

Combining Fuzzy Logic and Semantic Web to Enable Situation-Awareness in Service Recommendation

Alessandro Ciaramella¹, Mario G.C.A. Cimino²,
Francesco Marcelloni², and Umberto Straccia³

¹ IMT Lucca Institute for Advanced Studies, Piazza San Ponziano 6, 55100 Lucca, Italy
a.ciaramella@imtlucca.it

² Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni,
University of Pisa, Via Diotisalvi 2, 56122 Pisa, Italy
{m.cimino, f.marcelloni}@iet.unipi.it

³ Istituto di Scienza e Tecnologie dell'Informazione (ISTI - CNR), Pisa, Italy
straccia@isti.cnr.it

Abstract. Mobile Internet is rapidly growing and an enormous quantity of resources are currently available. Thus, the common mechanisms used up to now to locate resources, such as browsing and searching, do not look anymore to be effective in helping users in mobility. Indeed, the user's personal information space can be very large, with respect to the limited interaction capabilities of mobile devices. This paper proposes a situation-aware framework for providing personalized resources in a proactive manner. Current situations of the user are inferred by exploiting domain knowledge expressed in terms of ontologies and semantic rules, which are represented in the well-known Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL), respectively. Uncertainty in some contextual rule conditions is handled by defining appropriate linguistic variables through the Fuzzy Control Language (FCL), a standard representation of fuzzy systems for data exchange among different implementations, and adopting a purposely-adapted coding of ontologies and rules. Uncertain conditions bring to infer more than one situation with different certainty degrees: these degrees are used to assign a rank to concurrent situations. Finally, situations are connected to a set of related resources to be recommended to the user.

Keywords: Context-awareness; Fuzzy Inference System; Mobile Service Recommender; Web Ontology; Semantic Rules.

1 Introduction

Nowadays, the number of mobile users is rapidly growing and more and more services and documents are available on marketplaces. Categorizing such resources is a very difficult task: there exists no standardized taxonomy, and the user preferences can vary from an individual to another [9]. Hence, finding the best-fitting service by browsing and searching in resource repositories is very time expensive for the common user.

Moreover, mobile devices are becoming more and more capable to accommodate many types of resources such as services, documents and links to external sources, enlarging incredibly the personal information space of a mobile user. On the contrary,

the interaction features of the devices are very poor, due mainly to user interfaces less comfortable and manageable than in a personal computer. Furthermore, often services have to be configured with a set of proper parameters in order to be executed effectively. These parameters can vary in dependence of specific circumstances, and users have often to provide textual inputs on small keypads [6].

Hence, a significant cognitive effort is required to users in mobility to find and configure the most appropriate resources among the many available [8,15]. A tool able to automatically searching for the most suitable resources for the specific situation and configuring them would be strongly desirable. To this aim, in this paper we propose a situation-aware framework for providing personalized resources in a proactive manner. Situation awareness is enabled by a specific engine based on semantic web technologies and fuzzy logic.

More specifically, contextual information is maintained in the system by domain ontology [23] and is enriched with a truth degree depending on a level of certainty. Situations are inferred by means of semantic rules [25], which take the fuzziness of the contextual antecedents into account, and are ranked depending on their fuzzy values. Unlike in our previous work [1], here fuzziness is directly managed within the semantic rules and the semantic inference engine rather than by a specific fuzzy inference engine. These situations allow the identification of specific tasks, on the basis of domain knowledge expressed in terms of task ontology, which represents common sense knowledge about user usual activities. Finally, the specific current task together with contextual information is used to recommend a set of resources, in a task-navigation paradigm [7], where the user is supported to find appropriate services and documents by relying on the task ontology.

2 Background

A large amount of research in recommendation systems has focused on providing personalized services [10]. In our approach, the current user situation is exploited to narrow down the set of recommendable resources, thus showing to the user only those resources that are relevant in the specific situation. In [26], the term *situation* refers to a business level concept that allows targeting precisely and at different levels of granularity the demand of the user at a certain time. More precisely, a situation can be defined as a collection of context information that is invariant as long as the situation occurs [26]. According to Dey and Abowd [4], context refers to any relevant information that can be used to characterize a user. Hence, a situation can be inferred as a consequence of rules that verify whether a set of contextual conditions hold in the system. For instance, the situation *meeting* can be inferred when the contextual conditions *user is stationary*, *user is located in the scheduled place at the scheduled time*, and *user is close to the meeting organizer* are verified in the system.

Several projects consider the use of ontology as a promising means to develop context-aware systems. In the framework of semantic web, an ontology is a knowledge model that describes a domain of interest using semantic aspects and structure. An ontology consists of: (i) facts representing explicit knowledge, consisting of concepts, their properties, and instances that represent entities described by concepts; (ii) axioms

and predicates representing implicit knowledge, by means of rules used to add semantics and to derive knowledge from facts [26]. In [21], a comparative analysis shows that the most promising assets for context modeling in ubiquitous computing environments can be found in the use of ontology.

