An approach to improve semantics in Smart Spaces using reactive fuzzy rules

Natalia Díaz Rodríguez
and Johan Lilius
Turku Centre for Computer Science (TUCS),
Department of IT,
Åbo Akademi University, Turku, Finland
Email: {ndiaz, jolilius}@abo.fi

M. P. Cuéllar
and Miguel Delgado Calvo-Flores
Department of Computer Science and AI,
University of Granada, Spain
Email:{manupc, mdelgado}@decsai.ugr.es

Abstract—**Human behaviour representation and modelling in Smart Spaces are crucial tasks in Ambient Intelligence environments. A problem found among existing technologies is that the environment sensor data is always provided as crisp events. However, to model behaviour in reactive Smart Spaces we need to identify user patterns, which are not always performed in the same way or happening at the same time/frequency. Hence, the handling of imprecision is essential to model human behaviour in a realistic way. In order to connect these two paradigms, we suggest a combined architecture to fill the gap between quantitative Smart Space architectures (which provide raw crisp events) and qualitative aspects such as fuzzy reasoning and learning algorithms, that extract intelligence from data. With this aim, we propose to use Semantic Web principles of independence and interoperability to build an integrated framework to take advantage of the benefits of both crisp and fuzzy models. Our contribution in this work is a proof of concept to demonstrate how a hybrid architecture, with a reactive rule-based subscription mechanism, can empower users to model everyday activities semantically or control what happens on their surroundings, among other advantages.**

I. INTRODUCTION

The idea of ubiquitous space was proposed as an ideal world where humans and surrounding devices interact unobtrusively. A *Smart Space* (SS) [29] is any physical environment equipped with sensors and actuators able to perceive the human activity and environmental conditions, to make decisions from these perceptions, and to modify the space according to the system goal. Smart Spaces support the vision in which computers work on behalf of users, they have more autonomy, and they are able to handle unanticipated situations. Therefore, the development of a SS implies the usage of Artificial Intelligence (AI) and machine learning, among other technologies.

SSs are considered to be context-aware systems; therefore, a key requirement to design such systems is to give computers the ability to understand situations and environmental conditions [8]. To achieve this, contextual information should be represented suitably for machine processing and reasoning. Semantic technologies, and more specifically ontologies [12], [11], are well suited for this purpose because ontologies allow to share knowledge while minimizing redundancy. In addition, they are good tools for knowledge representation and reasoning.

On the other hand, the formal specification of human behaviour is difficult to handle when crisp reasoning mechanisms are used for this purpose, since natural human patterns are imprecise, imperfect and fully gifted with semantics. In this way, fuzzy ontologies and fuzzy extensions of Description Logics (DL)[3] arise as more appropriate formalisms to deal with the vagueness inherent to real-worlds domains [4]. These formalisms have been shown to be useful in applications on, e.g., information retrieval and image interpretation [7].

We identify a gap among the quantitative aspects of SS, i.e., the infrastructure and technology that handles large amounts of crisp sensor information [22], [34], [24], [20], and the qualitative aspects of SS, i.e., the techniques for human activity modelling and recognition [18], [27], [28], and the formal specification of how the system behaviour should be according to what the system is perceiving [26], [2], [29]. The latter approach involves reasoning about imprecise and vague information. In this way, our purpose is to develop a new methodology able to entitle end-users to work with a vague semantic specification of the SS. This new abstraction level will allow to manage known human activities, their relationships and environmental conditions, to provide expressiveness to the specification of the SS behaviour, and therefore getting closer to the end-user's everyday language.

Regarding the quantitative view of SSs, we identify the need for a semantic store with support for standards (OWL 2, SPARQL, etc.), as well as a scalable enough rule engine with subscription capability for real time rule-based applications. By *scalability* we mean the capability of handling a wide variety of heterogeneous events and data from different users and activities. Supporting the standards is needed to represent fuzzy ontologies, since nowadays, each reasoner uses its own fuzzy DL language [7]. We argue that the integration of these quantitative and qualitative requirements is important to precisely model human behaviour in SS, as well as to ease the development and deployment of semantic and intelligent applications. However, to the best of our knowledge, there is no integrated reasoning and storage solution supporting all our requirements.

