**IMT Institute for Advanced Studies, Lucca**

Lucca, Italy

# Situation Awareness in Mobile Recommendation Systems

PhD Program in Computer Science and Engineering

XXIII Cycle

**By**

**Alessandro Ciaramella**

**2011**

## Contents

## List of Figures

## List of Tables

## Publications

A. Ciaramella, M.G.C.A. Cimino, B. Lazzerini, F. Marcelloni, "*A Situation-Aware Resource Recommender Based on Fuzzy and Semantic Web Rules*", in International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS), World Scientific, Vol. 18 No 4. pp.411-430, 2010 (ISSN 0218-4885).

A. Ciaramella, M.G.C.A. Cimino, B. Lazzerini, F. Marcelloni, "*Using Context History to Personalize a Resource Recommender via a Genetic Algorithm*", Proc. IEEE International Conference on Intelligent Systems Design and Applications (ISDA'10), pp. 965-970, Cairo, Egypt, 2010 (ISBN 978-1-4244-8136-1).

A. Ciaramella, M.G.C.A.Cimino, F. Marcelloni, U. Straccia, "*Combining Fuzzy Logic and Semantic Web to Enable Situation-Awareness in Service Recommendation*", Proc. LNCS 21st International Conference on Database and Expert Systems Applications (DEXA '10), pp 31-45, LNCS 2011, Vol 6261, Bilbao, Spain, 2010 (ISBN 978-3-642-15363-1).

A. Ciaramella, M.G.C.A. Cimino, B. Lazzerini, F. Marcelloni, "*Situation-aware Mobile Service Recommendation with Fuzzy Logic and Semantic Web*", Proc. IEEE International Conference on Intelligent Systems Design and Applications (ISDA'09), pp. 1037-1042, Los Alamitos, CA (USA), 2009 (ISBN 978-0-7695-3872-3).

A. Ciaramella, M.G.C.A. Cimino, B. Lazzerini, F. Marcelloni, "*Using BPMN and Tracing for Rapid Business Process Prototyping Environments*" Proc. INSTICC International Conference on Enterprise Information Systems Conference on e-Commerce, e-Business, and e-Government (ICEIS 2009), pp 206-212, vol III, Milan, Italy, 2009 (ISBN 978-989-8111-86-9).

## Abstract

Nowadays, a huge quantity of resources for mobile users is made available on the most important marketplaces. Further, handheld devices can accommodate plenty of these resources, such as applications, documents and web pages, locally. Thus, to search for resources suitable for specific circumstances often requires a considerable effort and rarely brings to a completely satisfactory result. Moreover, mobile users are likely to devote only partial attention and time to the devices while using them, because the primary task is interacting with the reality, e.g. moving, chatting or even driving a car. A tool able to recommend suitable resources at the right time in each situation would be of great help for the mobile users and would make the use of the handheld devices less boring and more attractive. To this aim, new levels of granularity, together with some degree of self-awareness, are needed to assist mobile users in managing and using resources. Situation awareness can provide a powerful mechanism to identify the user needs at a certain time, enhancing the device usage. However, determining the correct user situation is not a trivial task, due to imperfect domain knowledge, uncertainty in data, and changing user behaviors.

In this thesis, we propose a situation-aware resource recommender, which helps mobile users to timely locate resources proactively. Situations are determined by a semantic reasoner that exploits domain knowledge expressed in terms of ontologies and semantic rules. This reasoner works in synergy with a fuzzy engine, which is in charge of handling the vagueness of some conditions in the semantic rules,

computing a certainty degree for each inferred situation. These degrees are used to rank the situations and consequently to assign a priority to the resources associated with the specific situations. Moreover, in order to adapt the situation recognizer to the specific user, the system collects data during the interaction of the user with the mobile device. This context history is exploited by genetic algorithms to learn user habits and adapt accordingly the meaning of the linguistic values used in the fuzzy engine.

The proposed framework is evaluated by means of real case studies concerning resource recommendations, and experimental results show the effectiveness of the approach.

