally Unique Identifier (OUUID) marking the reference ontology;
- *PID_OUUIDs* = 100 ($0x64_h$): specifies the OUUID group. This property is only present in GPDs, because a device may offer services described w.r.t. different ontologies, whereas each service must refer to a single ontology;
- *PID_SEMANTIC_HEADER* = 150 ($0x96_h$): header of compressed annotation, stored as a variable-length string;
- *PID_SEMANTIC_BODY* = 151 ($0x97_h$): body of encoded annotation, defined as a string.

Splitting the encoded annotation into a header and a body can allow more efficient querying of the annotation, particularly in case of *homomorphic* compression [7]. Modifications to KNX standard include also the definition of a new *DataPoint Type (DPT)* to characterize new semantic properties, ***DPT_OUUID***, identified by the code *7.1000*. Main number is 7 because OUUID is a 16-bit unsigned value whereas 1000 is the first value present in sub-number range reserved by KNX standard for future applications (1000-59999).

Data structures introduced so far are needed to support semantic-enhanced application-layer services. Particularly, two service primitives are added, allowing devices to exchange semantic annotations embedded in a standard *Application layer Protocol Data Unit (APDU)*. Each primitive is identified by a code included in *Application layer Protocol Control Information (APCI)* field. According to the definition presented in [2], the first two APCI codes reserved for user messages are used to define respectively: (i) ***A_SEMANTIC_SUBMISSION.req*** service, reflecting the KNX implementation of device semantic request shown in Figure 2. It will be used to send a semantic description generated by a device and based on events detected in the environment; (ii) ***A_SEMANTIC_SUBMISSION.res*** service, corresponding to semantic reply containing description of selected household functionalities covering the request.

Despite semantic annotations are compressed, they may often exceed maximum APDU data field size, fixed to 14 bytes. For this reason, we use *extended frames* –allowed by KNX protocol– having a size of 255 bytes, 249 of which reserved for data. Since there is no PDT for such data size, and for strings in particular, a new one has been introduced. *PDT_GENERIC_EXT*, associated to $0x20_h$ code, defines a data block with a maximum size of 249 bytes. If the semantic annotation still results longer than APDU field, then it is split in different APDUs. The total number of packets is present into the *number of elements* field of the related Interface Object.

IV. Ontology-based Intelligent Buildings: a Case Study

In order to illustrate the proposed approach, different scenarios have been defined involving typical home situations. In this section, we refer in particular to a case study simulating a house equipped with 7 semantic-based devices (*Light Controller, DVD Player, DVD Player for deafness, Music Player, Temperature Controller, Security Controller, Safety Controller*) providing 27 services. A specific ontology in $\mathcal{ALN}$ (Attributive Language with unqualified Number Restrictions) DL was defined to describe user needs, devices and building environmental parameters. Figure 3 shows a relevant excerpt of it. Hereafter a simple request example is provided to point out the main features of the proposed knowledge-based approach. *An adult man comes back home after a workday. He is tired and has a splitting headache. For these reasons, he is very nervous and wishes a relaxing home environment. Moreover it is a warm evening and he feels hot, so sends a semantic request to his home by means of the client running on his mobile phone.*

A possible formalization of user profile and request (reported in classical DL notation for the sake of readability) w.r.t. the reference ontology is:

User_Profile ≡ ∀suitable_For_Wish_Service.Relax ⊓ ∀suggested_For_Age.Adult ⊓ ∀suggested_For_Gender.Male ⊓ ∀suggested_For_Mood.Nervous ⊓ ∀suggested_For_Disease.Headache ⊓ ∀suggested_For_Tired.Mentally_Tired ⊓ ∀suggested_For_Temperature_Perception.Hot.

This description includes user features, such as age, gender, physical status, mood, as well as desired service category, in this case relax functionalities. Mobile client can also include music and movie preferences. In this way, an accurate home service selection can be performed. The central unit receives and processes the request as explained in Section III-A. At the end of the matchmaking process, the system proposes 7 services, shown in Figure 4. Their descriptions are reported hereafter, where portions covering the request are in bold:

Play_DVD_SciFi ≡ ∃suggested_For_Video_Preferences ⊓ ∀suggested_For_Video_Preferences.SciFi ⊓ ∃suitable_For_Wish_Service ⊓ ∀suitable_For_Wish_Service.Play_DVD ⊓ ∃suggested_For_Age ⊓ **∀suggested_For_Age.NotChild** ⊓ **∀suggested_For_Gender.Male** ⊓ ∃suggested_For_Handicap ⊓ ∀suggested_For_Handicap.no_Handicap.

