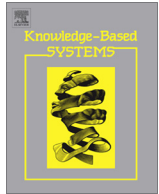




Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A fuzzy ontology for semantic modelling and recognition of human behaviour

Natalia Díaz Rodríguez^{a,1}, Manuel P. Cuéllar^{b,*}, Johan Lilius^a, Miguel Delgado Calvo-Flores^b^a Åbo Akademi University, Department of Information Technologies, Turku Centre for Computer Science (TUCS), Joukahainengatan, 3-5, FIN-20520 Turku, Finland^b University of Granada, Department of Computer Science and Artificial Intelligence, E.T.S.I. Informática y de Telecomunicación, C/ Poeta Daniel Saucedo Aranda s.n., Granada 18071, Spain

ARTICLE INFO

Article history:

Received 10 July 2013

Received in revised form 7 April 2014

Accepted 10 April 2014

Available online xxxx

Keywords:

Human behaviour

Human activity recognition

Fuzzy ontology

Context-awareness

Ambient intelligence

Semantic Web

ABSTRACT

We propose a fuzzy ontology for human activity representation, which allows us to model and reason about vague, incomplete, and uncertain knowledge. Some relevant subdomains found to be missing in previous proposed ontologies for this domain were modelled as well. The resulting fuzzy OWL 2 ontology is able to model uncertain knowledge and represent temporal relationships between activities using an underlying fuzzy state machine representation. We provide a proof of concept of the approach in work scenarios such as the office domain, and also make experiments to emphasize the benefits of our approach with respect to crisp ontologies. As a result, we demonstrate that the inclusion of fuzzy concepts and relations in the ontology provide benefits during the recognition process with respect to crisp approaches.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Human activity study is a complex but key aspect in the development of Ambient Intelligence (Aml) systems. Different techniques have been developed for activity and user modelling, and they may be classified as data-driven [1] and knowledge-based [2] approaches. The methods in the first category are aimed at providing robust models for handling human behaviour specific features using statistical and machine learning techniques. The best strengths of these models are their ability to handle noise, uncertainty, or incomplete sensor data [3], and they have proven to be accurate in different domains where semantics are not key. However, the need for training data and the time and performance required for these models are limitations in dynamic environments and situations where context-aware data prevail. Furthermore, data-driven algorithms do not offer abstract reasoning mechanisms that allow inferring the meaning of the actions according to their semantics [4].

On the other hand, knowledge-based techniques have been applied in pervasive computing environments to improve interoperability and adaptation to different context situations. Usually, context data sources are dynamic, continuously changing depending on the environment, not always mobile, known, nor taken into account in advance. For this reason, these methods show advantages with respect to data-driven models due to the inclusion of context management tools. Further features of knowledge-based techniques that are interesting for human activity representation are the possibility of providing both the environment and the user with semantics to aid in the context definition process, facilitate the definition and comprehension of human behaviours (e.g. machine readable and easier to interpret), and consequently, ease the development of new learning and recognition models able to better understand the meaning of human actions and execute logic reasoning about future needs, situations, or actions. In addition, all this can occur considering the context information where the activity is performed. Examples of knowledge-based techniques contain logic-based approaches [5,6], rule-based systems [7], and ontological models [8].

Despite the fact that most of the approaches about human activity recognition are focused in Ambient Assisted Living and Smart Home assistance [2,9,10], another emerging scenario is the office/work domain and public buildings environments. In this case, the goals are aimed at improving energy efficiency and work assistance [11–14]. For instance, *MOSES* [15] localizes work staff and identifies the tasks they are doing at any moment, being able in this way to give advice on the remaining tasks to be carried

erability and adaptation to different context situations. Usually, context data sources are dynamic, continuously changing depending on the environment, not always mobile, known, nor taken into account in advance. For this reason, these methods show advantages with respect to data-driven models due to the inclusion of context management tools. Further features of knowledge-based techniques that are interesting for human activity representation are the possibility of providing both the environment and the user with semantics to aid in the context definition process, facilitate the definition and comprehension of human behaviours (e.g. machine readable and easier to interpret), and consequently, ease the development of new learning and recognition models able to better understand the meaning of human actions and execute logic reasoning about future needs, situations, or actions. In addition, all this can occur considering the context information where the activity is performed. Examples of knowledge-based techniques contain logic-based approaches [5,6], rule-based systems [7], and ontological models [8].

* Corresponding author. Tel.: +358 22154764.

E-mail addresses: ndiaz@abo.fi (N. Díaz Rodríguez), manupc@decsai.ugr.es (M.P. Cuéllar), johan.lilius@abo.fi (J. Lilius), mdelgado@decsai.ugr.es (M. Delgado Calvo-Flores).¹ Principal corresponding author. Tel.: +34 956526159.

out and warn about potential oversights or forgotten actions. Another example is *iShopFloor* [16], a multi-agent architecture to plan and control industry processes. However, semantic technologies have not been generally included into these models, although there are exceptions such as in intelligent meeting rooms [17] or maintenance of large buildings [16,18]. Aml scenarios in offices or work environments focus on easing the work in groups and optimizing the office space. For example, *EasyMeeting* [19] is an intelligent meeting room system that builds on the design of *CoBrA* [20]. RFID sensors embedded in the walls and furniture detect the presence of the users' devices and clothing. On receiving information about the user's context and intention, the broker sharing platform allows the activation of the projector, slide downloading, and lighting control. In [18], ontology-based interoperability is applied to Smart Spaces for a context-aware maintenance of large buildings, monitoring environmental variables, automatically detecting building-related faults, and executing multi-modal interventions. An architecture for an ubiquitous group decision support system, *WebMeeting* [21], is able to consider the emotional factors of participants and their associated argumentation processes. The system shows available information to the participants, analyses the meeting trends, and suggests arguments to be exchanged with others. Further interesting projects regarding activity recognition in the office domain are the AIRE project [12], the *SmartOffice* or *Monica* project [14], the Interactive Room (*iRoom*) [22], or the NIST Smart Space and Meeting Recognition projects, which develop tools for assistance in meeting rooms [13].

The most widely used tool to integrate semantics into activity recognition systems are ontologies [4]. However, there are current limitations of ontology-based activity recognition techniques that must be tackled: they require good knowledge engineering skills to model the domain, OWL DL does not allow interval (i.e. overlapping) temporal reasoning, ontological reasoning can be computationally expensive [23], and they cannot deal with uncertainty [4]. In this work, we provide advances to solve this last limitation and propose a fuzzy ontology to give support for imprecision and uncertainty, typical of everyday life situations. For instance, a sensor can give readings with a certain degree of reliability, or work only at specific times or in certain conditions; users may perform subtle changes in the way they perform their activities, the execution of an activity may be detected with a certainty or satisfiability degree, and all this information should be taken into account into the reasoning process. Unfortunately, classic crisp ontology proposals cannot handle this type of information. In our approach, different levels of granularity are designed so that incremental context acquisition allows behaviour abstraction and a more accurate, i.e., low-level recognition. By setting a behaviour specification structure, a set of rules can define how to recognize a human behaviour out of a sequence of observations. And, since fuzzy ontologies can handle uncertainty, our approach is able to solve this limitation with respect to crisp approaches. Fuzzy logic was already proposed as an argument to “reject the maximality rule, according to which only altogether true sentences are true, and embracing instead the rule of endorsement, which means that whatever is more or less true is true” [24]. As argued in [24], positing fuzzy predicates usually simplifies theories in most scientific fields; fuzzy predicates are much more plausible and give a more cohesive world view than their crisp counterpart. In this way, classical ontologies are not suitable to deal with imprecise and vague knowledge, which is inherent to real world domains [25]. On the other hand, fuzzy ontologies have the advantage of extending information queries, allowing the search to also cover related results. This makes the decisions about relatedness based on modelled domain knowledge, i.e., instead of just offering exact matches, the search can be extended to cover also related concepts, so that precise wording is not needed to get a useful hit (as the

context of a document does not have to be exactly the same one for the user to benefit from it) [26]. This results on more effective retrieval. Likewise, another advantage of fuzzy ontologies is the fuzzy semantics, as they are more flexible towards mapping between different ontologies [26].

Let us put an example to show the benefits of fuzzy ontologies versus crisp ones. Because in a fuzzy ontology we can define that the *CoffeeBreak* activity is recognized accounting for different weights on the actions that compose it (e.g. 0.3 *TakeMug*, 0.3 *TakeCoffeePan*, 0.4 *TakeMilk*), thus, when one action has been skipped due to an exception (e.g. milk run out) or a missing sensor reading, the activity can still be recognized to a lower degree. In contrast, the same activity formalized in a crisp ontology could not be recognized if any of the exclusive elements that compose it is missing.

The rest of the paper is organized as follows: the following section describes related work on ontologies for human activity recognition and introduces fuzzy ontologies as the main tool for the rest of the manuscript. After that, in Section 3, we present a novel ontology for human activity modelling and its extension to Fuzzy OWL 2 with support for the *fuzzyDL* reasoner. We detail concepts and relationships in the fuzzy ontology as well as Section 4.1 presents the use case on domain specific entities for the office environment. Section 4.2 describes an evaluation of the approach with respect to the crisp case, and finally, conclusions and future work are shown in Section 5.

2. Related work

2.1. Ontologies for human behaviour recognition

The literature offers a wide range of ontology definitions [27], although the most extended one is that an ontology is a “formal specification of a shared conceptualization” [28]. It offers a formalism to represent classes or concepts, individuals, relations, functions, and attributes. As providers of a format for exchanging knowledge, they promote interoperability, knowledge reuse, and information integration with automatic validation. Ontologies separate declarative and procedural knowledge, making the modularity of the knowledge base (KB) [29] easier. They also allow information to become not only human but also machine-readable by agents. Ontologies have been used in heterogeneous problems such as intelligent m-Government emergency response services (e.g., disasters and attacks) through case-based reasoning [30] or detecting information system conflicts in requirement analysis phase [31], just to name a few. Using ontologies in human activity recognition provides a number of advantages [32]: it supports incremental progressive activity recognition, state based modelling, and a robust reasoning mechanism. Other benefits are the ability to discriminate the significance and urgency of activities through semantic descriptions, and the support for course-grained and fine-grained activity assistance and the possibility for data fusion and semantic reasoning, including activity recognition, activity learning, and activity assistance.

In order to model human activity and behaviour in Aml, the context needs to be modelled. With respect to other context models such as key-value models, object oriented or logic based models [33], ontology-based context modelling excels as regards simplicity, flexibility, extensibility, generality, expressiveness, and automatic code generation [34]. Approaches based on ontology reasoning [2,3] represent activities and each data that can be used to recognize them, from sensors to actors. There are also hybrid approaches combining data-driven and knowledge-based approaches for activity recognition, e.g., evidential network-based activity inference [10] or COSAR [35]. The COSAR system retrieves information about simple human activities using hybrid ontological/statistical reasoners.

Ontological reasoning with OWL 2 serves to recognize complex activities based on elementary sensor data and simple activities recognized through statistical reasoning [8]. In COSAR, the PalSPOT ontology [23] is used within the context aggregation middleware CARE [36]. Another hybrid approach incorporates ontology-based context reasoning together with computer vision research and integrates a scene tracking system with an ontological layer to avoid issues that make classical object tracking procedures fail in complex scenarios [37].

There is a set of ontologies for modelling context in human activity. CoBrA-Ont (from Context Broker Architecture) [20] is an extension from SOUPA. It particularly handles locations, people, and activities. The CoDAMoS [38] ontology focuses on modelling roles, hardware, and software services around four main core entities: user, environment, platform, and service. Delivery Context [39] is another ontology, from W3C, that treats hardware and software interfaces and networks. SOUPA [40] (Standard Ontologies for Ubiquitous and Pervasive Applications) concerns rely on time, agents, actions, locations, and policies. The mIO! network ontology [41] applies the NeOn re-engineering methodology to reuse other ontologies and stress different concepts around services and device interaction. CONON [42] (CONtext ONtology) provides an upper ontology with environmental conditions as well as diverse extensions such as activity planning (deduced and scheduled activities) and services. In PiVOn [43] (Pervasive Information Visualization Ontology), location models and device-based services can be highlighted as one of their focus, while the Situation Ontology [44] mainly considers the modelling of situation and context layers, as well as hardware and software interfaces and networks. A more in-depth survey of ontologies for human behaviour representation may be found in [4].