To reflect the varying nature of context and to ensure a universal applicability of context-aware systems, context is typically represented at different levels of abstraction [15]. At the first level of raw context sources there are context data coming from sensor devices, or user applications. At this level, logic embodied in semantic languages does not allow a treatment of uncertainty and imprecision existent in real world [17]. For instance, a typical smartphone GPS receiver provides a device position with dynamic accuracy ranging from some meters to hundreds of meters, depending on many environmental variables. Thus, the situation recognition from the environment should rely on a vague characterization. Furthermore, these situations are often connected to specific user requirements, and then the system should offer a specification mechanism that is intuitive, for instance in terms of standardized natural language, as guaranteed by employing linguistic terms [27].

3 Overall Architecture

In our implementation, the situation-aware resource recommender is running on the mobile device as an advanced menu, whose elements are dynamically updated, according to the different situations in which the user is involved. The overall system architecture is shown in Fig. 1.

In the server side, the main module is represented by the *situation engine*, which is in charge of interpreting contextual conditions and assessing the user situations. Contextual conditions that are inherently vague, such as mobility and proximity state of users, are evaluated by means of fuzzy logic, i.e., enriched with a truth degree maintained in the ontology. Such degrees represent the extent to which the conditions hold in the system. For instance *the user is close to a place* is a contextual condition that can be characterized with a truth degree representing the level of closeness of the user to the place. Semantic rules enhanced with the ability of managing the uncertainty allow inferring multiple situations with an appropriate ranking. This allows the system to recommend the related resources with different priorities.

The control flow of the application is steered by the *application controller* module, which manages the activities of each module, granting access to different functions and data sources. The *contextual data sources* package comprises a set of interfacing modules for different data sources, such as geographical maps, users' personal calendars and positions. In particular, numerical data concerning users positions are fed by the *location detector* module. This module provides outdoor/indoor location estimation, also on the basis of several possible technologies, such as GPS, GSM, WiFi [22]. Regardless of the available technologies, the *location detector* provides a generalized interface in terms of position and accuracy.

The *Rule Translator* is an off-line module that translates the rules, expressed in a high-level language, into a well-established standard for semantic rules, the Semantic Web Rule Language (SWRL, [25]). Thus, designers can express how the system should

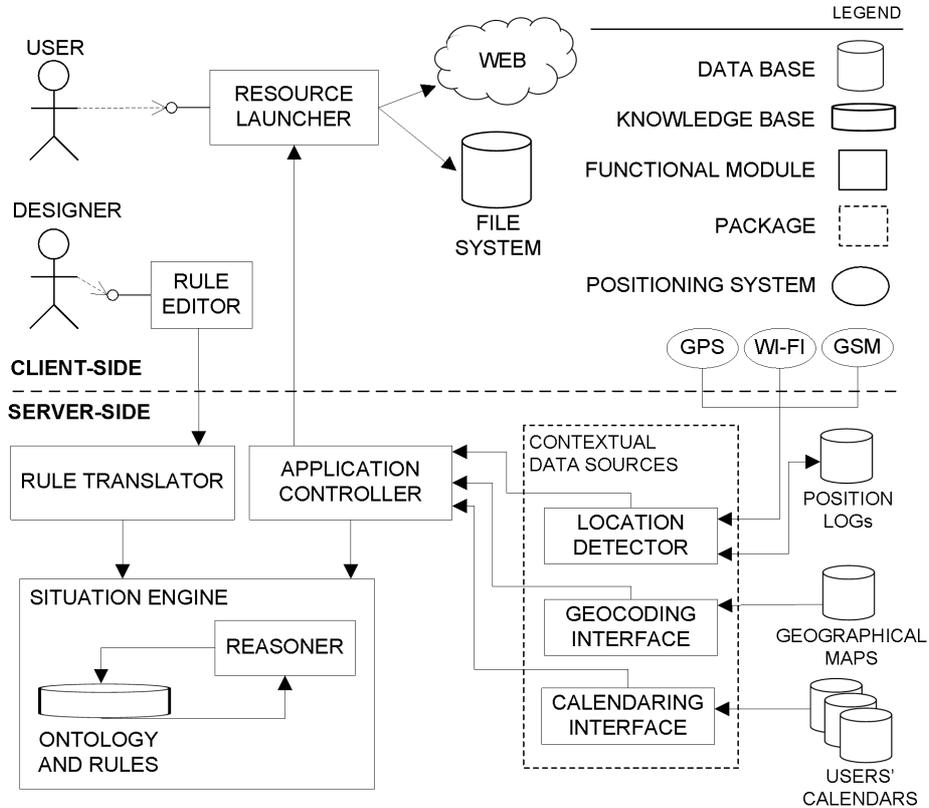


Fig. 1. Overall architecture of the situation-aware resource recommender

interpret contextual conditions in order to assess the most appropriate situations in a natural language close to their language. Further, the *Rule Translator* module allows the representation of the fuzzy logic within the SWRL, mapping directly the fuzzy information into the crisp ontology.

On the client side, the *Rule Editor* module allows a designer to configure and express the semantic rules for situation assessment. Finally, the *Resource Launcher* module shows the recommended resources to the user and allows the launch of these resources. In the following, the paper is focused on the design of the *situation engine* module.

4 Semantic Domain Knowledge

In the system, domain and general knowledge is represented by the situation ontology and related semantic rules. The ontology has been developed by using the Web Ontology Language (OWL [23]), a W3C standard well-supported in most semantic engines. In the upper situation ontology, general context information is represented by basic concepts such as *User*, *Calendar*, *Device*, *Time* and *Place*.