Play_DVD_Historical ≡ ∀suggested_For_Stamina.Rested ⊓ ∃suggested_For_Video_Preferences ⊓ ∀suggested_For_Handicap.no_Handicap ⊓ ∀suggested_For_Video_Preferences.Historical ⊓ ∃suitable_For_Wish_Service ⊓ ∀suitable_For_Wish_Service.Play_DVD ⊓ ∃suggested_For_Age ⊓ **∀suggested_For_Age.NotChild**.

Play_Classical_Music ≡ ∀suggested_For_Handicap.no_Deaf ⊓ **∀suggested_For_Stamina.Mentally_Tired** ⊓ ∃suggested_For_Video_Preferences ⊓ ∀suggested_For_Musical_Preferences.Classical ⊓ ∃suitable_For_Wish_Service ⊓ ∀suitable_For_Wish_Service.Play_Music.

```
Service ⊒ {
    Play_Radio
    Play_DVD
    Play_TV
    Environment_Temperature_Regulation
    Light_Level_Regulation ⊒ { Bright_Light_Level, Soft_Light_Level }
    Safety_Level_Regulation ⊒ { Normal_Safety_Level, High_Safety_Level }
    Security_Level_Regulation ⊒ { Low_Security_Level, Medium_Security_Level, High_Security_Level }
}
Characteristic ⊒ Physical_Status ⊒ {
    Disease ⊒ { Fever, Headache }
    Temperature_Perception ⊒ { Normal, Hot, Cold }
}
```

Fig. 3. Relevant axioms in the case study ontology

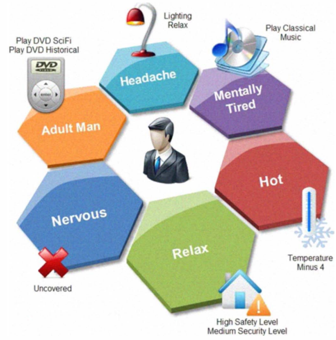

Fig. 4. Concept Covering result

Temperature_Minus4 ≡ ∃suggested_For_Temp_Perception ⊓ **∀suggested_For_Temp_Perception.Hot** ⊓ ∃suitable_For_Wish_Service ⊓ ∀suitable_For_Wish_Service.Environment_Temp_Regulation.

Lighting_Relax: ≡ ∃suggested_For_Disease ⊓ **∀suggested_For_Disease.Headache** ⊓ ∃suggested_For_Stamina ⊓ ∀suggested_For_Stamina.Eye_Tired ⊓ ∃suitable_For_Wish_Service ⊓ ∀suitable_For_Wish_Service.Soft_Light_Level.

The selected service set includes two suggestions for DVD playback, following movie genre preferences specified in the user profile; the final choice is left to the user. Classical music is selected because it fits tiredness, according to the reference ontology. A lower room temperature and soft lighting settings are selected to improve user comfort. Finally, system sets home safety and security levels: even though no explicit mention occurs in user request, axioms in the ontology allow the mobile matchmaker to infer the appropriate safety and security settings for maximum relax. An uncovered part of the request is also present, because there are no specific services able to match nervous user state. In such cases, the system computes the maximal approximate covering and provides the user with an explanation of what part of the request is not satisfied by the composite services.

Request descriptions can be also completely composed by device-driven statements. In this case, further different usage scenarios can be modeled, including complex events involving both user requests and the automatic detection of environmental features.

V. Simulations and Results

Performance evaluation of the proposed approach has been carried out simulating the previously described case study. The whole framework has been implemented using Java language and a modified version of Calimero 2.0 library [15] for the central unit. Tests were performed using two laptop PCs acting as central unit and mobile client, equipped with an Intel Core 2 Duo T9300 CPU at 2.5 GHz, 4GB DDR2 RAM and Microsoft Windows Vista operating system with Java Virtual Machine 1.6.0_14 and an Intel Atom CPU at 1.6 GHz, 1GB DDR2 RAM and Microsoft Windows XP operating system with Java Virtual Machine 1.6.0_17, respectively. In order to assess the performance impact of the communication medium between router and smart devices, a *TP1* cable was simulated by opportunely delaying each frame on the LAN to force a maximum transmission rate of 9600 bps. Two kinds of test were performed to evaluate: (i) *service retrieval time* –time needed to retrieve the semantic annotations from