The common lacking element found in existing ontologies is the support for modelling uncertain, vague, and imprecise information [4], which is an inherent feature of activity recognition. For this reason, in the next sections, we propose a solution to this problem based on fuzzy ontologies.

2.2. Fuzzy Ontologies

In many scenarios, and particularly in the human behaviour representation domain, we find elements whose nature is imprecise. A classic crisp ontology cannot represent this type of information, since they can only model relations between entities that may be either true or false. For instance, in the statement “User has-Event Event planned in Location L at Time T ”, T does not have to be exact in practice. In another example, “User isPerforming CoffeeBreak”, the activity *CoffeeBreak* could be recognized with some degree of truth depending on the sensor data acquired and how the user is performing the activity. Fortunately, fuzzy and possibilistic logic have proved to be suitable formalisms to handle imprecise/vague and uncertain knowledge, respectively [29]. Contrary to classical set theory, where elements either belong to a set or not, in the fuzzy set theory, elements can belong to a set with some degree. Formally, a fuzzy subset A of X is defined by a membership function $\mu_A(x)$, or simply $A(x)$, which assigns any $x \in X$ to a value in the real interval between 0 and 1. Fuzzy logic allows to perform approximate reasoning involving inference rules with premises, consequences, or both of them containing fuzzy propositions [45].

Up to 17 formal definitions can be found for fuzzy ontology [29]. However, one of the most accepted definitions is *an ontology that uses fuzzy logic to provide a natural representation of imprecise and vague knowledge, and eases reasoning over it*. Fuzzy Description Logic (DL) is the most developed formalism to work with fuzzy ontologies [29]. Formally, a Fuzzy Knowledge Base (FKB) or fuzzy ontology can be considered as a finite set of axioms that comprises a fuzzy ABox A and a fuzzy TBox T [46]. A fuzzy ABox consists of a

finite set of fuzzy (concept or role) assertions, while a fuzzy TBox consists of a finite set of fuzzy General Concept Inclusions (Fuzzy GCIs), with a minimum fuzzy degree of subsumption. Fuzzy ontologies and fuzzy extensions of DL have shown to be useful in applications from information retrieval and image interpretation to Semantic Web and others [46]. In [47], a fuzzy keyword ontology serves to annotate and search events in reports by superimposing a fuzzy partonomy² on fuzzy classifications.

Given a crisp ontology, elements that can be fuzzified include datatypes, object properties (through fuzzy modifiers), and data properties (through fuzzy modified data types). Furthermore, depending on the application domain, it can be frequent to have assertions of axioms about concrete individuals and classes, as well as (data and object) property assertions with a fuzzy degree. Among different approaches for fuzzifying an ontology, we can find Fuzzy OWL 2 mappings [45] or value discretization approaches [48]. To fuzzify a crisp ontology, it needs to be translated into a language supported by a fuzzy ontology reasoner. Fuzzy OWL 2 parsers convert Fuzzy OWL 2 ontologies into *DeLorean* [29] and *fuzzyDL* [45] reasoners' syntax. *DeLorean*³ [29] is a fuzzy rough DL reasoner that supports fuzzy rough extensions of the fuzzy DLs *SROIQ(D)* and *SHOIN(D)* (equivalent to OWL and OWL 2) and it is based on a discretization of the fuzzy ontology using α -cuts [29]. *DeLorean* computes an equivalent non-fuzzy representation in OWL or OWL 2. However, we consider *fuzzyDL* to be the most convenient existing tool for ontological reasoning with uncertainty. Next, we motivate our needs and its usage.

2.2.1. fuzzyDL reasoner

*fuzzyDL*⁴ [45] reasoner's main features are the extension of the classical description logic *SHIF(D)* to the fuzzy case. It allows fuzzy concepts with left-shoulder, right-shoulder, triangular, and trapezoidal membership functions, general inclusion axioms and concept modifiers. Fuzzy modifiers apply to fuzzy sets to change their membership function. *FuzzyDL* supports crisp intervals that can serve to define fuzzy concrete predicates. In fuzzy rule based systems (e.g. Mamdani IF-THEN system), fuzzy IF-THEN rules are fired to a degree which is a function of the degree of match between their antecedent and the input. The deduction rule is generalized Modus Ponens. *FuzzyDL*'s reasoning algorithm [45] uses a combination of a tableau algorithm and a MILP (Mixed Integer Linear Programming) optimization problem.

By adopting fuzzy reasoners, we can accept axioms (of activities or behaviours) happening with a certain degree of truth, completeness or certainty. In *fuzzyDL*, the notion of satisfaction of a fuzzy axiom E by a fuzzy interpretation $\mathcal{I}$, denoted $\mathcal{I} \models E$, is defined in [45] as follows:

- $\mathcal{I} \models \langle \tau \geq \alpha \rangle$ iff $\tau^{\mathcal{I}} \geq \alpha$,
- $\mathcal{I} \models (\text{trans } R)$ iff $\forall x, y \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, y) \geq \sup_{z \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, z) \otimes R^{\mathcal{I}}(z, y)$,
- $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff $\forall x, y \in \Delta^{\mathcal{I}}, R_1^{\mathcal{I}}(x, y) \leq R_2^{\mathcal{I}}(x, y)$,
- $\mathcal{I} \models (\text{inv } R_1 R_2)$ iff $\forall x, y \in \Delta^{\mathcal{I}}, R_1^{\mathcal{I}}(x, y) = R_2^{\mathcal{I}}(y, x)$.

In *fuzzyDL*, concept C is satisfiable iff there is an interpretation $\mathcal{I}$ and an individual $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) > 0$ [45]. For a set of axioms $\mathcal{E}$, we say that $\mathcal{I}$ satisfies $\mathcal{E}$ iff $\mathcal{I}$ satisfies each element in $\mathcal{E}$. $\mathcal{I}$ is a model of E (resp. $\mathcal{E}$) iff $\mathcal{I} \models E$ (resp. $\mathcal{I} \models \mathcal{E}$). $\mathcal{I}$ satisfies (is a model of) a fuzzy KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, denoted $\mathcal{I} \models \mathcal{K}$, iff $\mathcal{I}$ is a model of each component $\mathcal{A}$, $\mathcal{T}$ and $\mathcal{R}$, respectively.

An axiom E is a logical consequence of a knowledge base $\mathcal{K}$, denoted $\mathcal{K} \models E$ iff every model of $\mathcal{K}$ satisfies E . Given $\mathcal{K}$ and a fuzzy

² A fuzzy partonomy is a decomposition that fuzzifies the classical whole-part relationship.

³ *DeLorean* Reasoner: <http://webdiis.unizar.es/fbobillo/delorean>.

⁴ *fuzzyDL* Reasoner: <http://gaia.isti.cnr.it/straccia/software/fuzzyDL/fuzzyDL.html>.

axiom τ of the forms $\langle x : C, \alpha \rangle$, $\langle \langle x, y \rangle : R, \alpha \rangle$ or $\langle C \sqsubseteq D, \alpha \rangle$, it is of interest to compute τ 's best lower degree value bound.

The *greatest lower bound* of τ w.r.t. $\mathcal{K}$ (denoted $glb(\mathcal{K}, \tau)$) is $glb(\mathcal{K}, \tau) = \sup \{n | \mathcal{K} \models \langle \tau \geq n \rangle\}$, where $\sup \emptyset = 0$. Determining the *glb*, i.e., the Best Degree Bound (BDB) problem, consists of determining the best satisfiability bound of a concept C :

$$glb(\mathcal{K}, C) = \sup_{\mathcal{I}} \sup_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) | \mathcal{I} \models \mathcal{K}\}.$$

Reasoning tasks allowed by *fuzzyDL* are typical BDB, concept satisfiability and subsumption problems, optimization of variables and defuzzifications.

In the next section, we detail the components of our proposal on a crisp ontology for human behaviour modelling and how its fuzzification is performed by using *fuzzyDL*.

3. A fuzzy ontology for human activity modelling

In this section, we first describe the design principles used to represent human behaviour in a crisp subset/core of the ontology and then, the fuzzification procedure to obtain the proposed fuzzy approach. Among the different knowledge engineering methodologies, we mostly followed NeOn Ontology engineering methodology [49] to make our approach. Some highlighted aspects of this methodology are the ontology resource reuse, requirements specification, development of required scenarios, and dynamic ontology evolution. NeOn targets software developers and ontology practitioners, it has dynamic guidelines for ontology evolution and treats context dimension and distributed collaboration. As human activity recognition requires constant update of input sensor data and the ability to adapt to changes in behaviour, NeOn allows the evolution of the ontology.

3.1. A (crisp) human behaviour ontology

First we modelled a basic (crisp) ontology that gathers concepts from different basic subdomains. We can distinguish among four main entities: Users, Environment, Activities, and the corresponding interactions among them. The three first elements can be considered the core of the ontology, and all together define the *what*, *who*, *when*, and *where* for each relevant event to be annotated:

1. *Users*. Users can be divided into two categories: A *Single User* is used to represent a unique user performing activities. Aspects to consider when learning behaviours contain the user's location, role/position, calendar, the user social network, etc. On the other hand, the second category *Multi-user/Generic User* involves a group of users sharing a common behaviour or objective (meeting attendants, visitors, students, etc.). Since this kind of behaviour embraces a group of people performing the same action, they are not considered individually. We define a *Generic User* class as a way to represent an abstraction of a group of users. The *Generic User* shares some properties with *Single User* actions and activities, such as roles or access rights. This class is also useful when personal data about users is not known, but is relevant as an observed entity in the environment.
2. *Environment*. It is an organized hierarchy of locations, models, and generic and specific features of each kind of space. For example, in the office domain, environments are offices, meeting and lecture rooms, auditoriums, kitchen, toilets, etc. Different levels to track indoors positioning may include floor and room numbers, while outdoor locations may refer to open spaces or means of transport. A location can have associated measurements such as *Humidity*, *Temperature*, *Lighting*, *Noise-Level*, or *Pressure*. For a finer grained environment, objects

(e.g. doors, curtains, windows) can have, e.g., an *Aperture* state and rooms and locations a (seating) *Capacity*.

3. *Actions, Activities, and Behaviours*. We distinguish among three types of events or activity granularity levels.

- *Actions* or atomic events with a *timestamp*, e.g.: *OpenDoor*, *MoveObject*, *TurnLightOff*, *WalkBy*, *BeObservedInLocation*, etc. This is the lowest granularity degree of representation.
- *Activities*, considered as single actions with an inherent “purpose”, or composed by a set of different actions. An activity represents an intermediate granularity level of representation and has a *startDatetime* and *endDatetime*. E.g.: *TakeCoffee*, *AttendConference*, *GroupMeeting*, *VideoCall*, *SendEmail*, etc. An activity is defined by a set of compulsory actions plus a set of optional actions, where some of them can have temporal execution interdependencies, e.g. *MakeCoffee*, *SpeakAtConference*, *VisitLocation*, etc.
- *Behaviours*, a sequence of activities and/or actions. A behaviour is defined by a set of compulsory actions or activities plus a set of optional actions or activities, where some of them can have temporal execution interdependencies. E.g., the behaviour *CoffeeBreak* includes the Action *ExitOffice*, the Activity *MakeCoffee* or *TakeCoffee*, and the Action *EnterOffice* in this order. The difference between activity and behaviour is that an activity is always the same regardless the context, while a behaviour is always defined and valid within a concrete context. For instance the activity *Running* is always defined in the same way. However, if *Running* has a specific meaning or goal within a context, then it can be (part of) a specific behaviour.