To illustrate the expected expressiveness of our approach to program a SS system behaviour, we show the following example: We should be able to model when a user is waking up *very early*, the schedule for the activity *to have breakfast*, and what it does *mean* in terms of the Smart Space behaviour. In our approach, we will develop a system able to represent rules such as *If today is working day and Natalia wakes up later than usual, then she is late*; *When Natalia is very late, she usually does not have breakfast*; etc. The system should be able to infer that, if today is Monday and Natalia wakes up much later than usual, then the possibility to have breakfast is very low, and thus, it is not necessary for the SS to start the coffee machine, turn on the lights of the kitchen, etc.

In the next section, existing knowledge representation techniques such as fuzzy reasoners and required technology such as subscription-based RDF stores, are discussed. In Section III, we motivate and describe a selection of components for its integration in a single hybrid framework. Section III-A proposes a subscription-based RDF store with SPARQL support, *M3*, as basic infrastructure for the crisp Knowledge Base (KB) part of the overall system's rule engine. Section III-B adds the capability of having imprecise modelling and reasoning in the overall architecture through a fuzzy KB, consisting on the *fuzzyDL* reasoner. Section III-C describes how the integration of the previous components in the overall architecture happens and how the experiment for benchmarking the proof of concept is designed. Finally, Section IV discusses the approach and concludes with future directions.

II. Related Work

Typical challenges in human activity modelling are handling missing event data, adaptability to different users, or to changes in activities, etc. Ontology-based activity representation provides a number of advantages [9]: it supports incremental progressive activity recognition, state based modelling and a more robust reasoning since there is no fixed sequences for an activity, especially for *Activities of Daily Living* (ADL). Other benefits are the ability to discriminate importance and urgency of activities through semantic descriptions, support for course grained and fine-grained activity assistance and the possibility for data fusion and semantic reasoning, including activity learning and activity assistance.

For the quantitative part of human behaviour modelling, i.e., the infrastructure, we find a great offer in storage as well as reasoning solutions that push the Smart Spaces vision forward. Our requirement for these solutions is to be able to handle event subscription for data scalability. With regards to the fuzzy paradigm, imprecise knowledge reasoning, fuzzy Description Logics appear as an alternative to crisp DLs which lack the ability to represent uncertain or vague information. We consider a set of available fuzzy reasoners and evaluate a set of expressibility requirements, useful for AmI applications. Furthermore, we identify SPARQL support as a standardization requirement. Table I summarizes the identified requirements for behaviour representation in SSs and the fuzzy reasoners that support them to some extent (marked with *x*). It can be seen that *FiRE* and *DeLorean* allow the use of some DL constructs that *fuzzyDL* does not support (cardinality restrictions and, in the case of *DeLorean*, also nominals). On the other hand, *GURDL* supports a more general representation of uncertainty, not being limited to fuzzy logic [6]. At last, another useful feature in behaviour modelling is stream reasoning to semantically annotate events. This feature can be found in *TrOWL* [33], a tractable reasoning infrastructure of OWL 2 with built-in OWL 2 QL reasoner Quill and EL reasoner REL. *fuzzyDL* supports a series of distinct features with respect to expressivity of the representation, such as explicit fuzzy sets, concepts modifiers, data types and defuzzification [6]. However, none of the fuzzy reasoners includes a listener/observer subscription mechanism for effective changes notifications on real time.

Reasoner	**Fuzzy DL**	**Event Subscript.**	**SPARQL**	**Cardinality Restr.**	**Fuzzy Sets**	**Concept Modifier**	**Fuzzy Data Type**	**Defuzzification**	**Fuzzy Rule**	**Satisfiab. Degree**
FiRE [32], [31], [30]	$\mathcal{F}-\mathcal{SHIN}$			x						x
GURDL [13]	$\mathcal{F}-\mathcal{ALC}$									x
De-Lorean [5]	$\mathcal{F}-\mathcal{SROIQ}$			x	x	x	x			x
GERDS [14]	$\mathcal{F}-\mathcal{ALC}$									
fuzzy DL [6]	$\mathcal{F}-\mathcal{SHIF}$ $(\mathbf{D})$				x	x	x	x	x	x
YADLR [21]	SLG algorithm									x
FRESG [15]	$\mathcal{F}-\mathcal{ALC}(\mathbf{G})$						x			x

TABLE I
COMPARISON OF AVAILABLE FUZZY REASONERS AND THEIR SUPPORT FOR SMART SPACE MODELLING REQUIREMENTS

No fuzzy reasoner includes at the moment subscription features; it is only in crisp RDF stores where this can be found. To the contrary, it is uncommon to have RDF stores including imprecise reasoning. However, there are particular instances such as *f-SPARQL* [10], a "flexible extension of SPARQL", that allows in the *FILTER* constraint, the occurrence of fuzzy terms and fuzzy operators (by using α-cut operation), as well as weights in fuzzy constraints to have different importance and efficiently compute the top-k answers.