In order to manage fuzzy information in an OWL compliant ontology, we established a representation pattern. The pattern is applicable to properties that are related to the same base variable and to the same pair of concepts. For instance, let us consider the base variable *distance*, and the concepts *User* and *Place*. Depending on the actual value of the distance, and considering a prefixed set of distance intervals, we can establish properties like *User is-close-to a Place* or *User is-far-from a Place*. The presence of each property depends on the membership of the distance value to a prefixed interval. For example, considering the first interval as *LowDistance = 0-10 meters*, it can be said that *User is-close-to* depends on *LowDistance*, more formally *is – close – to*_{*LowDistance*}. Fig. 2-a shows an abstract representation of this mechanism, for a series of *n* properties and related *n* intervals. Here, concepts have been enclosed in oval shapes, whereas properties are represented by arrows. In order to capture vagueness in this representation, we propose the extension shown in Fig. 2-b. Here, an OWL group of properties is transformed into a concept, which includes a specification of the degree for each property. In other words, we assert that there is a property with a certain degree. Each degree is the membership level of the base variable to a specific fuzzy set.

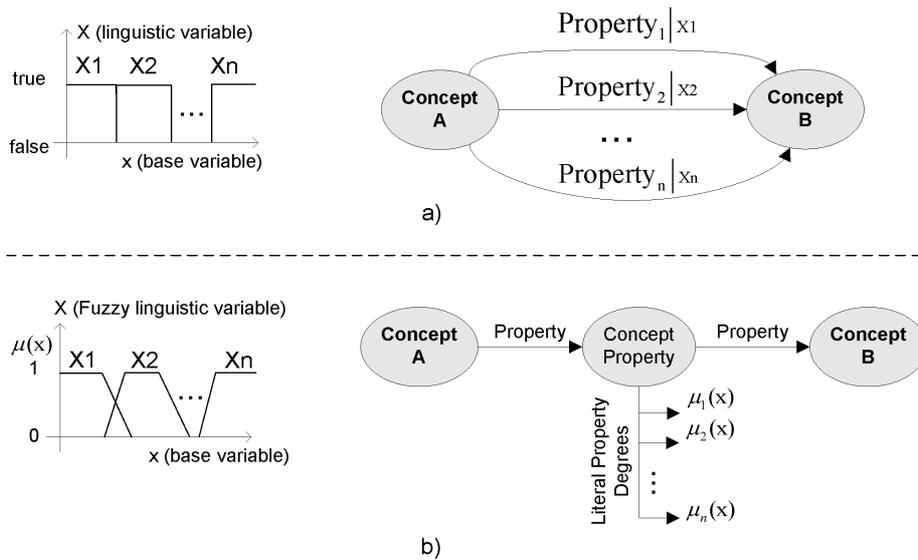


Fig. 2. An OWL-compliant fuzzy extension of a property

It is worth noting that this scheme can be used also in case of a property related to a single concept. In such case, the concept property corresponds to the concept itself.

In Fig. 3, the complete upper situation ontology is presented. This ontology is made of 10 general concepts and 25 properties, together with 5 concepts and 14 properties for the fuzzy representation. In particular, general concepts such as *Time* and *Place* are inherited from publicly available ontologies [24,5], according to the best practices of reusing domain ontologies. In the figure, such external ontologies are enclosed in

dashed rectangular shapes. Concepts are connected by properties, represented with directed black edges in the figure. Edges with white arrowhead show classical inheritance (i.e., an is-a relation).