Any user can perform (atomic) actions, activities and/or behaviours. These are structured in a hierarchy of ascending abstraction. Depending on the actors in the environment, they will be individual or collective (social) activities. It is important to mention the possibility of *recursion* in the definition of activities and behaviours: both can be defined semantically (through the *involves* property) as a set of actions and (sub) activities. For instance, the activity *going to bed* is composed by the activities *having infusion* and *taking pills*, and the actions *putting pyjama* and *brushing teeth*. This design allows us to work in different degrees of granularity and to decompose complex activity and behaviour recognition stages into simpler procedures.

4. *Relationships*. Object and data properties model interaction among users and the environment that can serve to specify behaviours. Relationships can link *Single Users*, *Generic User/Multi-users*, or users and environment elements.

An excerpt of types of users, locations, and a subset of Actions and Activities, as well as object and data properties can be seen in Figs. 1–4. In addition, Fig. 5 represents a systematic relationship among the activity, user, and environment context. Since the full proposal cannot be shown due to space limitations, we make the

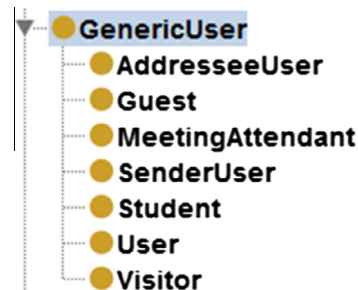


Fig. 1. Excerpt of *User* subclasses in the ontology (partial).

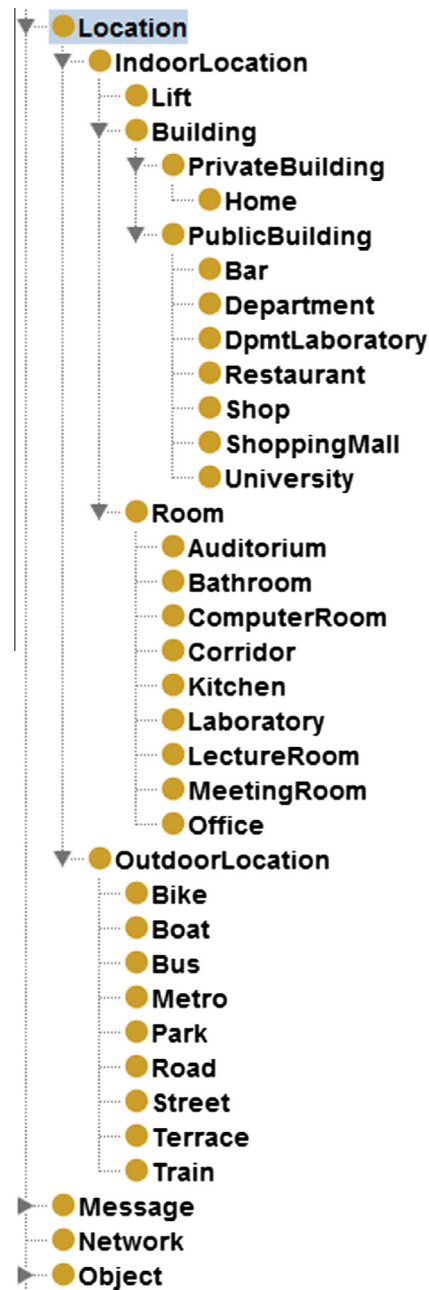


Fig. 2. Excerpt of Location subclasses in the ontology (partial).

complete ontology available through web.⁵ Our approach is focused on the office domain and reuses some hierarchy concepts from existing ontologies, e.g., the indoor/outdoor hierarchy from the CONON ontology [42] or environment features from CoDAMos ontology [38]. We also include other features of interest that we considered relevant in a human activity ontology:

- **Activity duration and concurrency:** Several activities can be performed, by the same or different people, at the same or overlapping times. Also, some events resulting from user actions, remain done for another user later, e.g., turn the lights on. The next users may not have to perform that action in order to complete the same activity as the first user. To model these situa-

tions, the Boolean data property *remainDone* of the class *Action* indicates when an action required by an activity can remain done or “active” to someone else.

- **Activity characterization and indispensable actions:** Not all users carry out a given behaviour or activity on the same way. Therefore, an activity can be performed according to more than one behaviour model, one per user. In order to know, in an efficient manner, if certain action is absolutely required to perform a given activity, the object property *isIndispensableForActivity* (*Action*, *Activity*) can be set to avoid further computations. The property *isIndispensableForBehaviour* (*Action OR Activity*, *Behaviour*) works in the same way for behaviours.
- **Messages or alerts:** Alerts are useful as reminders of forgotten actions or to warn about potentially hazardous situations. Types of messages are *Error*, *Alarm*, *Information*, or *Suggestion*; modelling the message recipients happens through *Sender* and *Address-see Device*, and *User* classes, respectively.

3.2. Fuzzification of the ontology

To represent our fuzzy entities in the created human activity ontology, we use Fuzzy OWL2 2.1.1 plug-in⁶ in Protégé 4.1, jre 1.6, that provides support in creating Fuzzy OWL 2 ontologies. The plug-in does not translate fuzzy representations into OWL 2, but rather eases their representation by allowing specification of the type of fuzzy logic used, definition of fuzzy data types, fuzzy modified concepts, weighted concepts, weighted sum concepts, fuzzy nominals, fuzzy modifiers, fuzzy modified roles and data types, and fuzzy axioms [46]. In Fuzzy OWL 2, three main alphabets of symbols are assumed: concepts (fuzzy sets of individuals), roles, and individuals [46]. These are represented in an ontology as classes, relations, and individuals, respectively. The degree of truth of a fuzzy assertion is equal to the proportion of observers who think that the crisp assertion is true [29].

Considering the designed human activity ontology, we can identify data types, concepts, properties, and relations that are susceptible of being fuzzy. A fuzzy data type $\mathbf{D}$ is a pair $\langle \Delta_{\mathbf{D}}, \Phi_{\mathbf{D}} \rangle$ where $\Delta_{\mathbf{D}}$ is a concrete interpretation domain, and $\Phi_{\mathbf{D}}$ is a set of fuzzy concrete predicates $\mathbf{d}$ with an arity n and an interpretation $\mathbf{d}^x : \Delta_{\mathbf{D}}^n \rightarrow [0, 1]$, which is an n -ary fuzzy relation over $\Delta_{\mathbf{D}}$ [46]. For fuzzy data types, the functions allowed in Fuzzy OWL 2, defined over an interval $[k_1, k_2] \subseteq \mathbb{Q}$, are $\mathbf{d} \rightarrow \{\text{left}(k_1, k_2, a, b)$ (Fig. 6c), $\text{right}(k_1, k_2, a, b)$ (Fig. 6d), $\text{triangular}(k_1, k_2, a, b, c)$ (Fig. 6b), $\text{trapezoidal}(k_1, k_2, a, b, c, d)$ (Fig. 6a), $\text{linear}(k_1, k_2, c)$ (Fig. 6e), $\text{mod}(\mathbf{d})\}$. More specifically, the fuzzification of each element in the ontology is done as follows:

• Fuzzy data types and fuzzy concrete roles (data properties).

Data properties in the original ontology can be transformed into fuzzy data types. Their range is expressed through data range expressions such as $\text{double}[\geq -100.0]$ and $\text{double}[\leq 100.0]$ (e.g. for *hasTemperature* data property range). We use as range the referential set over which the fuzzy membership functions associated are defined. On the other hand, fuzzy concrete role is defined in Fuzzy OWL 2 by setting its range data type to a previously defined fuzzy data type. Examples of membership functions used are:

- *LowTemperature*: fuzzy data type with left shoulder membership function ($a = -5$, $b = 5$).
- *MediumTemperature*: fuzzy data type with trapezoidal membership function ($a = 5$, $b = 10$, $c = 20$, $d = 25$).
- *HighTemperature*: fuzzy data type with right shoulder membership function ($a = 25$, $b = 30$).

⁵ Fuzzy Human Behaviour Ontology and experiments: <http://users.abo.fi/ndiaz/public/FuzzyHumanBehaviourOntology/>.

⁶ <http://nemis.isti.cnr.it/straccia/software/FuzzyOWL/#plug-in>.



Fig. 3. Excerpt of classes, data and object properties in the ontology (partial).

An annotation example in OWL 2 for the data type *highTemperature* is as follows (see Listing 1):

When creating a fuzzy role, an annotation property describing the type of the constructor and the value of its parameters are specified. Recursion is not allowed in the definition, and following the mapping in [46], only fuzzy modified roles are supported. The domain of the annotation will be any OWL 2 (object or data) property with the restriction that the modifier must be defined as a fuzzy modifier and that the base fuzzy role has a different name than the annotated role. Examples of fuzzy data types defined in our ontology are:

- *ShortDuration*, *MediumDuration*, *LongDuration* are fuzzy data types (FDT) used to represent duration (in seconds). The concrete role (CR) *Activity.hasDuration* indicates the duration of an *Activity*.
- The FDT *LowVolume*, *MediumVolume*, *HighVolume* are used to represent audio volume level in dB. The CR *Device.hasVolume* indicates the volume of a *Device* with audio capability (computer, radio, TV, etc.).

- The FDT *ClosedAperture*, *HalfAperture*, *OpenAperture* describe aperture angle in degrees of the CR {*Door*, *Window*, *Curtain*, etc.}.*hasAperture*.
- *SmallCapacity*, *MediumCapacity*, *LargeCapacity* are FDT representing the amount of people, and they are used in CR of *Location.hasCapacity* to describe the capacity of a *Location*.
- The FDT *LowHumidity*, *MediumHumidity*, *HighHumidity* are used to model humidity (in g/m^3) in the CR *Location.hasHumidity* of a *Location*.
- *LowTemperature*, *MediumTemperature*, *HighTemperature* are FDT to measure temperature in centigrade degrees, and they are applied over CR {*Location*, *Room*, *Environment*, *GenericUser*, *Object*}.*hasTemperature* of a *Location*, *Environment*, *User*, or *Object*.
- *LowLighting*, *MediumLighting*, *HighLighting* are FDT measuring lighting in lux, applied to CR of {*Location*, *Environment*}.*hasLighting* of a *Location* or *Environment*.
- The FDT *LowNoiseLevel*, *MediumNoiseLevel*, *HighNoiseLevel* are designed to measure noise level in dB in the CR {*Location*, *Environment*}.*hasNoiseLevel* of a *Location* or *Environment*.

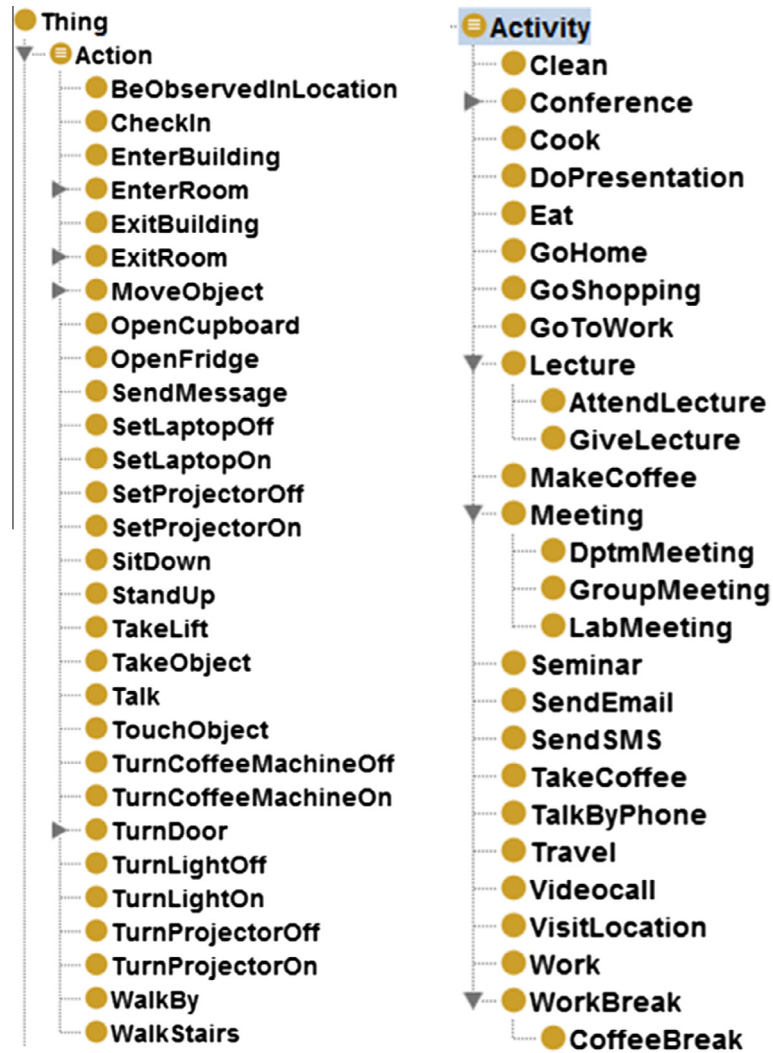


Fig. 4. Excerpt of Action and Activity classes in the ontology (partial).