Event subscription is not supported in the vast majority of RDF stores. However, we can find exceptions such as *M3* [17], some versions of *OWLIM* [1] or *RDFStore-js* [16]. The latter is a JavaScript implementation of an RDF quad store with support for SPARQL 1.0, most of SPARQL 1.1/update and a significant portion of SPARQL 1.1 query, that can be executed in the browser. The great advantage of event subscription features in semantic repositories is the capability of efficiently get notified when data of interest changes. This feature can avoid bottlenecks normally caused when rule conditions result in a constant checking for the status of specific nodes.

To the best of our knowledge, and as Table I shows, there does not exist a system which comprises support for all our requirements: expressive fuzzy queries, event processing for scalable, efficient and real time applications, as well as the possibility of federating queries with other SPARQL end-points. These are crucial elements when tackling real life problems in SS. For instance, decision support system or expert systems must react on time against forgotten actions, against potential errors produced in an industrial process or when following up a certain procedure with guidelines. To solve these problems, not only must standard query languages be supported to integrate heterogeneous data, but also efficient notification mechanisms must be supported to be alerted only when conditions we are actually interested happen. Additionally, support for every-day imprecise or vague terminology is to be provided. Lacking an efficient subscription/notification based system makes a large rule system impractical to real-time proactive applications. The requirements detailed in this section, together with an easy and reachable end-user language, can allow better modelling of expert knowledge and better involvement of the end-user into the problem modelling process. In next section we discuss concrete components for our proposal and how they can be connected.

III. A FRAMEWORK FOR CONTEXT-AWARE SMART SPACE APPLICATION DEVELOPMENT

After identifying, in the previous section, the main components required for realizing more powerful human activity modelling in SSs, we detailed some technologies which contribute to this aim. As no system was found fulfilling all the needs for the development and deployment of our vision of SSs, we suggest a configuration of technologies that allows us to construct an AmI framework to enable human behaviour representation and recognition through rules that include fuzzy concepts in form of linguistic terms.

The overall system, in Figure 1, is composed by two parts, a crisp KB and a fuzzy KB. These are connected by a main rule engine, which handles the subscription to each type of event condition (fuzzy and crisp) per rule. Next sections detail the components of the system architecture.

A. A subscription-based RDF store with SPARQL support: M3

The crisp element of the KB in the overall system in Fig. 1 is formed by *Smart-M3*, a Multi part, Multi device and Multi vendor platform consisting of independent agents which communicate implicitly by inserting and querying information in the space. *M3* is an open source, cross-domain architecture where the central repository of information, Semantic Information Broker (SIB), is responsible for information storage, sharing and management. Entities called Knowledge Processors (KPs) implement functionality and interact with the Smart Space by inserting/querying common information through the publish/subscribe Smart Space Access Protocol (SSAP) and now, also through SPARQL. Communication happens not device to device but through the SIB. Entities and services are described with OWL (Web Ontology Language).

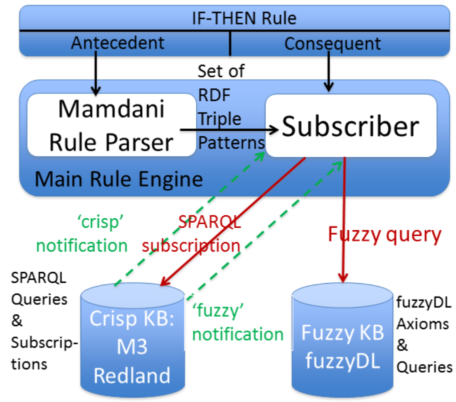

Fig. 1. Overall framework with fuzzy and crisp Knowledge Bases

Benefits of publish/subscribe (or "push") semantic architectures, such as *M3*, include the inherent polling and a strong decoupling of the communication clients with respect to time, reference and data schema, increasing flexibility in application design and allowing for more autonomous system architectures [23]. *M3* supports RDF triple pattern queries as well as WQL [1] and SPARQL queries. Furthermore, it allows to subscribe to a certain triple pattern for efficient awareness of data changes, as well as to join/leave a confined SS. Therefore, the role of a module such as *M3* is to serve as SPARQL persistent storage for (crisp) event processing.