# Introduction

In the last years, an enormous quantity of resources for mobile users has been made available on the most important marketplaces[1]. Such resources range from entertainment (games, songs, etc.) to information (weather forecast, traffic maps, news, etc.), from transactions (money transfer, airline reservation, etc.) to productivity (notepad, voice reader, etc.). The number and diversity of these resources make practically impossible for the average user to identify the most suitable service or application for a specific situation. Indeed, the users probably do not know the specific features of all the resources and therefore rarely can associate these resources with specific situations. Even if the users were conscious of which service is suitable for each specific situation, to manually retrieve them while the situation occurs might be very hard. Indeed, the standard categorization of resources, e.g. based on their functions, is often ineffective. Actually, mobile resources may belong to several categories depending on the use case and the user's individual preferences [HePu06].

Searching and browsing are the most used mechanisms to locate resources in repositories. Typically, due to the large number and variety of resources, a vast amount of time is required to find the most

[1] See for instance: *App Store* of Apple Inc. (www.apple.com/iphone/appstore), *Android Market* of Google Inc. (www.android.com/market), *Windows Mobile Catalog* of Microsoft Corp. (www.microsoft.com/windowsmobile/catalog), *Ovi Store* of Nokia Corp. (store.ovi.com), and *BlackBerry App World* of Research In Motion Limited (www.blackberry.com/appworld).

suitable one. Moreover, mobile users are likely to devote only partial attention and time to the devices while using them, because the primary task is interacting with the reality, e.g. moving, chatting or even driving a car [ChIK99]. Further, users in mobility are strongly limited in their search, using a hardware with reduced information presentation and interaction capabilities. Indeed, mobile devices feature small screens and miniaturized keypads in order to be portable and really handheld. Eventually, once installed, these resources have to be configured and launched with a set of proper parameters, which often vary in dependence of specific user circumstances [Figg04]. Thus, a significant cognitive effort is required to users in mobility to find and configure the most appropriate resources among the many available [LFWK08; GVCF07]. These users would considerably benefit from a system able to automatically recommend resources at the right time and for the specific context.

In the literature of mobile computing, the use of context information is introduced in terms of implicit input from changes in the environment [Hans06]. This model is usually referred to as context-awareness, because the output of the system depends on who is using the application, where, when, and in which situation. Designing context-aware applications involves two main steps: (i) designing a set of rules to infer high-level situations and (ii) designing proper input drivers to gather context information from the surrounding environment.

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 abstractions [LFWK08]. At the lowest level, which takes the raw context sources into account, there are contextual data coming from sensor devices and/or user applications. These contextual data are generally imprecise and vague. For instance, a typical smart phone GPS receiver provides a device position with dynamic accuracy ranging from some meters to hundreds of meters, depending on many environmental variables. Also, the time and location provided by the user's calendar are in practice ideal references only, because real events usually happen approximately at the referred time and place. Nevertheless, logic embodied in semantic languages does not allow managing uncertainty [SaYa06] and forces the resolution of uncertainty before the inference process. On the contrary, a situation recognizer should permit to express situations in terms of vague characterizations. Fuzzy logic has proved to be a very effective tool to manage uncertainty by using a very intuitive language [Zade08].

In the following, definition of context is discussed and a motivating scenario for the thesis is proposed. Finally, the outline and the structure of this document are presented.

## 1.1. Definition of Context

All human beings have a natural intuition of the term *"context"* and exploit it in everyday life to improve their communication skills. Communication is always enriched with a common context, like the current situation or a shared knowledge, which reduces the amount of information necessary to understand what is being talked about.