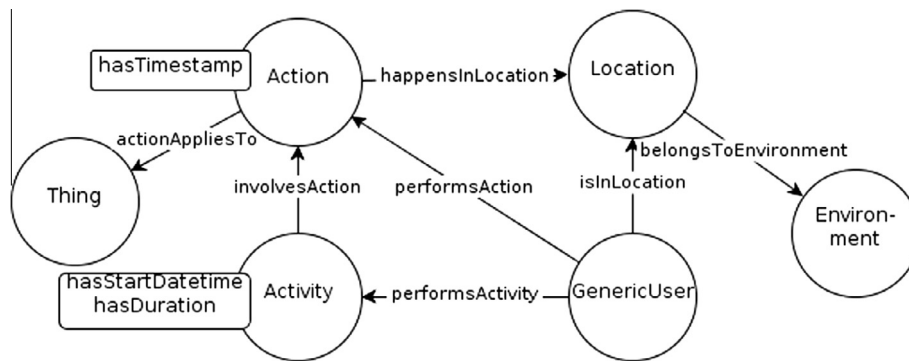


Fig. 5. Main relationship among User, Activity, and Environment in the modelled context ontology.

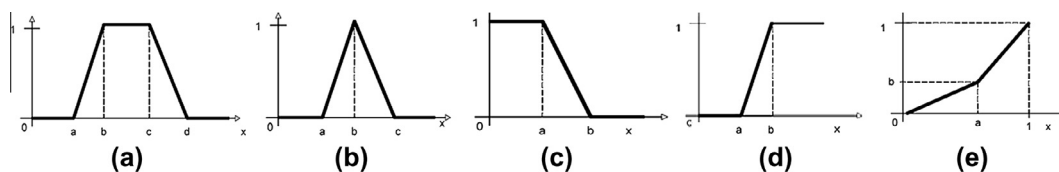


Fig. 6. Membership functions in [29] for fuzzy data type definition (i.e., fuzzy concrete domains) and fuzzy modifier functions: (a) Trapezoidal function; (b) triangular function; (c) left-shoulder function; (d) right-shoulder function; and (e) linear function.

```

1 <fuzzy0w12 fuzzyType="datatype">
2 <Datatype type="rightshoulder" a="25.0"
   b="30.0" />
3 </fuzzy0w12>

```

Listing 1. Annotation for a new fuzzy data type *highTemperature* using rightshoulder modifier.

- *LowPressure*, *MediumPressure*, *HighPressure* are FDT to measure pressure in atmospheres, and they are used in the CR of {Location, Environment}.hasPressure of a *Location* or *Environment*.
- *FewPeople*, *MediumPeople*, *ManyPeople* are integer-valued FDT to describe a number of people, in the CR of Event.hasAttendants such as *Conference*, *Seminar*, *Symposium*, *Workshop* or other *Event*.
- **Fuzzy Abstract Roles** (Fuzzy Object Properties). Object properties in the original ontology can be transformed into fuzzy abstract roles by means of the assignment of a fuzzy membership value. Examples in our ontology are:
 - *Thing-isInLocation*-Location: Represents the location of anything (a *Thing*) in a given *Location*. A fuzzy degree can represent proximity.
 - *User-attendsEvent*-Event: Identifies a given *User* who attends an *Event*. Also, *Event* can belong to a *Calendar*.
 - *Travel-toLocation*-Location: Characterizes the activity *Travel* by specifying the destination *Location* it refers to.
 - *Activity-happensInLocation*-Location: Associates an *Activity* with a *Location* to indicate where it occurs.
 - *Action-actionAppliesTo*-Thing: Indicates the object over which an *Action* or *Activity* directly falls or acts over. E.g., *WalkBy* activity applies to *Corridor* if a user walks by a corridor.
 - *GenericUser-hasPersonalStatus*-PersonalStatus: Indicates a personal status of any *User* in certain moment. *PersonalStatus* class is specialized into: *Available*, *Away*, *Busy*, *OnHoliday*, *OnLeave*.
 - *Activity-involvesAction*-Action (and *involvesActivity*, *involvesOptAction*, *involvesOptActivity*, respectively, where *Opt* stands for optional, relates an *Activity* with the *Actions* that it involves or requires. E.g., the *DoPresentation* activity involves actions *SetLaptopOn*, *SetProjectorOn*, *StandUp*, and *Talk*.
 - *GenericUser-performsAction*-Action and *GenericUser-performsActivity*-Activity specify which *User* performs an *Action* or an *Activity*, respectively. A fuzzy degree here represents the level of uncertainty about who performs the action.
 - *Thing-isNearTo*-Thing describes closeness among two entities (*Object*, *User*, *Location*, etc.).
- **Fuzzy Modifiers and Fuzzy Modified Data Types:** The degree of membership of fuzzy data types may be specialized by means of fuzzy modifiers. A fuzzy modifier is a function $f_{mod} : [0, 1] \rightarrow [0, 1]$ which applies to a fuzzy set to change its membership function, which can be *linear* (c) (Fig. 6e) or *triangular* (a, b, c) (Fig. 6b). We define the fuzzy modifiers *very* as *linear* (0.85), and *barely* as *linear* (0.15), to improve the expressiveness of the ontology. For instance, given the object property *isNearTo* (*Thing*, *Thing*), the fuzzy modifiers can be used to define new fuzzy properties such as *very* (*isNearTo*) or *barely* (*isNearTo*) (i.e., *isVeryNearTo*, or *isBarelyNearTo*), and relate objects that are not close to each other, but *very* (respectively *barely*) close.
- **Fuzzy Axioms or Assertions:** In practice, expressing degrees of truth is reflected on real time when asserting axioms. Examples of fuzzy axioms can be assertions such as the following example: the *User* individual *John* *isInLocation* *Office* with degree 0.7, if the system has detected usual activity of John at his office, but there are some changes in his routine).

When asserting axioms and defining data properties, linguistic labels make the editing of the rule closer to natural language and also easier for any kind of user, not requiring any technical knowledge. A degree within the interval (0,1] can, in addition, be provided for more precise assertions or axioms. For instance, to find whether there is any user which is *close enough* to the meeting room, we could assign a certainty value >0.8 to the query. Fuzzy queries will be provided in the experimental section as a proof of concept and for ontology evaluation.

3.3. Fuzzy reasoning

Our fuzzy ontology is mainly characterized by the presence of fuzzy data types, which are especially tackled by *fuzzyDL*. For this reason, we use this reasoner in our ontology. We do not specify degrees of truth initially on the ontology, these are asserted in execution time. Thus, α -cuts cannot be pre-calculated as in the *DeLoorean* reasoner. *FuzzyDL*, however, has some limitations that are analyzed in this section, as its implications in our proposal.

First of all, *fuzzyDL* does not allow representation of asymmetry and irreflexivity role axioms with the available constructors. In practice, this entails no critical implications to our work, since our roles are defined with domain and range for each property, and these are usually disjoint classes. Therefore, while it is possible to express reflexive and symmetric roles, we have to account that there will not be any asymmetry or irreflexivity that can be enforced by the reasoner. Secondly, cardinality restrictions are not implemented in *fuzzyDL*. This problem may cause 2 situations that we solve as follows: (1) E.g., in the *Event* class definition, *belongsToCalendar* *min* 1 *Thing* is equivalent to *belongsToCalendar* *some* *Thing*; and (2) In the *User* class definition, *hasNUsers* *exactly* 1 *int* can be substituted by *hasNUsers* *some* *int*. This fact, in addition to having *hasNUsers* as functional property, makes the number of relations *hasNUsers* to be exactly 1.

In general, activities are composed by a sequence of actions. However, this does not happen in practice all the time, e.g., some cycles can appear to complete “half done” actions, or forgotten things are made at the end. Some behaviour *procedures* are not followed in the logic order, but in the order in which the actions are actually needed, during its execution, without any specific sequence ordering (e.g., taking instruments while cooking). Since each behaviour has not a unique way to be performed, behaviours can be specified by a set of actions and/or activities that compose it. This is done through specifying the time of occurrence of actions, activities, and behaviours, and a membership function that represents the belonging of the action/activity to a given behaviour. Regarding the time of occurrence, actions have a crisp date-time as a timestamp. However, activities and actions have a *startDatetime*, *endDatetime*, and *duration*. Duration is a fuzzy data-type given by a triangular fuzzy function. Therefore, when an activity or behaviour is recognized, it will be with a degree of truth, and it will be logged as an activity or behaviour detected with a membership value in (0,1]. E.g.: The behaviour *GoingForAWalk* could have associated *recognizedWithDegree* (*NataliaGoingForAWalk*, 0.8).

4. Proof of concept and ontology validation

4.1. Case study: office domain and public buildings

Office and public buildings are a good scenario to test our approach, since the inclusion of the designed ontology into an automatic building control system could help to improve both energy efficiency and occupants' wellness. This section describes a proof of concept about the potential of the fuzzy ontology in this domain, and the next sections provide an evaluation and comparison with

crisp ontologies to show the benefits of the approach. The main elements that are used to model knowledge in the office university environment are the following:

- User status. A user can have at work different status such as away, at work, on leave, on holiday, busy, or available. Users can also have work positions (secretary, researcher, technical staff, students, lecturers, etc.). Generic users in this domain can be company representatives, internal representatives, research partners, visitor researchers, visitor students, technical staff, etc.
- Physical environment. Architectural environment is also represented in order to monitor activities through e.g. presence sensor or cameras. Part of the environment are locations and their corresponding GPS coordinates. For this reason, we allow the representation of semantic maps including different components of indoor locations such as different types of buildings (public and privates) and different specializations of rooms (lecture rooms, laboratories corridors, kitchen, etc.), and outdoor locations such as parks, terraces, and means of transport.
- Objects. Any element subjected to possible interaction can be considered in this category. From furniture like windows and doors to desktop tools and devices. These devices connect to a specific network (that provides the context), and can represent a role such as addressee or sender devices in a given communication.
- Actions. Actions considered are not only focused on the work environment (setting projectors on/off, turn devices on/off, check-in at location, etc.) but also include those that can be considered in other environments (open fridge, being observed at location, etc.).
- Activities. Activities in the work environment include general and specific tasks such as going to work, lectures, meetings, presentations, and writing.
- Behaviours. Behaviours can be defined for specific routines or patterns based on actions, activities, and other environmental conditions. For instance, it is possible to express relations of interest including *working for a certain project*, *being a partner, colleague (lab, department, university)*, *advisor, etc.* Modelling from general behaviours related with e.g., events and calendars, to in-office activities and object interaction is a possibility. When there is a lack of context information but there exists evidence of activities happening, we may reason with *UnknownUser(s)*, *UnknownLocation*, *UnknownActivity* or *UnknownBehaviour*. In these cases, other context information within the same time frame may be crucial to help disclosing the human behaviour.