B. Imprecise rules and fuzzy reasoning

Crisp RDF infrastructures can achieve scalable SS applications. However, these are features not always considered to be the main aim of fuzzy reasoners. The latter, on the contrary, provide expressive languages to model, e.g., routine activities or more complex processes. In order to allow not only crisp but also fuzzy rules for describing the behaviour of both, users and a semantic SS as a whole, the SW needs to become more imprecise to accommodate everyday problems and serve distinct kinds of users [25]. This is why the user should be able to vaguely or imprecisely express knowledge. We proceed to explain how a rule with imprecise concepts and/or relations, is mapped to a representation in our fuzzy KB.

In order to reason with human behaviours, as well as the behaviour of the Smart Space system as a whole, we can employ an expressive [2] fuzzy DL reasoner, such as *fuzzyDL* [6]. Let's assume the user wants to define rules such as:

1) IF (*WeatherSituation isCurrently VeryStormy*) OR (*Natalia hasStatus AwayForWeekend*), THEN (*TurnOff-AllElectricitySwitches(NataliasAppartment)*)).
2) IF (*Natalia hasPhone P*) AND (*Natalia hasCalendar C*) AND *(P isInLocation L*) AND (*L isVeryNearTo Jo-*

[1] Wilbur Query Language: http://wilbur-rdf.sourceforge.net/

[2] Allowing needed DL constructs as well as flexible behaviour descriptions

% Concrete features (Classes and Relations) *(instance Natalia Person)* *(define-concept NataliasAppartment (and HousingProperty (some rentedBy Natalia))* *(instance WeatherSituationTurku WeatherSituation)* *(instance TurnOffAllElectricitySwitches ExecutableApplication)* *(instance StartAudioRecording ExecutableApplication)* *(instance TranscribeMeetingAgenda ExecutableApplication)* *(functional isCurrently)* *(functional hasStatus)* % Labels for the variables *VeryStormy = triangular (50,100,150)* % **A) Definition of Logical Rules as Mamdani rules** *(define-concept Rule1 = (g-and (Natalia (some hasStatus AwayForWeekend)) (WeatherSituationTurku (some isCurrently VeryStormy)) (TurnOffAllElectricitySwitches (some withParams NataliasAppartment))))* % Encoding of Mamdami Rule Base *(define-concept MamdaniRuleBase (g-or Rule1 (...) RuleN))* % Input to the controller/Facts *(instance input (= and WeatherSituationTurku (some isCurrently NearlyCloudy)))* *(instance input (= and Natalia (some hasStatus AtWork)))* (...) % Defuzzification *(defuzzify-lom? MamdaniRuleBase input TurnOffAllElectricitySwitches)* % **B) Definition of Logical Rules as implication rules** *(define-concept antecedents1 (and (Natalia (some hasStatus AwayForWeekend)) (and WeatherSituationTurku (some isCurrently VeryStormy))))* *(define-concept consequents1 (and (TurnOffAllElectricitySwitches (some withParams NataliasAppartment))))* *(define-concept Rule1 (l-implies antecedents1 consequents1))* *(define-concept antecedents2 (and (Natalia (some hasPhone P) (and (Natalia (some hasCalendar C)) (and (P (some isInLocation L))) (and L (some isVeryNearTo JohansOffice))))))* *(define-concept consequents2 (and (StartAudioRecording (some withParams P)) (TranscribeMeetingAgenda (some withParams (P and C))))))* *(define-concept Rule2 (l-implies (g-and antecedents2) (g-and consequents2)))* % Input to the controller/Facts % Query for the consequent' satisfiability degree *(min-instance? input consequents1)*

TABLE II
EXAMPLE: KB AND RULES IN *fuzzyDL* FOR RULES 1 AND 2.