Computers, on the other hand, cannot understand information as human beings can. They only react following the way they are programmed and the inputs they get from the users. Moreover, these inputs have to be explicit even for the most trivial information such that the computer system can handle it. Users have to formulate their intent as a series of commands which the computer can understand, process, and produce output in response to [Hans06]. The reason of this shortage in the human-computer interaction is that, traditionally, the field of computer science has taken a position that is antithetical to the context problem: *the search for context-independence* [LiSe00]: the most common approach for all problems of computer science is treating the systems of interest as black boxes. Something goes in one side, something comes out of the other side, and the output is completely determined by the input (Figure 1a). Context-awareness, instead, aims to take into account the context in which the process takes place. Lieberman and Selker [LiSe00] claimed that the traditional interaction model of computer systems has to be extended in order to allow applications to decide what to do grounding not only on the explicitly presented input, but also on the context (Figure 1b). We arrive hence to a first definition of context [LiSe00]:

*"Context can be considered to be*
*everything that affects the computation*
*except the explicit input and output".*

This definition however is too intuitive and many problems arise when we try to define what explicit input is and what is not in a practical application. Much research has been done to understand better the concept of context, and in what follows we try to present a synthetic review.
The first idea of using context in applications is traditionally ascribed to the effort of the Palo Alto Research Center [Weis91; WHFG92] where the concept of context fell on the concept of location. Location indeed is the most important piece of information we can exploit to enrich the communication between users and computers, because, once this is known, there are many other pieces of information that can be inferred [Noki09]. For instance, knowing that a user is in her office can be exploited to infer that she is working. However, *"there is more to context than location"* [ScBG99] as Schmidt *et al.* claimed out in 1999. Some researchers defined context by enumerating the constituting parameters [Hans06], like *"location, time of day, season of the year, and temperature"* or *"information about the environment, such as location, time, temperature or user identity".* These definitions, anyway, are too specific to the particular application in which context is used, and they can vary from case to case. Hence, other researchers strove to define context in a broader manner.

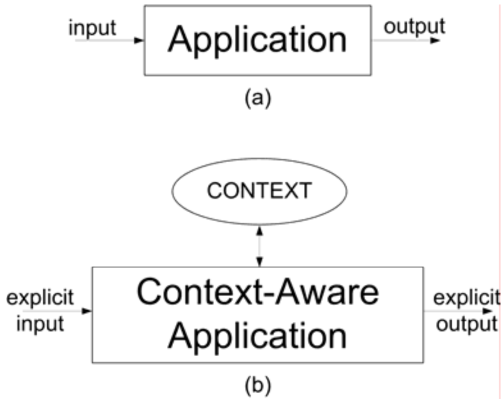



**Figure 1 – The traditional "black box" paradigm (a) versus the context-enriched one (b) [LiSe00].**

General definitions described context as *"the elements of the user's environment the computer knows about"*, *"the aspects of the user's local environment"*, *"the state of its [the context-aware application] surroundings"*, *"any environmental factor that might influence the activities of the computer, provided there is some mechanism for capturing it"* [Hans06].

Once more, these definitions are far to be satisfactory. They are too wide, and it's very difficult to model such context to be exploited in an application. Finally, two definitions have been presented to take into account both the general aspects of context ant its relevancy for the use the application makes of it. Chen and Kotz [ChKo00] in 2000 define context as:

*"Context is the set of environmental states and settings*
*that either determines an application's behavior*
*or in which an application event occurs and is interesting to the user".*

This definition highlights two different aspects of the context. The first one includes the characteristics of the surrounding environment that determine the behavior of the applications. The other aspect of context concerns all information that is relevant to the application, but not

critical. It is not necessary for applications to adapt to the second kind of context except to display them to interested users.
The other definition was proposed by Anind K. Dey in 2001 [Dey01]:

*"Context is any information*
*that can be used to characterize the situation of an entity.*
*An entity is a person, place, or object that is considered relevant*
*to the interaction between a user and an application,*
*including the user and applications themselves".*

We prefer this definition among all others because it is a very comprehensive and accurate definition, including all we can be experienced of when we think about context and, at the same time, narrowing us down to only relevant information. Context can be related to *"where you are, who you are with, and what resources are nearby, as well as to changes to such things over time"* [ScAW94], so it includes a very huge quantity of information, but we are interested only in the pieces of information relevant to characterize the situation of entities. For instance, the list of printers near a user is context if she is working with her word processor but it is not context if she is using her mobile phone to call someone. Moreover, this definition does not include only implicit input (as the first definition we provided above from the work of Lieberman and Selker [LiSe00]) but also explicit input. For instance, if the identity of the user is modeled as context for an application, this information may be supplied either implicitly via sensors or inference rules or explicitly when the user logs into the system. In both cases, the application can exploit the identity of the user to enhance the interaction process.