The current version of the developed ontology, validated with the OWL reasoners Hermit 1.3.6, Pellet 2.3.0 (2.2.0 Protégé.gé plug-in) and fuzzyDL 1.1, consists of 228 classes, 133 object properties, 62 data properties and 33 test individuals, within *SRQIQ(D)* DL expressiveness. Pitfalls found in the ontology modelling process were corrected using the OOPS! Pitfall Scanner [50]. To put an example of behaviour definition, we can identify the actions/activities that compose it and model it with timing ordering relations. Next, some examples show applications of use of our ontology:

Example 1 (Defining a Behaviour). The behaviour “having a coffee break” can be composed by a set of actions: exiting the office, going through the corridor to the kitchen, taking coffee if somebody already made it or making coffee if there is not any left, and then coming back to your office. In ontology terms, this could be expressed with a nested hierarchy of actions and two activities, *MakeCoffee* and *TakeCoffee* (underlined):

- Behaviour **To have a coffee break:** OpenDoor WalkBy (Corridor) BeObservedInLocation (Kitchen) (MakeCoffee + TakeCoffee) WalkBy (Corridor) OpenDoor.
- **Activity MakeCoffee:** OpenCupboard TurnCoffeeMachineOn MoveObject (CoffeeJar) OpenFridge.
- **Activity TakeCoffee:** MoveObject (CoffeeJar) OpenFridge.

Fig. 7 shows the underlying state machine to recognize the same behaviour. Activities are represented with white large nodes, while actions are represented with small purple nodes. Activities are, at the same time, abstractions of other state machines able to recognize fine-grained activities. Fig. 8 shows, in more detail, the same behaviour at action level. This hierarchical and recursive design is useful to decompose the recognition process in smaller ones and provide different levels of abstraction. As an example, the activity *MakeCoffee* could be expressed with the following axioms:

$$\begin{aligned} \forall \text{ User : } & \text{performsAction}(\text{User}, \text{OpenCupboard}) \\ & \wedge \text{performsAction}(\text{User}, \text{TurnCoffeeMachineOn}) \\ & \wedge \text{performsAction}(\text{User}, \text{MoveCoffeeJar}) \\ & \wedge \text{performsAction}(\text{User}, \text{OpenFridge}) \\ \rightarrow & \text{performsActivity}(\text{User}, \text{MakeCoffee}). \end{aligned}$$

Example 2 (Defining a behaviour with OWL 2 axioms (II)). Let us suppose we want to define a meeting with the vice-chancellor or Rector as a special type of meeting. The concept *HavingRectorMeeting* can be defined as a meeting in any auditorium or meeting room, which is hosting at least 10 people and someone of those room occupants has the Rector work position. *HavingRectorMeeting* will be a specialization class of *Meeting*; more concretely, with the following restrictions:

$$\begin{aligned} \text{HavingRectorMeeting} = & \text{Meeting and (happensInLocationsome} \\ & \times ((\text{AuditoriumorMeetingRoom}) \text{ and} \\ & \times (\text{isLocationFor some (hasWorkPosition} \\ & \times \text{some Rector})) \text{ and (isHostingNPeople} \\ & \times \text{some int } [\geq \text{“10”} \wedge \text{int}]) \text{ and} \\ & \times (\text{hasProjector some} \\ & \times (\text{hasDeviceStatus value On}))))). \end{aligned}$$

Example 3 (Defining Crisp Rules). Concrete application domain rules can be modelled in pure OWL 2, Fuzzy OWL 2 (e.g. with Mamdani rules) or rule languages such as SWRL [51] or SPIN [52]. For instance, if there is a scheduled conference for more than 25 people, we can automatically activate the lights 10 min before the event starts. Let *L* be a variable for any Location, *X* be an integer variable, and *T* be a timestamp. Then the rule can be modelled as follows:

$$\begin{aligned} \forall \text{ Meeting, Event, L, X, T : } & \text{isProgrammedWithinEvent} \\ & \times (\text{Conference, Event}) \wedge \text{isScheduledAtLocation}(\text{Event, L}) \\ & \times \wedge \text{hasNAttendants}(\text{Event, X}) \wedge X > 25 \\ & \times \wedge \text{hasProgrammedStartDateTime}(\text{Event, T}) \\ & \times \rightarrow \text{atDatetime}(\text{TurnLightsOn, 'T - 00 : 10 : 00'}) \\ & \times \wedge \text{lightsAtLocation}(\text{TurnLightsOn, L}) \end{aligned}$$

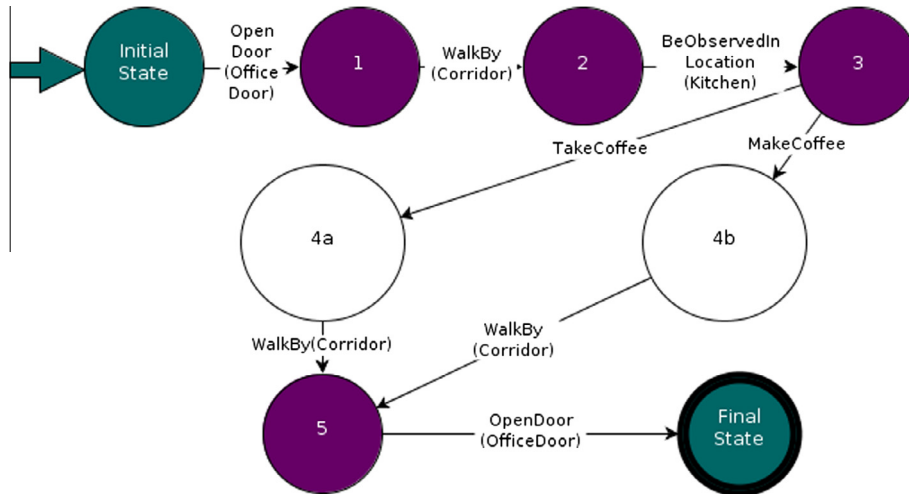


Fig. 7. Underlying regular automaton to model the behaviour *To have a coffee break*.

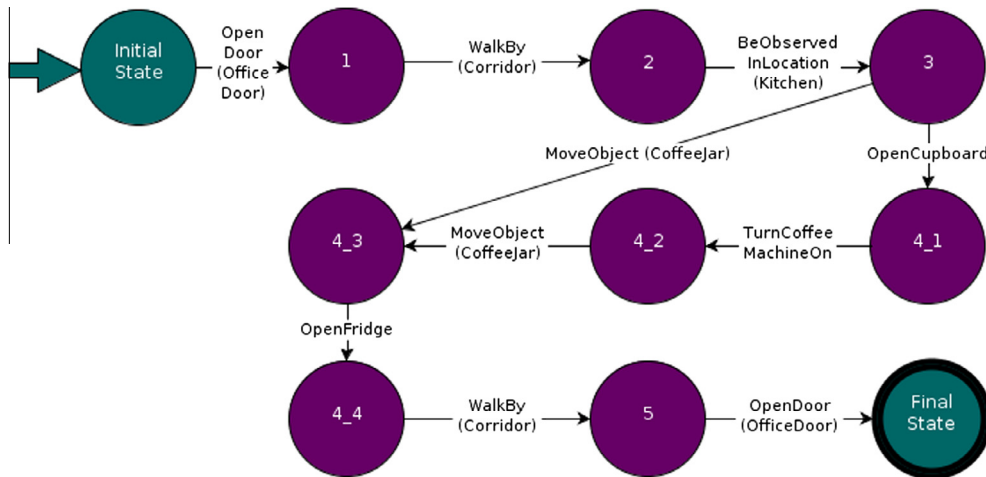


Fig. 8. Underlying regular automaton (at lower-Action level) to model the behaviour *To have a coffee break*.

TurnLightsOn is an example of external applications with service grounding. They are modelled as subclasses of the *Service* class and their associated data or object properties represent the application parameters.

Example 4 (Executing Fuzzy Queries). There are several ways of running a query. One of the aims of running a query is finding if a rule is triggering. For instance, to determine the minimal degree to which individual *NataliasNokiaN8* is an instance of concept *Phone*, we would run: *(min-instance? NataliasNokiaN8 Phone)*. However, if we want to know all *Phone* concept instances, the following query applies the previous query to every individual in the KB: *(all-instances? Phone)*.

On the other hand, if we want to determine the maximal degree to which individual pair *(Natalia, JohanLiliusOffice)* is an instance of role *isInLocation*, this can be obtained as: *(max-related? Natalia JohanLiliusOffice isInLocation)*. In the opposite side, another possible query is finding the minimal degree to which a concept A, e.g., *VeryFullCapacity*, subsumes a concept B, e.g., *FullCapacity*. This query would be expressed as: *(min-subs? VeryFullCapacity FullCapacity)*. Optionally, an individual, as well as Lukasiewicz, Gödel, or Kleene-Dienes implications, can be used for this type of query [45].

Example 5 (Defining Fuzzy Rules). Fuzzy rules in *fuzzyDL* can be expressed with the Mamdani structure or as implication rules. These can be mapped to a set of statements in a fuzzy KB as a fuzzy control system [45]. The definition of logical rules as Mamdani rules is performed as follows: *(define-concept MamdaniRuleBase (g-or Rule1 (...) RuleN))*. For example, the rule “If the User Natalia is away for the weekend and the weather situation in Turku becomes very stormy, all electricity appliances should be turned off” can be expressed as: *(define-concept Rule1 = (g-and (Natalia (some hasStatus AwayForWeekend)) (WeatherSituationTurku (some isCurrently VeryStormy)) (TurnOffAllElectricitySwitches (some with-Params NataliasAppartment))))*.

The input to the controller/facts can be done using the following syntax: *(instance input (= and WeatherSituationTurku (some isCurrently NearlyCloudy))) (instance input (= and Natalia (some hasStatus AtWork))) (...)*.

In addition to the previous example, we may be interested in finding the real value of a fuzzy concept when using different fuzzy linguistic labels. This process is known as defuzzification and it can be done with the following command for the previous rules: *(defuzzify-lom? MamdaniRuleBase input TurnOffAllElectricitySwitches)*.

Alternatively and equivalently, the definition of logical rules can be done as implication rules instead. In this case, we show how to

Table 1
Fuzzy Rules 1–5.

Rule 1 description	If a person opens the kitchen cupboard to take coffee, turns the coffee machine on, holds the coffee jar, and possibly takes milk from the fridge, then the person is making coffee
Definition	<i>(define-concept antecedent1 (g-and User (some performsAction MakeCoffee)))</i> <i>(define-concept MakeCoffee (w-sum (0.1 OpenCupboard) (0.5 TurnCoffeeMachineOn) (0.3 MoveCoffeeJar) (0.1 OpenFridge)))</i> <i>(define-concept consequent1 (g-and User (some performsActivity MakingCoffee)))</i> <i>(define-concept Rule1 (l-implies antecedent1 consequent1))</i>
Rule 2 description	If a meeting room has the lights on, there is a meeting going on at the room
Definition	<i>(define-concept antecedent2 (g-and MeetingRoom (some hasLights (some hasStatus ON))))</i> <i>(define-concept consequent2 (g-and MeetingRoom (some hasOccupancyStatus HoldingAMeeting)))</i> <i>(define-concept Rule2 (l-implies antecedent2 consequent2))</i>
Rule 3 description	If a person makes use of a bottle, a plate, a fork, a spoon, and a knife, we have high certainty that he/she is having lunch
Definition	<i>(define-concept antecedent3 (g-and User (some performsAction Lunch)))</i> <i>(define-concept Lunch (w-sum (0.1 UseBottle) (0.4 UsePlate) (0.2 UseFork) (0.2 UseSpoon) (0.1 UseKnife)))</i> <i>(define-concept consequent3 (g-and User (some performsActivity HaveLunch)))</i> <i>(define-concept Rule3 (l-implies antecedent3 consequent3))</i>
Rule 4 description	If a person goes out of his/her office, passes by the corridor, opens the storage room, and takes the exercise stick, then he/she is performing a Keppijumpa stretching exercise
Definition	<i>(define-concept antecedent4 (g-and User (some performsAction KeppijumpaStretchingExercise)))</i> <i>(define-concept KeppijumpaStretchingExercise (w-sum (0.1 ExitOffice) (0.1 ExitCorridor) (0.1 OpenStorageRoom) (0.4 UseStick)))</i> <i>(define-concept consequent4 (g-and User (some performsActivity DoStretching)))</i> <i>(define-concept Rule4 (l-implies antecedent4 consequent4))</i>
Rule 5 description	If a user's phone is in a location near the office of the head of the laboratory, then the user is having a meeting with his/her supervisor
Definition	<i>(define-concept antecedent5 (g-and User (some hasPhone (some isInLocation (some isNearTo JohanLiliusOffice))))</i> <i>(define-concept consequent5 (g-and User (some performsActivity MeetingSupervisor)))</i> <i>(define-concept Rule5 (l-implies antecedent5 consequent5))</i>

encode a rule that detects if Natalia's phone is in a location near Johan's office. If this happens, it is recognized that they are having a meeting and starts recording in her phone the agenda and transcribing it from her phone to her calendar to have it into account for the next meeting:

(define-concept antecedents (and (Natalia (some hasPhone P)) (and (Natalia (some hasCalendar C)) (and (P (some isInLocation L))) (and L (some isVeryNearTo JohansOffice))))))

(define-concept consequents (and (StartAudioRecording (some withParams P)) (TranscribeMeetingAgenda (some withParams (P and C))))))

(define-concept Rule2 (l-implies (g-and antecedents) (g-and consequents)))

The query for the consequent's satisfiability degree could be carried out with the query: *(min-instance? input consequents)*.