SPARQL query	fuzzyDL query	Subscription in M3
SELECT DISTINCT ?user WHERE ?user mo:hasStatus mo:AwayForWeekend. ?user mo:hasName "Natalia"xsd:string.	*(min-related? Natalia AwayForWeekend hasStatus)*	*triple = [Triple(URI(NS+"Natalia"), URI(NS+"hasStatus"), URI(NS+"AwayForWeekend"))] self.st = self.CreateSubscribeTransaction(self.ss_handle) initialResult = self.st.subscribeRDF(triple, subscriptHandler(self))*

TABLE III
EXAMPLE: MAPPING OF RULE ANTECEDENT *"IF Natalia hasStatus AwayForWeekend"* TO SPARQL AND *fuzzyDL* QUERIES

hansOffice), THEN (*StartAudioRecording(P)* AND *TranscribeMeetingAgenda (P, C)*)

These rules follow the Mamdani structure and can be mapped to a set of statements in a fuzzy KB as a fuzzy control system [6]. For instance, for Rules 1 and 2 we would have the mapping to fuzzy axioms in Table II.

We chose *fuzzyDL* because it supports important features for expressing imprecise common knowledge when users model knowledge in SS. *fuzzyDL* provides fuzzy rough set reasoning and fuzzy reasoning for fuzzy $\mathcal{SHIF}$, which includes concrete fuzzy concepts ($\mathcal{ALC}$) augmented with transitive roles, a role hierarchy, inverse, reflexive, symmetric roles, functional roles, and explicit definition of fuzzy sets. We believe that letting end-users express domain-specific knowledge by allowing imprecise terms, can bring technology closer to them and thus, it can be better exploited.

C. Overall framework integration and implementation

Once described, in last subsections, the main components of the reactive context-aware SS architecture, we proceed to study its integration within an event based hybrid rule-based system. Figure 1 shows the structure and main processing modules of the fuzzy-crisp overall architecture, as well as the information flow.

The first module, *Rule Parser*, takes as input a Mamdani format IF-THEN rule's antecedent and extracts a set of (ontologically correspondent) RDF triples. These will be the event triple patterns to be subscribed to (for modification-awareness) when executing the equivalent subscription. The second module is called *Subscriber* and takes as input the RDF triples produced by the Mamdani Rule Parser, as well as the consequent of the rule. The consequent represents the actions to be performed every time the subscription's triple pattern is inserted, removed or updated in any of the KBs. The *Subscriber* then creates a subscription as output (either SPARQL or RDF based).

When an event notification is received, the consequent of the rule is to be updated, in both KBs, to keep consistency. In the case of having a fuzzy term in the antecedent of the rule, an explicit fuzzy query needs to be executed from the subscription handler method -*subscriptHandler* in Table III-. The types of different subscription patterns, and the correspondent *fuzzyDL* queries [3] that they origin, are shown in the mapping on Table IV. In this table, the subscription patterns containing *s*, *p*, and *o* represent fixed values for subject, predicate and object respectively, while *?* represents a wild-card entity. As for the query results, the entities returned will be of interest (for rule triggering) if their satisfiability degree is >0.

In order to test the feasibility and practicality of the proposed hybrid architecture, benchmarking over the proof of concept is required. It can be noted that, with the technology available, the current solution assumes data redundancy, as it initially requires two (crisp and fuzzy) databases, where updates need to be twofold. Accepting this current technological drawback, we can design the experiment, where the objective is to realize a viability study of the framework. The main

[3]fuzzyDL syntax available in: http://gaia.isti.cnr.it/ straccia/software/fuzzyDL/syntax.html

Subscription pattern	fuzzyDL query
(?, ?, ?)	*∀ Concept C: (all-instances? C)*
(s, ?, ?)	*If s is a Concept: (min-sat? s)* *If Individual s ∈ Concept C: (min-instance? s C)*
(?, p, ?)	*If D is p's Domain and R is p's Range; ∀ Individual d ∈ D and ∀ Individual r ∈ R: (min-related? d r p)*
(?, ?, o)	*If o is a Concept: (min-sat? o)* *If Individual o ∈ Concept C: (min-instance? o C)*
(s, p, ?)	*If R ∈ p.Range: ∀ Individual i ∈ R: (min-related? s i p)*
(?, p, o)	*If D ∈ p.Domain: ∀ Individual i ∈ D: (min-related? i o p)*
(s, ?, o)	*∀ Role r, (min-related? s o r)*
(s, p, o)	*(min-related? s o p)*