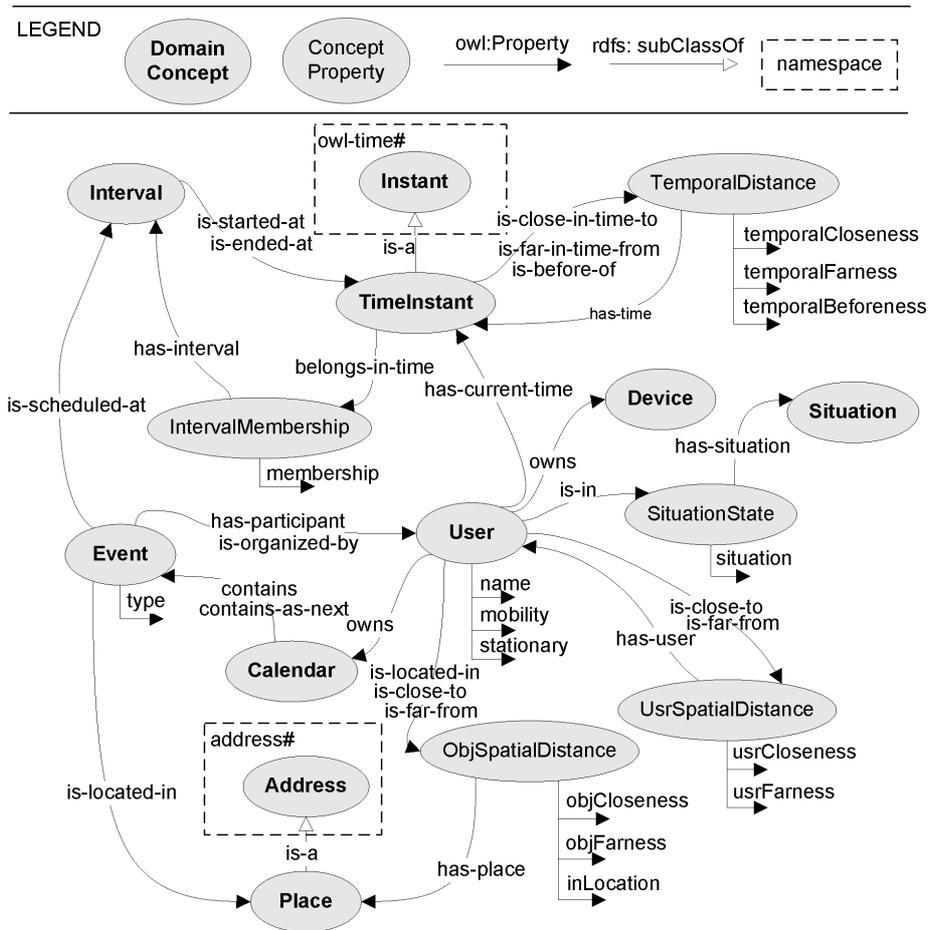


Fig. 3. The upper situation ontology

The model comprises a set of rules to infer the current situations on the basis of the situation ontology. Rules are expressed in the Semantic Web Rule Language SWRL, an emerging standard that extends OWL with additional rule-based knowledge representation. In terms of expressiveness, this reasoning standard corresponds to description logics, a particular decidable fragment of first order logic, and is named OWL DL [5]. Fig. 4 shows an example of rule in human readable syntax (a), commonly used in the literature, and in natural language (b). We point out that there are two types of antecedent

conditions, i.e., crisp (binary) and fuzzy, represented in Fig. 4 in bold and italic bold, respectively. The conditions *is a participant* and *has type* are derived from the user's calendar, and are inherently crisp, whereas the other conditions can be assessed only with vagueness. This implies that also the conclusion inferred from the rule is characterized by vagueness. This vagueness can be represented directly in SWRL (see next sections), which implements some mechanisms to express truth degrees and related membership functions.

<pre>owns(?user1, ?aCalendar) ^ contains-as-next(?aCalendar, ?nextEvent) ^ is-scheduled-at(?nextEvent, ?anInterval) ^ is-started-at(?anInterval, ?aTime) ^ mobility(?user1, ?user1mobility) ^ has-current-time(?user1, ?user1Time) ^ is-before-of(?user1Time, ?temporalDistance) ^ has-time(?temporalDistance, ?aTime) ^ type(?nextEvent, "business") ^ Pre-Meeting-on-Movement(?aSituation) ⇒ is-in-a-situation(?user1,?aSituation)</pre> <p style="text-align: center;">(a)</p>
<pre>IF <i>user1</i> IS A PARTICIPANT to the scheduled event AND <i>user1</i> IS moving AND <i>user1Time</i> IS BEFORE the scheduled event start-time AND event HAS TYPE business THEN <i>user1</i> IS IN A SITUATION OF pre-meeting-on-movement</pre> <p style="text-align: center;">(b)</p>

Fig. 4. A rule example

Once some situations have been inferred, with a certainty degree, a task ontology allows connecting a situation to specific tasks, and then specific tasks to specific resources to be recommended. Furthermore, such resources are tailored by proper contextual information, selected according to the identified user task. In Fig. 5 the upper task ontology is represented.

5 Managing the Uncertainty

There is some uncertainty in many contextual conditions related to real-world events. For instance, the condition *user1 is before the scheduled event start-time*, in Fig. 4.b, can be assessed only with a certainty degree. This uncertainty can arise, for instance, from lack of precision in the information stored in the users calendar. Furthermore, it is possible that noise affects sensed data. For instance, the condition *user1 is moving* requires an estimation of the user's speed, often known only with a limited accuracy.

Fuzzy set theory and fuzzy logic have proved to be a promising approach to manage the natural uncertainty that affects such contextual data [16]. In order to evaluate

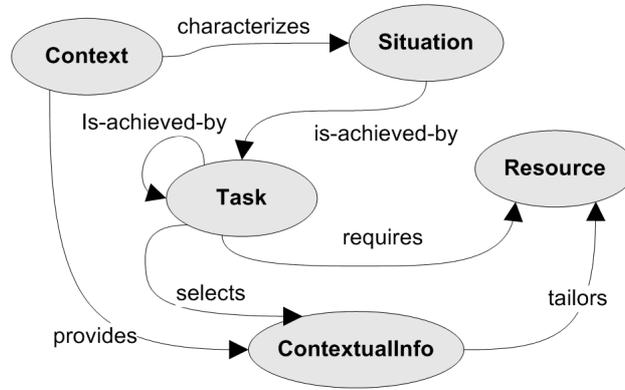


Fig. 5. The upper task ontology

the certainty degree of the contextual conditions, a number of linguistic variables have been defined. The universe of definition of such variables is partitioned with trapezoidal membership functions. An appropriate tuning of these functions has been carried out by means of experimental data.