### 1.1.1. Context-Aware Systems

It is universally recognized that the first application based on context-awareness was the Olivetti Active Badge proposed by Want *et al.* [WHFG92] in 1992. This system exploits context information about employee's location to help receptionists to forward incoming calls towards the correct room.
An application can be considered as context-aware if:

*"it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task"*
(Dey and Abowd [DeAb00]).

This is a very general definition, encompassing both the systems that modify their behavior to auto-adapt to the context and also the systems that simply provide information about context to the user.
Dey and Abowd also list the features a context-aware system can provide: presentation of information and services to a user; automatic execution of a service; and tagging of context to information for subsequent retrieval. The latter feature allows linking a piece of information to a user's context like a virtual note.

## 1.2. Motivating Scenario

The number of available resources for mobile devices is continually growing. Mobile marketplaces host thousands of applications for all kinds of user needs. Moreover, mobile devices have increasing capability to store applications and documents, enlarging the personal information space of a mobile user in terms of dimensionality and variety. Thus, for an average user, finding the desired resource can be very time consuming.
A *resource recommender* is a software application which takes the current user situation into account, in order to recommend resources while user needs them. The front-end can be thought of as an intelligent

menu, whose items are automatically changing on the basis of a dynamic situational ranking, and whose parameters are automatically loaded on the basis of the context. For instance, a recommender can be useful for an off-site university student, who performs a daily travel to go to university and return. More specifically, let us consider the following excerpt of an off-site student scenario. *Alan, a university student, leaves his home in Lucca in the early morning, and catches the train to go to Pisa. While the train is arriving at the Lucca railway station, the student accesses the latest news. During the travel, Alan would like to revise the slides of the next lesson. While the train is arriving at the Pisa railway station, he checks the course timetable for knowing the classroom of the next lesson. When he is close to the classroom, he checks for classmate messages.*

In this scenario, a resource recommender should recognize, e.g., the situation *Traveling* when the user is moving towards the faculty and close in time to the scheduled lesson. According to the situation, the recommender should also propose specific resources (such as Alan's course timetables and his lesson slides) that are also implicitly parameterized in terms of the context (e.g., next appointment and current location). Following the example, after the situation *Traveling* has been recognized, the client interface of the recommender should appear as shown in Figure 3.

Hence, two main processes can be characterized: the *situation assessment* (SA) and the *resource recommendation* (RR), as represented in Figure 2.

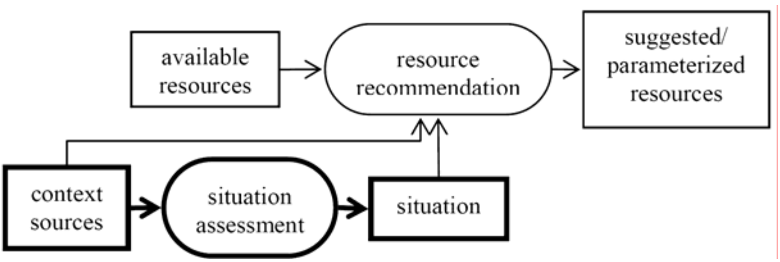



**Figure 2 – Macro-processes of a Resource Recommender.**

The RR process can be considered as a resource classifier which is modulated by both the user context and situation, whereas the SA process generates a higher level concept, i.e. the situation, starting from context sources. Further, the RR process is typical of recommenders, whereas the SA process can be modeled as a general purpose component of any situation aware application.


**Figure 3 – An example of the client interface, for the student case study.**

The term 'situation' is used as an abstraction of context, allowing a precise identification of the user demand at a certain time [Weiß06]. A situation can be represented in terms of collection of contextual information that does not change as long as the situation occurs. For instance, the situation *"attending-a-lesson"* can be inferred from a set of contextual parts such as *"user is located in classroom"*, *"user is stationary"*, *"user time belongs to a course schedule"*, and so on. Table 1 shows an excerpt of rules concerning the situation assessment of lesson-related events.