TABLE IV
MAPPING OF SUBSCRIPTION TYPES TO *fuzzyDL* QUERIES

variable factors to consider are the time for a) Reasoning, b) Querying/Updating and c) Subscription response with respect to ontology size. For c), we account the time difference between the update of the data of interest, and the time when the notification is received. Likewise, different types of hybrid ontologies must be used, containing different proportions of both fuzzy and crisp entities. For this purpose, datasets of very large number of triples, such as the provided by the *LUBM* benchmark's data generator[4], can be used. As for the test queries, three main kinds of queries are to be considered with regard to the type of rule antecedent and consequent, which can be fuzzy, crisp, or hybrid (i.e., involving triples with crisp and fuzzy entities). Rules with different order of performance needs, are to be studied. Different kinds of rule antecedent translate into different implementation of subscription. In the case of existence of a fuzzy entity, explicit polling queries are executed in the *fuzzyDL* reasoner every time this is updated. For these cases, the crisp RDF store' subscription capability is used.

As we commented before, in an *IF(x) THEN(y)* rule, *x* and *y* can contain RDF triples with only crisp, only fuzzy or hybrid terms. Due to the disparity on both KBs' capabilities and content, tasks involving different kind of rule antecedent will result having different performance. This can be due, e.g., to the fact that fuzzy rules can require more computing resources, because of the explicit continuous querying required if *manual* subscription is implemented, or due to the use of specialized semantics. However, having both crisp and fuzzy KBs can be used as an advantage for optimizing the execution time of different types of queries and datasets.

To show the equivalence among SPARQL and *fuzzyDL* queries and a subscription in *M3*, we present an example. Let us assume the user wants to add the following rule to the KB: *"IF Natalia hasStatus AwayForWeekend, THEN TurnOff-AllElectricitySwitches"*. Table III shows the expressions for the equivalent mapped queries to be executed in both crisp (SPARQL) and fuzzy (*fuzzyDL*) KBs. Note that the *fuzzyDL* expression in Table III can be formulated in different ways depending on the rule's triggering criteria that best fulfils the application's needs. Another option could be querying the min. or max. satisfiability degree of the IF condition and set the triggering of our rule when, e.g., it has a satisfiability degree of min. 0.8. This is an example on how fuzzy reasoning provides more flexible or loose querying.

We identified some technical inconvenience. The programming languages of crisp and fuzzy systems do not coincide at the moment of writing (Python and Java respectively). However, both *M3* and *fuzzyDL* are under continuous development, and a Java Knowledge Processor Interface for *M3*-Redland is expected to fully support SPARQL-based subscriptions. Therefore, a complete realization of the described experiment is part of future work.

[4]http://swat.cse.lehigh.edu/projects/lubm/

IV. DISCUSSION AND FUTURE WORK

When defining and modelling human activity, to ultimately achieve behaviour learning and recognition in Smart Spaces, its characterization is not intrinsically crisp. However, the data stream of events, i.e., the SS' input that serves as base to model these human behaviours, is crisp. Therefore, we identify a gap, where connecting the qualitative view of SSs with the quantitative approach of SSs (i.e., the machinery around SS frameworks and architectures such as RDF stores and query languages) seems evident. Joining quantitative and qualitative paradigms would be crucial for expressing behaviour and adapt the SS to the user.

More concretely, our contribution consists of the proposal of a hybrid architecture with a common interface that does not only support a quantitative view of SS, with crisp (SPARQL) queries and event-based rules, but also provides a qualitative factor that takes advantage of fuzzy reasoning's expressive power to handle imprecise knowledge and rules, i.e., queries with imprecise expressions or with higher complexity, abstraction or semantic levels. Such an integrated framework can be applied in a wide range of domains, from monitoring or automating activities in assisted living or e-health, to home automation or industry processes.