Linguistic variables have been described using the Fuzzy Control Language (FCL, [2]), a standard representation of fuzzy systems for data exchange among different implementations. An example of the linguistic variable *speed*, used to decide about the user mobility, is shown in Fig. 6.

6 A Simple Integration of Fuzzy Logic into SWRL

In our implementation, we expressed fuzzy rules, such as the one described in Fig. 4-b, within SWRL, which however does not directly support fuzzy rules. While we refer the interested reader on fuzzy extensions of the logics behind Semantic Web Languages to [14,19,20], here we show that there is a simple way to encode the fuzzy rules into a crisp rule language supporting arithmetic built-in functions and, thus, in SWRL, making them directly available in current reasoners and in the Protégé editor¹. In fact, we followed the below mentioned method to correctly deal with our fuzzy rule base.

In our setting, a *fuzzy rule* is of the form (which closely resembles [13])

$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}. R_1(\mathbf{z}_1)[s_1], \dots, R_l(\mathbf{z}_l)[s_l], s = f(s_1, \dots, s_l)$$

where

1. R is an n -ary relation, every R_i is an n_i -ary relation;
2. \mathbf{x} are the *distinguished variables*.
3. \mathbf{y} are existentially quantified variables called the *non-distinguished variables*. We omit to write $\exists \mathbf{y}$ when \mathbf{y} is clear from the context;

¹ <http://protege.stanford.edu/>

```

FUNCTION_BLOCK speed
VAR_INPUT
    speed : REAL;
END_VAR
FUZZIFY speed
    TERM low := (0,1) (0,1) (3,1) (5,0);
    TERM high := (0,0) (10,1) (300,1) (300,1);
END_FUZZIFY
END_FUNCTION_BLOCK
    
```

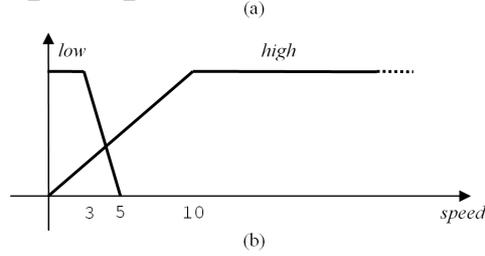


Fig. 6. Definition of linguistic labels in FCL

4. $\mathbf{z}_i, \mathbf{z}'_j$ are tuples of constants or variables in \mathbf{x} or \mathbf{y} ;
5. s, s_1, \dots, s_l are distinct variables and different from those in \mathbf{x} and \mathbf{y} , called *scores* or *truth degrees*;
6. f is a *scoring* total function $f: [0, 1]^l \rightarrow [0, 1]$, which combines the scores of the l relations $R_i(\mathbf{c}'_i)$ into an overall *score* to be assigned to the rule head $R(\mathbf{c})$. We assume that f can be computed in finite time.

We call $R(\mathbf{x})[s]$ the *head*, $\exists \mathbf{y}. R_1(\mathbf{z}_1)[s_1], \dots, R_l(\mathbf{z}_l)[s_l]$ the *body* and $s = f(s_1, \dots, s_l)$ the *scoring atom*. We also allow the scores $[s], [s_1], \dots, [s_l]$ and the scoring atom to be omitted. In this case we assume the value 1 for s_i and s instead. The informal meaning of such a rule is: if \mathbf{z}_i is an instance of R_i to degree at least or equal to s_i , then \mathbf{x} is an instance of R to degree at least or equal to s , where s has been determined by the scoring atom, *i.e.* $s = f(s_1, \dots, s_l)$.

As an example, in the following we show the high-level encoding of the fuzzy rule in Fig. 4-b:

```

is-in-a-situation(?user1, ?aSituation)[s] ← owns(?user1, ?aCalendar),
    contains-as-next(?aCalendar, ?nextEvent),
    is-located-in(?nextEvent, ?aPlace),
    is-scheduled-at(?nextEvent, ?anInterval),
    is-started-at(?anInterval, ?aTime),
    mobility(?user1, 'moving')[s1],
    has-current-time(?user1, ?userTime),
    is-before-of(?userTime, ?aTime)[s2],
    type(?nextEvent, 'business'),
    Pre-Meeting-on-Movement(?aSituation),
    s = min(s1, s2)
    
```

(1)

Note that the final degree s of being in a “pre-meeting-on-movement” situation is determined by the minimum of the users’ degree of being moving (s_1) and being before a meeting (s_2) (here, “moving” and “before” are considered fuzzy concepts). So, here the scoring combination function f is the minimum, which is also the function used in all the rules we have developed in our specific application. Of course, other functions can be used as well such as any so-called *t-norm* (used to combine conjunctive information) [12].

A rule base \mathcal{R} is a finite set of fuzzy rules, which we assume to be *acyclic*. This latter notion is defined as follows: we say that a relation R *directly uses* a relation R' if there is a rule in \mathcal{R} having R as head and R' occurring in its body. Let *uses* be the transitive closure of the relation “directly uses”. Then we say that \mathcal{R} is *acyclic* iff for any relation R it is not the case that R uses R . Please note that acyclicity is required to guarantee decidability. Note that cyclic rules bases can be allowed if specific conditions are met on the score combination functions (see *e.g.*, [18], for more on this issue), but we do not address them here.