**Table 1. Excerpt of rules concerning the situation assessment of lessons-related events.**

| **PREMISE** | **CONSEQUENCE** |
|---|---|
| *If* user is moving<br>*And* user-time is before in time to the scheduled start-time of the lesson | Then Situation is Pre-Lesson on Movement |
| *If* user is stationary<br>*And* user-time belongs to the scheduled interval-time of the lesson<br>*And* user is located in the scheduled place of the lesson<br>*And* user is far from the other participants | Then Situation is Pre-Lesson |
| *If* user is stationary<br>*And* user-time belongs to the scheduled interval-time of the lesson<br>*And* user is located in the scheduled place of the lesson<br>*And* user is close to the other participants | Then Situation is On-Going-Lesson |

It is worth noting from this motivating scenario that context rules rely on domain ontology. For instance, the terms *moving, stationary, time, place, lesson,* and so on, need to be semantically defined within a domain knowledge, connected to real-world instances, allowing the above semantic rules to be expressed in a machine readable form, and evaluated for determining the current situation.

## 1.3. Outline of the Thesis

The rest of the thesis describes our research in detail and is organized as follows.

Chapter 2 (**Background technologies**) is divided in three main sections. In Section 2.1 the Semantic Web is presented with particular emphasis on its key technologies, i.e., ontologies and semantic rules. Section 2.2 recalls fundamentals of Fuzzy Set Theory, Fuzzy Logic and fuzzy systems as powerful tools to handle the intrinsic uncertainty in contextual data. Finally, in Section 2.3 Genetic Algorithms are presented with particular focus on Genetic Fuzzy Systems.

Chapter 3 (**Related Work in Context- and Situation-Awareness**) situates our work in the research field of context and situation-awareness. In particular, the first section proposes a survey of the main literature in sensing and exploring user situations, whereas the other section is narrowed in the field of recommendation systems.

In Chapter 4 (**A Mobile Resource Recommendation with Fuzzy Logic and Semantic Web**), a recommender system based on the user situation is proposed. Situations are determined by a semantic reasoner that exploits domain knowledge expressed in terms of ontologies and semantic rules. This reasoner works in synergy with a fuzzy engine, which is in charge of handling the vagueness of some conditions in the semantic rules, computing a certainty degree for each inferred situation. These degrees are used to rank the situations and consequently to assign a priority to the resources associated with the specific situations.

In Chapter 5 (**Combining Fuzzy Logic and Semantic Web**), an alternative manner to determine user situations is proposed, adopting a purposely-adapted coding of ontologies and rules. Hence, another recommender system is described, where fuzziness is directly managed within the semantic rules and the semantic inference engine rather than by a specific fuzzy inference engine.

Chapter 6 (**Adapting the Situation Recognition to the User Behavior**) shows how a Genetic Fuzzy System can be employed to better personalize a context-aware recommender. Data collected during the interactions of the user with the mobile device are used to build a context history. This context history is then exploited to personalize the recommender system by a genetic algorithm.

In Chapter 7 (**Evaluation Case Studies**), experimental results on real business cases are shown, validating the proposed approaches.

Finally, Chapter 8 (**Conclusions**) draws final conclusions. Moreover, we propose possible extension of the current work, hypothesizing future research trends on the study of situation-aware resource recommenders.

## Chapter 2

# Background Technologies

In this Chapter, we recall fundamentals on the main technologies that we have employed in our work. The first Section is devoted to present the Semantic Web, a collaborative effort led by W3C to provide a common framework that allows data to be shared and reused across multiple agents [Sema01]. In particular, two key technologies of the Semantic Web, widely used in this thesis, are presented: *ontologies* as the standard for representing and sharing knowledge among distribute agents, as well as the languages to express *semantic rules,* in order to enable complex reasoning among data.
The Section 2.2 introduces the Fuzzy Set theory and Fuzzy Logic, employed in this thesis as powerful tools to handle the intrinsic uncertainty in contextual data.
Finally, in Section 2.3 Genetic Algorithms are presented as a key technology to solve optimization problems by imitating the process of natural evolution. Genetic Algorithms have been widely employed to tune some parameters in fuzzy systems, in the so called Genetic Fuzzy Systems. In this thesis, a genetic fuzzy system is exploited to enhance personalization and adaptation to the user behavior.