An alternative to our proposed hybrid system could be implementing subscriptions within the fuzzy reasoner itself, as well as supporting SPARQL fuzzy querying by providing a crisp-to-fuzzy mapping. Then, issues on maintaining crisp or fuzzy semantics arise. This option would suppose extra engineering. Thus, we proposed a first basic approach, that keeps both architectures, to provide benefits from both crisp and fuzzy paradigms. In this way, the system is optimized to avoid continuous querying for changes when it is not needed (i.e., when rule conditions are fully crisp and, in some cases, hybrid). This implementation avoids computationally expensive approaches such as continuous polling/querying, or fuzzy discretization-based solutions such as the one that *DeLorean* [5] employs. The latter results on an exponential growth of data. However, rules with hybrid antecedent (i.e., with crisp and fuzzy terms) can be generalized by setting their subscription condition to a set of *semantically wider* crisp entities, reducing in this way, the number of explicit queries to the fuzzy reasoner, any time there is a change. For instance, hybrid antecedent conditions represented by RDF triples such as *(Natalia, isVeryNearTo, JohansOffice)* can be mapped to a semantically more general subscription, formed by a crisp-only pattern: *(Natalia, isNearTo, JohansOffice)*. Additionally, there can be cases where strict semantics are to be preserved, but we want to balance it with query efficiency. In this case, the condition *(WeatherSituationTurku, isCurrently, VeryStormy*) can be mapped to create a subscription for *(WeatherSituationTurku, isCurrently, ?)*. Therefore, this hybrid architecture' strategy allows for loosening of either semantics or efficiency, depending on our needs.

In the future, scalability and performance of the proposed architecture will be studied, as well as possible alternatives against data redundancy (due to having dual crisp and fuzzy KBs), apart from managing consistency (e.g., double update synchronization) in the joint KB. The proposed architecture, the crisp to fuzzy- language extension and the support for fuzzy reasoning show the path for dealing with current issues on SSs' usability as well as for setting the base for precise, and at the same time flexible, personalized and adaptive Smart Spaces.

ACKNOWLEDGMENTS

The research work presented in this paper is funded by TUCS (Turku Centre for Computer Science) and Hans Bang Foundation.

REFERENCES

[1] OWLIM-SE. Semantic repository for RDF(S) and OWL. owlim-se (standard edition) v4.3. [online]: http://owlim.ontotext.com/display/owlimv43/owlim-se.

[2] G. Acampora and V. Loia. Fuzzy control interoperability and scalability for adaptive domotic framework. *IEEE Transactions on Industrial Informatics*, 1:97 – 111, 2005.

[3] F. Baader. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[4] F. Bobillo. *Managing Vagueness in Ontologies*. PhD thesis, University of Granada (Spain), 2008.

[5] F. Bobillo, M. Delgado, and J. Gómez-Romero. DeLorean: A reasoner for fuzzy OWL 2. *Expert Systems with Applications*, 39(1):258 – 272, 2012.

[6] F. Bobillo and U. Straccia. *fuzzyDL*: An expressive fuzzy description logic reasoner. In *2008 International Conference on Fuzzy Systems (FUZZ-08)*, pages 923–930. IEEE Computer Society, 2008.

[7] F. Bobillo and U. Straccia. Fuzzy ontology representation using OWL 2. *Int. J. Approx. Reasoning*, 52(7):1073–1094, Oct. 2011.

[8] H. Chen, T. Finin, and A. Joshi. Semantic web in a pervasive context-aware architecture. In *Artificial Intelligence in Mobile System (AIMS 2003), In conjunction with UBICOMP*, pages 33–40, 2003.

[9] L. L. Chen and J. Biswas. Tutorial: an introduction to ontology-based activity recognition. Proceedings of the 7th International Conference on Mobile Computing and Multimedia (MoMM2009). 2009.

[10] J. Cheng, Z. Ma, and L. Yan. f-SPARQL: A flexible extension of SPARQL. In P. Bringas, A. Hameurlain, and G. Quirchmayr, editors, *Database and Expert Systems Applications*, volume 6261 of *Lecture Notes in Computer Science*, pages 487–494. Springer Berlin Heidelberg, 2010.

[11] T. R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[12] N. Guarino, D. Oberle, and S. Staab. What is an ontology? In S. Staab and R. Studer, editors, *Handbook on Ontologies*, volume XIX of *International Handbooks on Information Systems*. Springer, 2009.

[13] V. Haarslev, H.-I. Pai, and N. Shiri. Optimizing tableau reasoning in ALC extended with uncertainty. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June*, 2007.