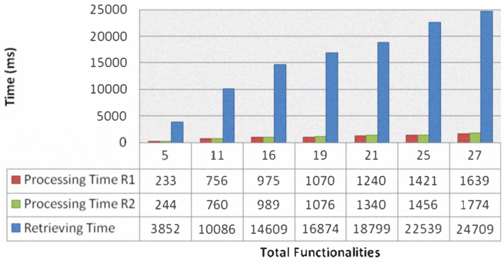

Fig. 5. Service retrieval and processing time

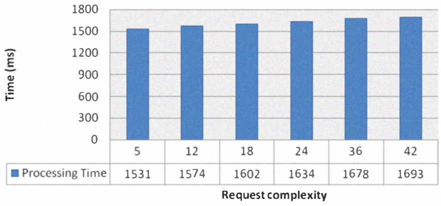

Fig. 6. Processing time with increasing request complexity

all connected devices; (ii) *processing time* –time used to solve CCoP problem and return selected services. Request complexity has been characterized by a function defined as the sum of the number of concepts and the maximum nesting depth of property relationships in the logic expression: $complexity(R_i) = concept\_number(R_i) * max\_depth(R_i)$

In the first test, two requests R_1 and R_2 (with a complexity value of 25 and 35, respectively) were processed, gradually increasing the number of available services. Figure 5 reports results, showing a linear dependency between the number of services and retrieval time. Due to constraints of low-throughput fieldbus, retrieving annotations resulted by far the longest step, taking about 94% of overall turnaround time. Therefore optimizations are needed for subsequent system versions: cache implementation is being performed when writing, in order to allow pre-loading of descriptions during system startup. Preliminary assessments have suggested that request turnaround time can be reduced by about 80%. Processing time is slightly affected by the increase in available services, though absolute values remain rather acceptable.

In the second test, different requests with increasing complexity were processed, while keeping the number of available services at 27. As ensues from results in Figure 6, the complexity of requests weighs moderately on system performance. In fact, the most complex request required only additional 162 ms (11%) w.r.t. the simplest one, despite a complexity difference of 37 (740%). This is a significant outcome because the system can remain responsive even in complex scenarios, thanks to the efficient embedded mobile matchmaker.

VI. Conclusion and Future Work

The integration of knowledge representation and reasoning techniques in current standards and technologies for home and building automation reveals several benefits as it allows to improve user comfort and building efficiency and to increase the autonomicity and controllability of household environments. The proposed approach introduces a semantic-based extension of application layer in EIB/KNX standard, preserving legacy applications and opening interesting possibilities for further enhancements. The system has been simulated in order to test the feasibility and the usability of the proposed solution, obtaining significant results.

Future framework extensions will see the integration of concurrency control and caching of both device and user profile annotations, in order to improve system reliability and reduce response times. The proposed framework is being implemented in a real testbed to fully evaluate the approach..

References

[1] UNI EN 15232, "Energy performance of buildings - Impact of Building Automation, Controls and Building Management," 2007.

[2] Konnex Association, *Handbook for Home and Building Control: Application Layer*, 1st ed., 2001, vol. 3, ch. 3.7.

[3] 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," *International Journal of Web Services Research (JWSR)*, vol. 4, no. 3, pp. 85–112, 2007.

[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 International Workshop on Description Logics (DL'03)*, ser. CEUR Workshop Proceedings, vol. 81, Rome, Italy, September 2003.

[6] W3C, "OWL Web Ontology Language Overview, W3C Recommendation," Online http://www.w3.org/TR/owl-features/, February 2004.

[7] F. Scioscia and M. Ruta, "Building a Semantic Web of Things: issues and perspectives in information compression," in *Semantic Web Information Management (SWIM'09). In Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009)*. IEEE Computer Society, 2009, pp. 589–594.

[8] M. E. Pardo, G. E. Strack, and D. C. Martinez, "A Domotic System with Remote Access based on Web Services," *Journal of Computer Science & Technology*, July 2008.

[9] P. Pellegrino, D. Bonino, and F. Corno, "Domotic House Gateway," *ACM Symposium on Applied Computing (SAC 2006)*, April 2006.

[10] 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.

[11] T. Di Noia, E. Di Sciascio, F. M. Donini, M. Ruta, F. Scioscia, and E. Tinelli, "Semantic-based Bluetooth-RFID interaction for advanced resource discovery in pervasive contexts," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 4, no. 1, pp. 50–74, 2008.

[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 International Conference on Web Intelligence*. IEEE, 2009, pp. 145–148.

[13] T. Di Noia, E. Di Sciascio, and F. Donini, "Semantic matchmaking as non-monotonic reasoning: A description logic approach," *Journal of Artificial Intelligence Research (JAIR)*, vol. 29, pp. 269–307, 2007.

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

[15] B. Malinowsky, G. Neugschwandtner, and W. Kastner, "Calimero: Next generation," in *Proc. KNX Scientific Conference 2007*, Nov. 2007.