## 2.1. The Semantic Web

As Berners-Lee *et al.* [BeHL01] announced, the traditional Web is not sufficient to allow real interoperations among computer systems, because its content is designed only for human beings. Programs are able to render Web pages in a very good graphical manner, but they cannot understand the underground meaning in any reliable way. The *Semantic Web* [BeHL01] aims to build a robust environment where data and information can be processed automatically by computer systems, with a shared understanding of information and a set of rules to reason and infer new knowledge. In [Bern01], the Semantic Web is conceived as a layered architecture where several enabler technologies and standards cooperate to support this vision. To illustrate the architecture of the Semantic Web, Berners-Lee proposed the Figure 4, called Semantic Web Stack.

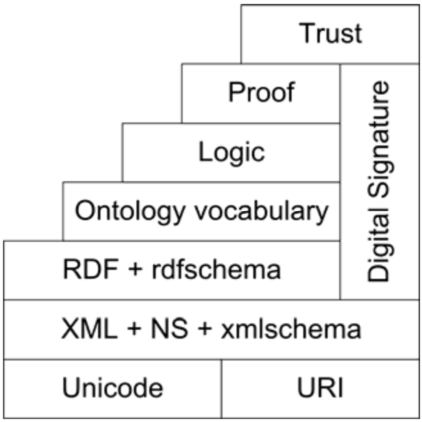



**Figure 4 – The Semantic Web Stack proposed by Berners-Lee.**

Identifying resources universally, URIs are fundamental to build a shared ground in which it is clear the entity each name refers to. XML enables a common way to structure and communicate data among heterogeneous systems. Resource Description Framework (RDF) [RDF04] provides a data model to express statement about the meaning of things by means of triples (subject, predicate, and object) and RDF Schema allows combining RDF statements for describing classes of resources and relationships among them.

Ontologies are *"a collection of definitions of concepts"* [BaHS05] and they can define more complex taxonomy than RDF Schema about classes of object and relationships among them. By means of ontologies, it is possible to make new inferences and inductions (the logic and proof layer) and eventually provide a trusted environment in which systems can interoperate (the highest layer).
In the following, ontologies are described with a particular reference to the standard language to author them (The Web Ontology Language, OWL) [OWL04]. Then, ontologies are presented as the key technology to model contextual information in context-aware systems. Finally, Semantic Rules are introduced to infer new knowledge from facts declared in ontologies, and the Semantic Web Rule Language (SWRL) [SWRL04] is described as the reference standard to express semantic rules.

### 2.1.1. Ontologies

The term *ontology* originates from Philosophy to describe the study of being as such. In particular, ontology refers to all entities that one can think of, and aims to describe their meanings and relationships. In the late seventies, the term was applied to Information Science to describe in a formal manner the knowledge about a specific domain. Gruber [Grub93] proposed a concise definition of an ontology as *an explicit specification of a conceptualization.* This definition has been further enriched by Borst [Bors97] in his PhD Thesis as following:

"An ontology is a *formal* specification of a *shared* conceptualization",

where (i) the *formal* specification allows the ontology to be processed and interpreted by software agents, and (ii) the *shared* conceptualization requires that at least a group of agents has to agree

on the meaning of each concept in the ontology, i.e., the ontology has to capture *consensual knowledge* [Sure03].
More specifically, an ontology consists of [Weiß06]:

- Facts representing *explicit* knowledge, consisting of concepts and their properties, and instances that represent entities described by concepts;

- Axioms and predicates representing *implicit* knowledge, by means of rules used to add semantics and to derive knowledge from facts on demand.

The advantages of using ontologies can be summarized in two features [DKD+05]: (i) data has a shared and clear meaning among people or software agents; (ii) the semantic description of data can be used to infer new knowledge. Moreover, ontologies enable reuse of domain knowledge. Reuse is one of the principles of programming methods, and it is also valid for knowledge, representing a promising way to reduce development costs of software systems and knowledge-based systems [StBF98].