4.2. Ontology validation

In this section, we show the benefits of a fuzzy ontology for human behaviour recognition with respect to crisp approaches. With that purpose, we define two evaluation parameters. The scalability is understood as the capability of the ontology to perform with a rule set and a reasoner to achieve activity recognition, in reasonable execution time, for large amounts of data size (KB' size). The satisfiability degree (or firing accuracy of rules, $\in [0, 1]$) is another parameter considered, not directly present in crisp ontologies, where either an activity is recognized or not. The satisfiability degree influences the recognition accuracy and gives more precise information about the certainty of having recognized certain activity happening.

Although the literature offers a wide variety of activity recognition datasets, it is rare to find them expressed in the form of semantic ontology-based axioms. One exception is the Opportunity Dataset [53], adapted to an ontological framework [54]. Their multilevel activity ontology-based dataset is validated with ELOG reasoner and it shows a high degree of concurrency in fine-grained activities [54]. ELOG is a probabilistic reasoner for OWL $\mathcal{EL}$, focused

on the log-linear description logic based on $\mathcal{EL}++$ without nominals and concrete domains ($\mathcal{EL}++-LL$). It is worth noting that the computational cost of OWL $\mathcal{EL}$ is much lower than for *fuzzyDL*. However, this is a trade-off that makes *fuzzyDL* stronger due to its *SHIF* expressivity, much superior to the one of OWL $\mathcal{EL}$.

When dealing with uncertainty, it is important to distinguish when probabilistic reasoning is suitable with respect to fuzzy reasoning. Probabilistic reasoning can model uncertainty associated to previous experience cases. However, fuzzy reasoning can help modelling vagueness or natural language-based descriptions, based on knowledge representation. For instance, *almost every day* Peter has muesli for breakfast. Expressions such as “almost every day”, “quite”, and “little”. could be modelled to better preserve natural language when expressing rules from experts. To validate our own ontology, we create a fuzzy rule KB for human activity recognition with concepts and relationships. In addition, for more complex queries, based on triple patterns such as in standard SPARQL queries, a mapping can be seen in [55] between triple pattern queries (s, p, o) and *fuzzyDL* queries. E.g. for queries with wildcards such as, e.g., (? , p, o), the following *fuzzyDL* query could be used to find the minimal degree of satisfiability for that given predicate form:

If $D \in p$. Domain: $\forall$ Individual $i \in D$: (min-related? i o p).

For these reasons, as we require a fuzzy reasoner due to semantics, we cannot reuse existing datasets and therefore, we designed our own set of rules for *fuzzyDL* 2.04 (using Gurobi optimizer 5.0.2).

The rules we used for the ontology validation experiment are shown in Tables 1 and 2, but the whole file with *fuzzyDL* instances and other definitions and queries is available online together with the ontology.⁷ Additional information about the *fuzzyDL* syntax may be found in <http://gaia.isti.cnr.it/straccia/software/fuzzyDL/fuzzyDL.html>.

The goal of our experiment is to verify that fuzzy ontologies can outperform crisp ontologies in human behaviour recognition. Thus,

⁷ Fuzzy Human Behaviour Ontology, rules and queries: <http://users.abo.fi/ndiaz/public/FuzzyHumanBehaviourOntology/>.

Table 2
Fuzzy Rules 6–8.

Properties for Rules 6, 7, 8	Fuzzy properties for representing the personal average sleep quality in number of sleeping h/day, average number of steps walked per day, and overall immune-defense level for overall health in a integer scale of [0, 10]
Definition	<pre> (functional hasAvgSleepQuality) (functional didAvgNSteps) (functional hasImmuneDefenseLevel) (range hasAvgSleepQuality "real" 0.0 20.0) (range didAvgNSteps "integer" 0 50000) (range hasImmuneDefenseLevel "integer" 0 10) (define-fuzzy-concept lowImmuneDefenseLevel left-shoulder (0, 10, 0, 5)) (define-fuzzy-concept badSleepQuality left-shoulder (0.0, 20, 4.0, 5.0)) (define-fuzzy-concept highNSteps right-shoulder (0, 50000, 7000, 10000)) (define-fuzzy-concept lowNSteps left-shoulder (0, 50000, 2000, 2500)) </pre>
Rule 6 description	If sleep quality is very bad and pedometer-based step counter measures a low number and stress is high, then immune-defense is low (having a coffee break may be recommended)
Definition	<pre> (define-concept antecedent6 (g-and (some hasAvgSleepQuality badSleepQuality) (some didAvgNSteps highNSteps))) (define-concept consequent6 (g-and User (some hasImmuneDefenseLevel lowImmuneDefenseLevel))) </pre>
Rule 7 description	Analoge to Rule 6 but caused when the person is not having enough exercise
Definition	<pre> (define-concept antecedent7 (g-and (some hasAvgSleepQuality badSleepQuality) (some didAvgNSteps lowNSteps))) (define-concept consequent7 (g-and User (some hasImmuneDefenseLevel lowImmuneDefenseLevel))) </pre>
Rule 8 description	A disjunction among two previously defined rules
Definition	<pre> (define-concept Rule8 (g-or Rule6 Rule7)) </pre>

for the test, we created 8 implicative rules that were applied over the fuzzy approach and its corresponding crisp ontology. Rules 1–5 test flexibility in the sense that skipping some actions does not prevent the recognition of a behaviour. These rules were designed with the aim of giving different weights to certain actions within an activity, so that weights can strengthen the importance of the different actions that compose an activity to finally give an overall degree of satisfiability. On the other hand, rules 6, 7, and 8 use fuzzy membership functions such as the following:

```

(define-fuzzy-concept lowImmuneDefenseLevelCrisp left-shoulder (0, 10, 0, 5))
(define-fuzzy-concept badSleepQualityCrisp left-shoulder (0.0, 20, 4.0, 5.0))
(define-fuzzy-concept highNStepsCrisp right-shoulder (0, 50000, 7000, 10000))
(define-fuzzy-concept lowNSteps left-shoulder (0, 50000, 2000, 2500))

```

To transform these fuzzy concepts to the crisp case, thresholds need to be taken from the membership functions above and then hard-coded to work as thresholds as shown by the next lines:

```

(instance lowImmuneDefenseLevelCrisp ImmuneDefenseLevel)
(define-concept antecedent6 (and (>= hasAvgSleepQuality 5) (>= didAvgNSteps 7000)))
(define-concept consequent6 (and User (some hasImmuneDefenseLevel lowImmuneDefenseLevelCrisp)))

```

This rule formulation for the crisp case shows more limitations in expressivity and further reasoning than the fuzzy approach, since the formulation of the rule above would not allow linguistic variables for further imprecise queries of the type (*all-instances?* (*some hasImmuneDefenseLevel lowImmuneDefenseLevel*)).

After creating the set of rules, we instantiated individuals and formulated queries 1–8 in Table 3, to verify the triggering of rules 1–8, respectively. All queries were designed so that they should fire (in fuzzy case, to certain degree), except rule 6, that is made to not fire at all. On the fuzzy case, we assume that the firing of a rule to detect a behaviour is subject to an activity/behaviour-dependent threshold. For the sake of simplicity, in Table 4, we considered as fired those rules whose answer had a degree of satisfiability larger than zero.

This experiment helps us to verify the expressiveness power and practical implications of fuzzy ontologies with respect to crisp approaches. In Table 4, none of the rules fires except rules 7 and 8 for the crisp case. On the other hand, all rules except rule 6 fire in the fuzzy case, with a degree of truth or certainty. An example is rule 4: Here, *KeppijumpaStretchingExercise* is a concept that is the

Table 3
Fuzzy queries.

Description	Code
Query 1	<pre>(min-instance? Victor (some performsActivity MakingCoffee))</pre>
Query 2	<pre>(min-instance? ADARoom (some hasOccupancyStatus HoldingAMeeting))</pre>
Query 3	<pre>(min-instance? Robin (some performsActivity HaveLunch))</pre>
Query 4	<pre>(min-instance? Natalia (some performsActivity DoStretching))</pre>
Query 5	<pre>(min-instance? Ana (some performsActivity MeetingSupervisor))</pre>
Query 6	<pre>(min-instance? facts Rule6)</pre>
Query 7	<pre>(min-instance? facts Rule7)</pre>
Query 8	<pre>(define-concept Rule8 (g-or Rule6 Rule7)) (min-instance? facts Rule8)</pre>
fuzzyDL Facts for queries 6, 7 and 8	<pre> (instance facts (= hasAvgSleepQuality 3) 0.7) (instance facts (= didAvgNSteps 19000)) (instance Facts 1 (= hasAvgSleepQuality 4) 0.2) (instance Facts 1 (= didAvgNSteps 2000) 0.9) </pre>

Table 4
Degree of satisfiability for crisp VS fuzzy (equivalent) human behaviour rules. In crisp case, 0 = No firing; 1 = Firing.

Query	Crisp answer	Fuzzy answer
Query 1	0	0.1
Query 2	0	0.5
Query 3	0	0.4
Query 4	0	0.4
Query 5	0	0.8
Query 6	0	0.0
Query 7	1	1.0
Query 8	1	1.0
Firing accuracy	37.5%	100%

Table 5

Average execution time (in s) for each rule (fuzzy and crisp) in different Knowledge Base (KB) sizes in number of individuals/instances (User).

Crisp and fuzzy query	KB size = 100	KB size = 1000	KB size = 10000	KB size = 100000
Crisp Q1	0.0754	0.1688	3.2986	911.222
Crisp Q2	0.074	0.1656	3.0992	962.054
Crisp Q3	0.0752	0.1622	3.3778	984.197
Crisp Q4	0.0752	0.1606	3.2124	1071.07
Crisp Q5	0.0778	0.1656	3.3284	1076.05
Crisp Q6	0.0834	0.156	3.2374	1001.38
Crisp Q7	0.078	0.1656	3.0688	1021.18
Crisp Q8	0.078	0.1626	3.2016	1082.07
Fuzzy Q1	0.0846	0.172	3.1594	800.31
Fuzzy Q2	0.078	0.1626	3.2844	1008.21
Fuzzy Q3	0.0814	0.1626	3.5488	1025.97
Fuzzy Q4	0.078	0.1656	3.4594	912.07
Fuzzy Q5	0.0782	0.175	3.3816	1138.75
Fuzzy Q6	0.0812	0.175	3.5252	1128.29
Fuzzy Q7	0.0782	0.1812	3.3534	978.135
Fuzzy Q8	0.0876	0.1814	3.7248	1078.7138

weighted sum of the concepts it is composed of. In this case, if the instance *Natalia* performs the actions going out of her office, exiting the corridor, opening the storage room and using the exercise stick, we can recognize that she did the stretching exercise session. This is expressed in *fuzzyDL* as follows:

(related *Natalia* exitOffice performsAction 0.6)
 (related *Natalia* exitCorridor performsAction 1.0)
 (related *Natalia* openStorageRoom performsAction 1.0)
 (related *Natalia* useStick performsAction 1.0)

In the crisp case, if for any reason, one of these actions is not detected, the activity *DoStretching* is not recognized, as Table 4 shows. However, the fuzzy approach still fires the rule with a certainty degree value lower than 1. To implement this behaviour in the crisp ontology, we need to establish a threshold criteria for each fact, behaviour, or object to be considered in the case that the certainty of a fact is not 1, so that the rules can fire. This may be done manually or using optimization techniques such as genetic or evolutionary algorithms. This is part of future work directions on specific domain scenarios where threshold values need to be tuned and optimized accordingly. However, a fuzzy approach deals with uncertainty and eases the management of these situations. Thus, we may conclude that fuzzy ontologies can provide improvements in expressivity with respect to the crisp cases, but also in performance due to their power to manage uncertainty.