We point out that we may represent a fuzzy rule in a succinct way as

$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})[s],$$

where $\phi(\mathbf{x}, \mathbf{y})[s]$ is

$$R_1(\mathbf{z}_1)[s_1], \dots, R_l(\mathbf{z}_l)[s_l], s = f(s_1, \dots, s_l).$$

We also impose that a rule base \mathcal{R} is such that there are no two rules in it with the same head. Note that this restriction is harmless. Indeed, in case we would like to have n rules with same head², *i.e.*

$$\begin{aligned} R(\mathbf{x})[s] &\leftarrow \exists \mathbf{y}_1. \phi_1(\mathbf{x}, \mathbf{y}_1)[s_1] \\ R(\mathbf{x})[s] &\leftarrow \exists \mathbf{y}_2. \phi_2(\mathbf{x}, \mathbf{y}_2)[s_2] \\ &\vdots \\ R(\mathbf{x})[s] &\leftarrow \exists \mathbf{y}_n. \phi_n(\mathbf{x}, \mathbf{y}_n)[s_n] \end{aligned}$$

then we may replace them with the $n + 1$ rules

$$\begin{aligned} R_1(\mathbf{x})[s] &\leftarrow \exists \mathbf{y}_1. \phi_1(\mathbf{x}, \mathbf{y}_1)[s_1] \\ R_2(\mathbf{x})[s] &\leftarrow \exists \mathbf{y}_2. \phi_2(\mathbf{x}, \mathbf{y}_2)[s_2] \\ &\vdots \\ R_n(\mathbf{x})[s] &\leftarrow \exists \mathbf{y}_n. \phi_n(\mathbf{x}, \mathbf{y}_n)[s_n] \\ R(\mathbf{x})[s] &\leftarrow R_1(\mathbf{x})[s_1], \dots, R_n(\mathbf{x})[s_n], s = g(s_1, \dots, s_n) \end{aligned}$$

where R_1, \dots, R_n are new relation symbols, and g specifies how to combine the scores of the individual rules into one overall score to be assigned to R . Usually, $g(s_1, \dots, s_n) = \max(s_1, \dots, s_n)$, but in general, any so-called *s-norm* [12] (used to combine

² In our specific fuzzy rule base, we do not have this scenario, though we present how to deal with it as it works generally.

disjunctive information) may be appropriate as well. This transformation guarantees then that \mathcal{R} remains acyclic and that there are no two rules in it with same head.

It remains to show how to represent fuzzy rules in a crisp rule language, which however supports arithmetic built-in predicates to perform arithmetic operations. To this end we proceed as follows.

1. Any n -ary relation R becomes an $n + 1$ -ary relation. The additional slot is used to store the score s . So, in any rule, an expression $R(\mathbf{z})[s]$ is replaced with the predicate $R(\mathbf{z}, s)$. For instance,

$$\text{is-before-of}(\text{?userTime}, \text{?aTime})[s2]$$

becomes

$$\text{is-before-of}(\text{?userTime}, \text{?aTime}, s2).$$

2. As our crisp rule language supports arithmetic built-in predicates, there is a way to express a rule

$$P_f(s_1, \dots, s_l, s) \leftarrow \text{built-in}(s = f(s_1, \dots, s_l))$$

which defines a predicate $P_f(s_1, \dots, s_l, s)$ such that $s = f(s_1, \dots, s_l)$, using the built-in arithmetic operations of the rule language.

3. Now, we replace each rule

$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}. R_1(\mathbf{z}_1)[s_1], \dots, R_l(\mathbf{z}_l)[s_l], s = f(s_1, \dots, s_l)$$

with the crisp rule

$$R(\mathbf{x}, s) \leftarrow \exists \mathbf{y}. R_1(\mathbf{z}_1, s_1), \dots, R_l(\mathbf{z}_l, s_l), P_f(s_1, \dots, s_l, s)$$

which concludes the case in which the rule language supports n -ary predicates. For instance, fuzzy rule (1) becomes

$$\text{min}(s1, s2, s3) \leftarrow \text{built-in}(s3 = \text{min}(s1, s2))$$

$$\begin{aligned} \text{is-in-a-situation}(\text{?user1}, \text{?aSituation}, s) \leftarrow & \text{owns}(\text{?user1}, \text{?aCalendar}), \\ & \text{contains-as-next}(\text{?aCalendar}, \text{?nextEvent}), \\ & \text{is-located-in}(\text{?nextEvent}, \text{?aPlace}), \\ & \text{is-scheduled-at}(\text{?nextEvent}, \text{?anInterval}), \\ & \text{is-started-at}(\text{?anInterval}, \text{?aTime}), \\ & \text{mobility}(\text{?user1}, \text{'moving'}, s1), \\ & \text{has-current-time}(\text{?user1}, \text{?userTime}), \\ & \text{is-before-of}(\text{?userTime}, \text{?aTime}, s2), \\ & \text{type}(\text{?nextEvent}, \text{'business'}), \\ & \text{Pre-Meeting-on-Movement}(\text{?aSituation}), \\ & \text{min}(s1, s2, s) \end{aligned}$$