[14] H. Habiballa. Resolution strategies for fuzzy description logic. In M. Stepnicka, V. Novák, and U. Bodenhofer, editors, *EUSFLAT Conf. (2)*, pages 27–36. Universitas Ostraviensis, 2007.

[15] J. Y. Hailong Wang, Z. M. Ma. FRESG: A kind of fuzzy description logic reasoner. Lecture Notes in Computer Science / Information Systems and Applications, incl. Internet/Web, and HCI, pages 443–450. Database and Expert Systems Applications: 20th International Conference, DEXA 2009, Linz, Austria, August 31 - September 4, 2009, Proceedings, 2009.

[16] A. G. Hernández and M. N. M. García. A JavaScript RDF store and application library for linked data client applications. In *Proceedings of the 21st International World Wide Web Conference, WWW2012*, 2012.

[17] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö. Smart-M3 interoperability platform release. http://sourceforge.net/projects/smart-m3/, October 2009.

[18] V. Jakkula, A. Crandall, and D. Cook. Enhancing anomaly detection using temporal pattern discovery. In A. D. Kameas, V. Callagan, H. Hagras, M. Weber, and W. Minker, editors, *Advanced Intelligent Environments*, pages 175–194. Springer Berlin Heidelberg, 2009.

[19] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):48 –53, jan.-march 2010.

[20] J. S. Kim, B. O. Kim, and K. S. Park. Development of HIHM (Home Integrated Health Monitor) for Ubiquitous Home Healthcare. In *Engineering in Medicine and Biology Society. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 363 –365, aug. 2007.

[21] S. Konstantopoulos and G. Apostolikas. Fuzzy-DL reasoning over unknown fuzzy degrees. In *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems - Volume Part II*, OTM'07, pages 1312–1318, Berlin, Heidelberg, 2007. Springer-Verlag.

[22] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks. BSN 2006.*, pages 4 pp. –116, april 2006.

[23] M. Murth and E. Kühn. A heuristics framework for semantic subscription processing. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 96–110. Springer Berlin / Heidelberg, 2009.

[24] U. Naeem and J. Bigham. A Comparison of Two Hidden Markov Approaches to Task Identification in the Home Environment. In *2nd International Conference on Pervasive Computing and Applications. ICPCA 2007.*, pages 383 –388, july 2007.

[25] H. Paulheim and J. Z. Pan. Why the semantic web should become more imprecise. In *What will the Semantic Web look like 10 years from now?, Boston*, 2012.

[26] D. Riboni and C. Bettini. OWL 2 modeling and reasoning with complex human activities. *Pervasive Mob. Comput.*, 7(3):379–395, June 2011.

[27] M. Ros, M. Cuéllar, M. Delgado, and A. Vila. Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows. *Information Sciences*, 220(0):86 – 101, 2013. Online Fuzzy Machine Learning and Data Mining.

[28] M. Ros, M. Pegalajar, M. Delgado, A. Vila, D. Anderson, J. Keller, and M. Popescu. Linguistic summarization of long-term trends for understanding change in human behavior. In *IEEE International Conference on Fuzzy Systems (FUZZ), 2011*, pages 2080 –2087, june 2011.

[29] M. Saleemi, N. Díaz Rodríguez, J. Lilius, and I. Porres. A Framework for Context-aware Applications for Smart Spaces. In S. Balandin, Y. Koucheryavi, and H. Hu, editors, *ruSMART 2011 : The 4th Conference on Smart Spaces, 2011.*, pages 14–25, St. Petersburg, 2011. LNCS.

[30] N. Simou, G. Stoilos, and G. B. Stamou. Storing and querying fuzzy knowledge in the semantic web using FiRE. In *URSW (LNCS Vol.)*, pages 158–176, 2013.

[31] G. Stoilos, N. Simou, G. Stamou, and S. Kollias. Uncertainty and the semantic web. *Intelligent Systems, IEEE*, 21(5):84 –87, sept.-oct. 2006.

[32] G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, and I. Horrocks. The fuzzy description logic f-SHIN. International Workshop on Uncertainty Reasoning For the Semantic Web (2005)., 2005.

[33] E. Thomas, J. Z. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *the Proc. of the Extended Semantic Web Conference (ESWC2010)*, 2010.

[34] T. van Kasteren and B. Krose. Bayesian activity recognition in residence for elders. *IET Conference Publications*, 2007(CP531):209–212, 2007.