Having shown the benefits of fuzzy ontologies over crisp ones, in modelling human behaviour, we are also interested in evaluating if our approach scales to large sizes of Knowledge Bases. With this purpose, we measured execution times for queries 1–8. We implemented the equivalent set of crisp queries in *fuzzyDL* as well, to evaluate performance and recognition accuracy and to better assess the rules' scalability factor in practice. Both fuzzy and crisp experiments can be compared in Table 5, and the differences among average times of our 8-query dataset can be seen in Table 6. We must notice that execution time does not vary substantially for

fuzzy queries with respect to crisp queries. However, a larger increase in average time occurs when we enlarge the KB dataset to at least 10^5 instances in both crisp and fuzzy cases. In the fuzzy case, query average response time goes to the order of up to $\sim(13 - 16)$ minutes. We are aware that this delay may be too long if the purpose is to notify about safety-critical activities. Therefore, in the future, we will focus on prioritizing the detection of critical activities so that the current reaction time obtained can decrease.

It is important to note that, although KBs of size 10^5 impose a considerable increase of execution time with respect to KBs of size 10^4 (from orders of about 300 times more), this is not significant due to the comparison with respect to crisp reasoning in KBs of the same size. In fact, importantly enough, for KB sizes of 10^5 instances, some queries are faster in the fuzzy case than in the crisp one. *FuzzyDL* internal reasoning tableau algorithm optimizations for large numbers of axioms/instances, Java virtual memory swapping or cache memory functioning seem to be implementation-related reasons for this phenomenon to happen. In any case, it is worth mentioning that all times were measured when running each query isolated and independently from others. This is to make queries comparable, since once a first query has been answered, the rest of queries take much less time due to the reuse of the internal graph model already built. As conclusion, we can affirm that fuzzy ontology-reasoning for activity recognition is scalable.

Both crisp and fuzzy experiments, measuring query running time in seconds, were run on an Intel(R) Core i7-4500@1.80 GHZ 2.40 GHZ, 8 GB RAM 64-bit and Windows 8.1. A limitation of *fuzzyDL* is the maximum number of individuals allowed (maxIndividuals = 1000000000). Nevertheless, a KB of size 10^6 was not handled by Java memory.

As a final remark, and considering future larger data size scenarios, we believe on the potential of combining the use of a crisp-fuzzy hybrid architecture approach of KBs for performance and scaling reasons. This would allow advantages of both paradigms to be fully exploited (see further discussion in [55]).

5. Conclusions and future work

Knowledge-based techniques, such as ontology-based activity modelling, add a set of advantages for incremental and context-aware recognition. It is a suitable approach to achieve interoperability, abstraction, and modularity in an easier way. However, some expressiveness limitations in OWL DL are also found related with the lack of support for temporal reasoning [35] and often, an external rule engine is used to express more complex modelling. A set of ontologies for human activity modelling has been created in the recent past [4] which is able to deal with different context

Table 6

Average execution time differences (in s) for the 8 queries (fuzzy and crisp) dataset and different Knowledge Base (KB) sizes in number of individuals/instances (User).

KB size in number of individuals (Users)	Average query running time (s) difference (fuzzy–crisp)	Standard deviation
10^2	0.0038	0.0047
10^3	0.0086	0.0086
10^4	0.2016	0.205
10^5	–4.8468	128.8485

information. The proposal in this work shows a set of advantages in respect to the existing ones. The main contribution is the support to model and treat uncertain, incomplete, vague, or imprecise information, therefore easing the modelling of more flexible models as well as allowing incomplete but real-life queries. For instance, we may want to get notifications the days when a certain office has “very high” temperatures, or we may want to know at what time window our work’s restaurant is least crowded. When exact numbers are not known or they are not relevant to get a fast answer, imprecise knowledge eases the task of information retrieval. In addition, we demonstrate that fuzzy ontologies may be more realistic and provide better accuracy for human behaviour recognition than crisp ontologies. In the experimental section, we show that if facts are not completely true, crisp rules cannot fire, while a fuzzy approach fires them with a satisfiability or certainty degree. Solving this situation in crisp solutions would require a continuous threshold management that would make the problem more complex, while fuzzy systems deal with this type of situations in a natural way. Apart from being more accurate, we also showed that the fuzzy approach is scalable for larger sizes of KBs.

Furthermore, varied (from fine to coarse-grained) levels of abstraction are provided to identify atomic actions, activities composed by actions, and behaviours comprised by an aggregation of the latter two. Recursion, for more flexible and scalable modelling, is also allowed at activity and behaviour levels. A behaviour can be customized and associated to a unique user, user group, or certain type of action, activity, or context dependence. Our ontology is applied to human behaviour recognition in the office and public buildings domain, but is easily expandable to other domains. Since we provide different granularity levels of activities, incremental context acquisition – typical of ontology-based reasoning – is supported and made easier to abstract behaviours. Likewise, this results on a precise and accurate definition of activities. We provide not only knowledge engineering for dealing with uncertainty, but we also allow for modelling activities where its actors or locations where they take place are unknown. These conditions allow, in this way, more realistic settings and also, more abstract and different group activities.

After experimenting with a real life and complex enough ontology, we can affirm that there is a need for more complete fuzzy reasoners that can handle real-time notifications (such as a subscription mechanism as in e.g. M3 RDF store [56]) to avoid bottlenecks with constant querying. This does not occlude *fuzzyDL*’s potential and the fact that it has shown to be successful in diverse domains. In our presented case study, modelling rules in *fuzzyDL* also showed some challenges. As *fuzzyDL* does not allow yet to express implication rules where the subject in a triple (s,p,o) is a concrete individual, our experiment focused on general rules for individuals of a given class C that acts as the subject of the query. In the future, we expect to have more efficient ways of concretizing the rule so that it can specifically apply to unique instances/individuals so as to achieve a proper rule personalization. At the moment, three workarounds solve this situation. a) An extra class (e.g. *NataliaClass* concept) can be created for each rule we want it to affect solely to a concrete individual (e.g. individual *Natalia* from class *User*). This makes explicit, by naming a class with an individual’s name, that rule should only apply to the given individual. b) The firing of the rule for a given individual can be detected by first instantiating the individual of interest (in this case *Natalia*) as an instance of that rule (e.g., Rule 4): *((instance Natalia Rule4))*. This step is required to give a correct answer in the second step. Secondly, a rule firing can be detected by querying the degree of satisfiability of individual *Natalia* satisfying Rule 4. This is done by querying for *Natalia* being an instance of the rule’s consequent (this applies to a given moment, i.e. state of the KB). E.g.: *(min-instance? Natalia (some performsActivity DoStretching))*. c) It is also possible to

find all individuals satisfying a given rule by querying: *(all-instances? Rule4)* after having instantiated all individuals that we want the rule to apply to (as in step b). E.g.: *(instance Natalia Rule4)*.

Although the novelty of our ontology is to overcome the problems of existing proposals which are unable to model uncertainty, the limitations are those common problems of ontology-based modelling on complex behaviours. This is because ontology-based modelling may not overcome performance as its most characteristic feature and definitions can seem unnatural at times [35]. Therefore, future works should consider these aspects more specifically and improve fuzzy ontology modelling real human interaction, as well as interfaces for end-users. One possible solution would be to create hybrid approaches involving data-driven techniques and fuzzy ontologies for human behaviour recognition as a next step in our research. Another future direction to explore is modelling and detecting human behaviour changes. Using *learning instance matching*, i.e., (data level, non-schema) ontology mapping for new data integration [57], could be an approach towards automating the evolution or learning of behaviour changes.

Acknowledgments

This work was funded by TUCS (Turku Centre for Computer Science), Hans Bang Foundation, Finnish Cultural Foundation, Nokia Foundation, CIE BioTIC Project CEI2013-P-3, and Project TIN2012-30939 from National I + D Research Program (Spain). We also thank Fernando Bobillo and Robin Wikström for their advice with *FuzzyOWL* and *fuzzyDL* tools.

References

- [1] R. Casas, R. Blasco Marín, A. Robinet, A.R. Delgado, A.R. Yarza, J. McGinn, R. Picking, V. Grout, User modelling in ambient intelligence for elderly and disabled people, in: Proceedings of the 11th International Conference on Computers Helping People with Special Needs, ICCHP '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 114–122. http://dx.doi.org/10.1007/978-3-540-70540-6_15.
- [2] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, X. Hong, Semantic smart homes: towards knowledge rich assisted living environments, in: S. McClean, P. Millard, E. El-Darzi, C. Nugent (Eds.), Intelligent Patient Management, Studies in Computational Intelligence, vol. 189, Springer, Berlin, Heidelberg, 2009, pp. 279–296. http://dx.doi.org/10.1007/978-3-642-00179-6_17.
- [3] L. Chen, C.D. Nugent, Ontology-based activity recognition in intelligent pervasive environments, *Int. J. Web Inform. Syst. (IJWIS)* 5 (4) (2009) 410–430.
- [4] N. Díaz Rodríguez, M.P. Cuéllar, J. Lilius, M.D. Calvo-Flores, A survey on ontologies for human behavior recognition, *ACM Comput. Surv.* 46 (4) (2014) 43:1–43:33. <http://doi.acm.org/10.1145/2523819>.
- [5] R. Kowalski, M. Sergot, A logic-based calculus of events, *New Gen. Comput.* 4 (1) (1986) 67–95. <http://dx.doi.org/10.1007/BF03037383>.
- [6] F. Doctor, H. Hagrass, V. Callaghan, A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments, *IEEE Trans. Syst. Man Cybern. Part A* 35 (1) (2005) 55–65.
- [7] H. Hagrass, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, H. Duman, Creating an ambient-intelligence environment using embedded agents, *Intell. Syst. IEEE* 19 (6) (2004) 12–20.
- [8] D. Riboni, C. Bettini, OWL 2 modeling and reasoning with complex human activities, *Pervasive Mob. Comput.* 7 (3) (2011) 379–395. <http://dx.doi.org/10.1016/j.pmcj.2011.02.001>.
- [9] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, X. Hong, M. Poland, Using event calculus for behaviour reasoning and assistance in a smart home, in: S. Helal, S. Mitra, J. Wong, C. Chang, M. Mokhtari (Eds.), Smart Homes and Health Telematics, Lecture Notes in Computer Science, vol. 5120, Springer, Berlin, Heidelberg, 2008, pp. 81–89. http://dx.doi.org/10.1007/978-3-540-69916-3_10.
- [10] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, S. Devlin, Evidential fusion of sensor data for activity recognition in smart homes, *Pervasive Mob. Comput.* 5 (3) (2009) 236–252. <http://dx.doi.org/10.1016/j.pmcj.2008.05.002>. Pervasive Health and Wellness Management. <<http://www.sciencedirect.com/science/article/pii/S157411920800045X>>.
- [11] G. Villarrubia, A. Sánchez, I. Barri, E. Rubión, A. Fernández, C. Rebate, J.A. Cabo, T. álamos, J. Sanz, J. Seco, C. Zato, J. Bajo, S. Rodríguez, J. Corchado, Proximity detection prototype adapted to a work environment, in: P. Novais, K. Hallenborg, D.I. Tapia, J.M.C. Rodríguez (Eds.), Ambient Intelligence – Software and Applications, Advances in Intelligent and Soft Computing, vol. 153, Springer, Berlin, Heidelberg, 2012, pp. 51–58. http://dx.doi.org/10.1007/978-3-642-28783-1_7.