However, SWRL is a rule language supporting unary and binary predicates only. This is not a particular problem, as to this end, we may rely on a well-known procedure, called *reification*³ (see also [3]), which allows to represent an n -ary relation via unary and binary relations. So, for instance, for the relation

$$\text{is-before-of}(\text{?userTime}, \text{?aTime}, s2)$$

we create a new class

$$\text{is-before-ofRelation}(\text{?aTimeReification})$$

with two additional properties

$$\begin{aligned} &\text{is-before-ofValue}(\text{?aTimeReification}, \text{?aTime}) \\ &\text{is-before-ofDegree}(\text{?aTimeReification}, s2) \end{aligned}$$

and, thus, $\text{is-before-of}(\text{?userTime}, \text{?aTime}, s2)$ will be replaced with

$$\begin{aligned} &\text{is-before-of}(\text{?userTime}, \text{?aTimeReification}), \\ &\text{is-before-ofValue}(\text{?aTimeReification}, \text{?aTime}), \\ &\text{is-before-ofDegree}(\text{?aTimeReification}, s2) . \end{aligned}$$

This allows removing n -ary ($n \geq 3$) relations from the rules bodies.

Concerning a n -ary ($n \geq 3$) relation in the rule head, such as

$$\text{is-in-a-situation}(\text{?user1}, \text{?aSituation}, s)$$

as before, we create a new class

$$\text{is-in-a-situationRelation}(\text{?aSituationReification})$$

with two additional properties

$$\begin{aligned} &\text{is-in-a-situationValue}(\text{?aSituationReification}, \text{?aSituation}) \\ &\text{is-in-a-situationDegree}(\text{?aSituationReification}, s) \end{aligned}$$

then add

$$\begin{aligned} &\text{is-in-a-situation}(\text{?user1}, \text{?aSituationReification}), \\ &\text{is-in-a-situationValue}(\text{?aSituationReification}, \text{?aSituation}) \end{aligned}$$

to the rule body and replace the head with

$$\text{is-in-a-situationDegree}(\text{?aSituationReification}, s)$$

For instance, our fuzzy rule about pre-meeting becomes in SWRL (here, the minimum is implemented as $\min(a, b) = (a + b - |a - b|)/2$), see [11]):

³ <http://www.w3.org/TR/swbp-n-aryRelations/>

```

min(s1, s2, s3)                                     ← sum(sm, s1, s2)
                                                    subtract(diff, s1, s2),
                                                    abs(absdiff, diff),
                                                    subtract(sd, sm, absdiff),
                                                    divide(s3, sd, 2),

is-in-a-situationDegree(?aSituationReification, s) ← owns(?user1, ?aCalendar),
                                                    contains-as-next(?aCalendar, ?nextEvent),
                                                    is-located-in(?nextEvent, ?aPlace),
                                                    is-scheduled-at(?nextEvent, ?anInterval),
                                                    is-started-at(?anInterval, ?aTime),
                                                    mobility(?user1, ?mobilityReification),
                                                    mobilityValue(?mobilityReification, 'moving'),
                                                    mobilityDegree(?mobilityReification, s1),
                                                    has-current-time(?user1, ?userTime),
                                                    is-before-of(?userTime, ?aTimeReification),
                                                    is-before-ofValue(?aTimeReification, ?aTime),
                                                    is-before-ofDegree(?aTimeReification, s2),
                                                    type(?nextEvent, 'business'),
                                                    Pre-Meeting-on-Movement(?aSituation),
                                                    is-in-a-situation(?user1, ?aSituationReification),
                                                    is-in-a-situationValue(?aSituationReification, ?aSituation),
                                                    min(s1, s2, s)

```

which concludes. We don't go further into the reification procedure as it is pretty common and well-known in the Semantic Web literature.

7 Conclusions and Future Work

In this paper, a framework for providing personalized resources based on situation awareness has been proposed, showing situation awareness as a key asset to enable proactive behavior in the system. In particular, the study focuses on combining semantic web standards with fuzzy logic. Domain knowledge is maintained by means of proper ontologies and exploited to infer the current user situations. Inference is carried out by semantic rules which embody fuzzy logic to take the assessment of real-world inaccurate information into account.

We applied the system in two real business cases, in order to assess the effectiveness of our approach. The first evaluation case study concerns a pharmaceutical consultant in typical business situations. In particular, the situations of interest are: (i) *Meeting-Planning*, (ii) *Pre-Meeting*, (iii) *Ongoing-Meeting*, (iv) *Post-Meeting*, (v) *Hospital-Conference*, (vi) *Call-for-Tenders*, and (vii) *Meal*. The second evaluation case study concerns an off-site student, who performs a daily travel to go to university and return. In this case, the identified situations are the following: (i) *Pre-University-Day*, (ii) *Preparing-for-Transportations*, (iii) *Traveling*, (iv) *Studying*, (v) *Attending-Courses*, and (vi) *Meal*. For each case study, by means of a series of interviews, a domain-specific ontology has been added to the upper ontology and the fuzzy linguistic variables have been tuned properly. Hence, a set of 13 rules has been designed for the above business cases. Finally, the system has been tested by considering the events concerning one overall working week of two different consultants and three different students, with 53 and 82 different test events, respectively. The system has been able to recognize the right situations related to all the test events under the different conditions. Further, the differences between the time at which actually the situation occurred and the time at which the system recognized the situation was in the range of few seconds to few minutes. This proves the usefulness of our service recommender which exploits data

collected by different sensors to determine the situations of interest with respect to a recommender based only on the scheduled events stored in the users calendar.

A weakness of the system concerns the design of the linguistic variables, which is domain-specific and does not take into account actual differences among users. We are currently working on improving the possibility of adaptation of the system to the specific user. We are focusing on the exploitation of the user's profile, expressed in terms of user's preferences. Further, we are considering the application of machine learning paradigms for the tuning of the linguistic variables.

References

1. Ciaramella, A., Cimino, M.G.C.A., Lazzarini, B., Marcelloni, F.: Situation-aware mobile service recommendation with fuzzy logic and semantic web. In: IEEE International Conference on Intelligent Systems Design and Applications (ISDA'09), pp. 1037–1042 (2009)
2. Cingolani, P.: Jfuzzylogic, a java package that implement fuzzy control language (fcl) specification (iec 1131p7), <http://jfuzzylogic.sourceforge.net>
3. Dahchour, M., Pirotte, A.: The semantics of reifying n-ary relationships as classes. In: 5th International Conference on Enterprise Information Systems (ICEIS-02), pp. 580–586 (2002)
4. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. In: Workshop on the What, Who, Where, When, and How of Context-Awareness, pp. 304–307. ACM Press, New York (2000)
5. Ding, L., Chen, H., Kagal, L., Finin, T.: Public address ontology, <http://dam1.umbc.edu/ontologies/ittalks/>
6. Figge, S.: Situation-dependent services a challenge for mobile network operators. *Journal of Business Research* 57(12), 1416–1422 (2004)
7. Fukazawa, Y., Naganuma, T., Fujii, K., Kurakake, S.: A framework for task retrieval in task-oriented service navigation system. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 876–885. Springer, Heidelberg (2005)
8. Goix, L.W., Valla, M., Cerami, L., Falcarin, P.: Situation inference for mobile users: a rule based approach. In: The 2007 International Conference on Mobile Data Management (MDM'07), pp. 299–303. IEEE Computer Society, Los Alamitos (2007)
9. Heinonen, K., Pura, M.: Classifying mobile services. Helsinki Mobility Roundtable. Working Papers on Information Systems, p. 6 (2006)
10. Hong, J., Suh, E.H., Kim, J., Kim, S.Y.: Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications* 36(4), 7448–7457 (2009)
11. Kalman, D.: The maximum and minimum of two numbers using the quadratic formula. *The College Mathematical Journal* 15(4), 329–330 (1984)
12. Klement, E.P., Mesiar, R., Pap, E.: *Triangular Norms*. Trends in Logic - Studia Logica Library. Kluwer Academic Publishers, Dordrecht (2000)
13. Lukasiewicz, T., Straccia, U.: Top-k retrieval in description logic programs under vagueness for the semantic web. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 16–30. Springer, Heidelberg (2007)
14. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics* 6, 291–308 (2008)
15. Luther, M., Fukazawa, Y., Wagner, M., Kurakake, S.: Situational reasoning for task-oriented mobile service recommendation. *The Knowledge Engineering Review* 23(01), 7–19 (2008)
16. Mantjarvi, J., Seppanen, T.: Adapting applications in handheld devices using fuzzy context information. *Interacting with Computers* 15(4), 521–538 (2003)

17. Sanchez, E., Yamanoi, T.: Fuzzy ontologies for the semantic web. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 691–699. Springer, Heidelberg (2006)
18. Straccia, U.: Uncertainty management in logic programming: Simple and effective top-down query answering. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 753–760. Springer, Heidelberg (2005)
19. Straccia, U.: Managing uncertainty and vagueness in description logics, logic programs and description logic programs. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) Reasoning Web. LNCS, vol. 5224, pp. 54–103. Springer, Heidelberg (2008)
20. Straccia, U.: A minimal deductive system for general fuzzy RDF. In: Polleres, A. (ed.) RR 2009. LNCS, vol. 5837, pp. 166–181. Springer, Heidelberg (2009)
21. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Workshop on Advanced Context Modelling, Reasoning and Management (UbiComp'04), Citeseer (2004)
22. Sun, G., Chen, J., Guo, W., Liu, K.J.R.: Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs. *IEEE Signal Processing Magazine* 22(4), 12–23 (2005)
23. W3C.Owl web ontology language reference (February 10, 2004), <http://www.w3.org/TR/owl-ref/>
24. W3C.Public time ontology, <http://www.w3.org/TR/owl-time>
25. W3C.SWRL: a semantic web rule language combining OWL and RuleML, W3C member, <http://www.w3.org/Submission/SWRL/>(submission May 21, 2004)
26. Weißenberg, N., Gartmann, R., Voisard, A.: An ontology-based approach to personalized situation-aware mobile service supply. *Geoinformatica* 10(1), 55–90 (2006)
27. Zadeh, L.A.: Is there a need for fuzzy logic? *Information Sciences* 178(13), 2751–2779 (2008)