- [12] A. Adler, R. Davis, Speech and sketching for multimodal design, in: Proceedings of the 9th International Conference on Intelligent User Interfaces, 2004, pp. 214–216.
- [13] V. Stanford, Infrastructure for distributed and embedded systems and interfaces, in: Proceedings of the Embedded Systems, Ambient Intelligence and Smart Surroundings Conference, 2005.
- [14] C.L. Gal, Smart offices, in: *Smart Environments: Technology, Protocols, and Applications*, first ed., Wiley Series on Parallel and Distributed Computing, Wiley, 2004.
- [15] M. Stoettinger, Context-Awareness in Industrial Environments, 2004. <www.mobile-safety.com>.
- [16] W. Shen, S. Lang, L. Wang, iShopFloor: an internet-enabled agent-based intelligent shop floor, *IEEE Trans. Syst. Man Cybern. Part C* 35 (3) (2005) 371–381.
- [17] H. Chen, T. Finin, A. Joshi, Semantic web in a pervasive context-aware architecture, in: *Artificial Intelligence in Mobile System (AIMS 2003)*, In conjunction with UBIComp, 2003, pp. 33–40.
- [18] A. D'Elia, L. Roffia, G. Zamagni, F. Vergari, P. Bellavista, A. Toninelli, S. Mattarozzi, Smart applications for the maintenance of large buildings: how to achieve ontology-based interoperability at the information level, in: *IEEE Symposium on Computers and Communications (ISCC)*, 2010, pp. 1077–1082. <http://dx.doi.org/10.1109/ISCC.2010.5546639>.
- [19] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, D. Chakraborty, Intelligent agents meet the semantic web in smart spaces, *IEEE Internet Comput.* 8 (6) (2004) 69–79.
- [20] H. Chen, T. Finin, A. Joshi, An Ontology for context-aware pervasive computing environments, in: *Proceedings of the Workshop on Ontologies in Agent Systems*, 2003.
- [21] G. Marreiros, R. Santos, C. Ramos, J. Neves, P. Novais, J. Machado, J. Bulas-Cruz, Ambient intelligence in emotion based ubiquitous decision making, in: *Proceedings of the Second Workshop on Artificial Intelligence Techniques for Ambient Intelligence*, 2007, pp. 86–91.
- [22] A. Fox, B. Johanson, P. Hanrahan, T. Winograd, Integrating information appliances into an interactive space, *IEEE Comput. Graph. Appl.* 20 (3) (2000) 54–65.
- [23] D. Riboni, L. Pareschi, L. Radaelli, C. Bettini, Is ontology-based activity recognition really effective?, in: *PerCom Workshops*, 2011, pp. 427–431.
- [24] L. Peña, Identity, fuzziness and noncontradiction, *Nous* 18 (2) (1984) 227–259.
- [25] E. Sanchez, T. Yamanoi, Fuzzy Ontologies for the Semantic Web, *Flexible Query Answering Systems*, 2006, pp. 691–699.
- [26] A. Pakonen, T. Tommila, J. Hirvonen, A fuzzy ontology based approach for mobilising industrial plant knowledge, in: *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010, pp. 1–8. <http://dx.doi.org/10.1109/ETFA.2010.5641200>.
- [27] O. Corcho, M. Fernández-López, A. Gómez-Pérez, Methodologies, tools and languages for building ontologies: where is their meeting point?, *Data Knowl. Eng.* 46 (1) (2003) 41–64. [http://dx.doi.org/10.1016/S0169-023X\(02\)00195-7](http://dx.doi.org/10.1016/S0169-023X(02)00195-7).
- [28] W.N. Borst, Construction of Engineering Ontologies for Knowledge Sharing and Reuse, Ph.D. Thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands, 1997.
- [29] F. Bobillo, Managing Vagueness in Ontologies, Ph.D. Thesis, University of Granada, Spain, 2008.
- [30] K. Amailef, J. Lu, Ontology-supported case-based reasoning approach for intelligent m-government emergency response services, *Decis. Support Syst.* 55 (1) (2013) 79–97. <http://dx.doi.org/10.1016/j.dss.2012.12.034>. <<http://www.sciencedirect.com/science/article/pii/S0167923613000043>>.
- [31] C.-L. Liu, H.-L. Yang, Applying ontology-based blog to detect information system post-development change requests conflicts, *Inform. Syst. Front.* 14 (5) (2012) 1019–1032. <http://dx.doi.org/10.1007/s10796-011-9314-0>.
- [32] L.L. Chen, J. Biswas, Tutorial: an introduction to ontology-based activity recognition, in: *Proceedings of the 7th International Conference on Mobile Computing and Multimedia (MoMM2009)*. <http://www.iwas.org/conferences/iwas2009/doc/Tutorial_LukeChen.pdf>.
- [33] M. Baldauf, S. Dustdar, F. Rosenberg, A survey on context-aware systems, *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, Inderscience Publishers, Geneva, Switzerland, 2007. <http://dx.doi.org/10.1504/IJAHUC.2007.014070>.
- [34] M. Saleemi, N. Díaz Rodríguez, J. Lilius, I. Porres, A framework for context-aware applications for smart spaces, in: S. Balandin, Y. Koucheryavi, H. Hu (Eds.), *ruSMART 2011: The 4th Conference on Smart Spaces*, 2011, LNCS, St. Petersburg, 2011, pp. 14–25. <<http://www.springerlink.com/content/d8618k217710th32/>>.
- [35] D. Riboni, C. Bettini, COSAR: hybrid reasoning for context-aware activity recognition, *Pers. Ubiquitous Comput.* 15 (3) (2011) 271–289.
- [36] B. Krose, van T. Kasteren, C. Gibson, van den T. Dool, CARE: context awareness in residences for elderly, in: *Proceedings of ISG'08: The 6th International Conference of the International Society for Gerontechnology*, 2008, pp. 101–105.
- [37] J. Gómez-Romero, M.A. Patricio, J. García, J.M. Molina, Ontology-based context representation and reasoning for object tracking and scene interpretation in video, *Expert Syst. Appl.* 38 (6) (2011) 7494–7510. <http://dx.doi.org/10.1016/j.eswa.2010.12.118>.
- [38] D. Preuveneers, J.V.D. Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, E. Berbers, K. Coninx, K.D. Bosschere, Towards an extensible context ontology for ambient intelligence, in: *Proceedings of the Second European Symposium on Ambient Intelligence*, Springer-Verlag, 2004, pp. 148–159.
- [39] J. Cantera-Fonseca, R. Lewis, Delivery Context Ontology. W3C Working Draft, June 2009.
- [40] H. Chen, T. Finin, A. Joshi, The SOUPA ontology for pervasive computing, in: V. Tamma, S. Cranefield, T. Finin, S. Willmott (Eds.), *Ontologies for Agents: Theory and Experiences*, Whittestein Series in Software Agent Technologies, Birkhäuser Basel, 2005, pp. 233–258. <http://dx.doi.org/10.1007/3-7643-7361-X-10>.
- [41] M. Poveda-Villalón, M.C. Suárez-Figueroa, R. García-Castro, A. Gómez-Pérez, A context ontology for mobile environments, in: *Proceedings of Workshop on Context, Information and Ontologies – CIAO 2010 Co-located with EKAW 2010*, vol. 626, CEUR-WS, Germany, 2010. <<http://oa.upm.es/5414/>>.
- [42] X.H. Wang, D. Zhang, T. Gu, H. Pung, Ontology based context modeling and reasoning using OWL, in: *Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications*, 2004.
- [43] R. Hervás, J. Bravo, J. Fontecha, A context model based on ontological languages: a proposal for information visualization, *J. Univ. Comput. Sci.* 16 (12) (2010) 1539–1555. <http://www.jucs.org/jucs_16_12/a_context_model_based/>.
- [44] S.S. Yau, J. Liu, Hierarchical situation modeling and reasoning for pervasive computing, in: *Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, and the Second International Workshop on Collaborative Computing, Integration, and Assurance, SEUS-WCCIA '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 5–10. <http://dx.doi.org/10.1109/SEUS-WCCIA.2006.25>.
- [45] F. Bobillo, U. Straccia, fuzzyDL: an expressive fuzzy description logic reasoner, in: *2008 International Conference on Fuzzy Systems (FUZZ-08)*, IEEE Computer Society, 2008, pp. 923–930.
- [46] F. Bobillo, U. Straccia, Fuzzy ontology representation using OWL 2, *Int. J. Approx. Reason.* 52 (7) (2011) 1073–1094. <http://dx.doi.org/10.1016/j.ijar.2011.05.003>.
- [47] J. Hirvonen, T. Tommila, A. Pakonen, C. Carlsson, M. Fedrizzi, R. Fullér, Fuzzy keyword ontology for annotating and searching event reports, in: *KEOD*, 2010, pp. 251–256.
- [48] F. Bobillo, M. Delgado, J. Gómez-Romero, DeLorean: a reasoner for fuzzy OWL 2, *Expert Syst. Appl.* 39 (1) (2012) 258–272. <http://dx.doi.org/10.1016/j.eswa.2011.07.016>. <<http://www.sciencedirect.com/science/article/pii/S095741741100978X>>.
- [49] M.C. Suárez-Figueroa, K. Dellschaft, E. Montiel-Ponsoda, B. Villazón-Terrazas, Z. Yufei, G.A. de Cea, A. García, M. Fernández-López, A. Gómez-Pérez, M. Espinoza, M. Sabou, NeOn Deliverable d5.4.1. NeOn Methodology for Building Contextualized Ontology Networks. NeOn Project, 2008. <<http://www.neonproject.org>>.
- [50] M. Poveda-Villalón, M. Suárez-Figueroa, A. Gómez-Pérez, Validating ontologies with OOPS!, in: A. Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, N. Hernandez (Eds.), *Knowledge Engineering and Knowledge Management, Lecture Notes in Computer Science*, vol. 7603, Springer, Berlin, Heidelberg, 2012, pp. 267–281. http://dx.doi.org/10.1007/978-3-642-33876-2_24.
- [51] SWRL: A Semantic Web Rule Language Combining OWL and RuleML. <<http://www.w3.org/submission/swrl/>>.
- [52] SPIN. <<http://www.spinrdf.org/>>.
- [53] P. Lukowicz, G. Pirkil, D. Bannach, F. Wagner, A. Calatroni, K. Förster, T. Holleczek, M. Rossi, D. Roggen, G. Tröster, J. Doppler, C. Holzmann, A. Riemer, A. Ferscha, R. Chavarriaga, Recording a complex, multi modal activity data set for context recognition, in: *Workshop on Context-Systems Design, Evaluation and Optimisation at ARCS*, 2010. <<http://www.duslab.de/cosdeo/>>.
- [54] R. Helaoui, D. Riboni, H. Stuckenschmidt, A probabilistic ontological framework for the recognition of multilevel human activities, in: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13*, ACM, New York, NY, USA, 2013, pp. 345–354. <http://doi.acm.org/10.1145/2493432.2493501>.
- [55] N. Díaz Rodríguez, J. Lilius, M.P. Cuéllar, M. Delgado Calvo-Flores, An approach to improve semantics in smart spaces using reactive fuzzy rules, in: W. Pedrycz, M.Z. Reformat (Eds.), *IFSA World Congress – NAFIPS Annual Meeting (IFSA/NAFIPS)*, IEEE, Edmonton, Canada, 2013, pp. 436–441.
- [56] J. Honkola, H. Laine, R. Brown, O. Tyrkkö, Smart-M3 Interoperability Platform Release, October 2009. <<http://sourceforge.net/projects/smart-m3/>>.
- [57] C. Wang, J. Lu, G. Zhang, Integration of ontology data through learning instance matching, in: *IEEE/WIC/ACM International Conference on Web Intelligence*, 2006. WI, 2006, pp. 536–539. <http://dx.doi.org/10.1109/WI.2006.100>.