#### 2.1.1.1. OWL – The Web Ontology Language

In the Semantic Web domain, the Web Ontology Language (OWL) is traditionally employed to author ontologies. OWL is a W3C standard well-supported in most semantic engines. OWL is syntactically layered on RDF, but it adds new facilities to describe concepts and relations.
Concepts in OWL are called *classes*, and instances of concepts are called *individuals*. According to the specification [OWL04], six types of class descriptions are distinguishable:

1. a class identifier (a URI reference);

2. an exhaustive enumeration of individuals that together form the instances of a class;
3. a property restriction, i.e., a constraint on the properties that the individuals of such class has to satisfy;
4. the intersection of two or more class descriptions;
5. the union of two or more class descriptions;
6. the complement of a class description.

The first type allows defining a class through a class name, as usual in programming languages, whereas the other types describe an anonymous class by placing some constraints on the resources belonging to the class itself.
Relations or predicates in OWL are called *properties,* and the RDF triple (subject, predicate, object) is called *property statement.* A property can link individuals to individuals *(object properties)* or individuals to data values *(datatype properties)*. In OWL, it is possible to describe characteristics of properties, such as:

- The subject of the property statement, i.e., the class description to which the property is referred. This class is called *domain.*
- The object of the property statement, i.e., the class description or a data range expressed by the property. This object is called *range.*
- Two properties that have same domain and range are called *equivalent,* and indicated with the construct *owl:equivalentproperty.*
- A property that describe an inverse relation with respect another property is called *inverse,* and indicated with the construct *owl:inverseOf.* For instance, a user owns a device and a device is owned by a user.
- A property can be *transitive*, i.e., if an element $x$ is related to an element $y$, and $y$ is in turn related to an element $z$, then $x$ is also related to $z$. The OWL construct to express transitive properties is *owl:TransitiveProperty*.

- A property can be *symmetric*, i.e. if an element $x$ is related to an element $y$, then $y$ is related to $x$ with the same property. The OWL construct to express symmetric properties is *owl:SymmetricProperty.*

The standard [OWL04] proposes three sublanguages of OWL, with increasing expressive power:

- *OWL Lite* supports only a classification hierarchy with very simple constraints. The language constructs provide the basics for subclass hierarchy construction: subclasses and property restrictions, such as cardinality. Its minimal expressiveness should lead to very efficient complete reasoners to develop.

- *OWL DL* provides the maximum expressiveness possible while retaining computational completeness and decidability. It owes its name to Description Logics (DLs) on which it is based. DLs are a formal knowledge representation language that can be used to reason and infer new knowledge. In this thesis, OWL DL has been employed as language to author all the proposed ontologies.

- *OWL Full* is the most expressive sublanguage of OWL, but it loses some guarantees concerning decidability. It provides full compatibility with RDF, allowing free and unconstrained use of all constructs of RDF.

Recently, at the end of 2009, W3C has proposed a recommendation [OWL09] for a new version of OWL, called OWL 2. Its overall structure is very similar to OWL, and in particular backwards compatibility with OWL is totally guaranteed.

#### 2.1.1.2. Ontologies as Knowledge Model for Context

As in all scientific fields, at the beginning of context-awareness research, context models were specific of the individual applications and designed to be exploited and shared only between the system's components. Afterwards, the necessity of more formal models arose to facilitate context sharing and interoperability of heterogeneous systems.
Strang and Linnhoff-Popien [SiLi04] presented an accurate survey of the most relevant approaches to modeling context by means of the schemes of data structures exploited:

- *Key-Value* models. This approach uses key-value pairs to represent the context information, in the same way of environment variables in operative systems. The greatest drawback within this approach is the impossibility of representing context attributes.

- *Markup scheme* models. This approach uses markup tags and content to represent a hierarchical data structure. Typical markup-scheme models are profiles based upon an XML-based syntax.

- *Graphical* models. This approach represents context via graphics like for instance UML diagrams or Entity-Relationship schemes and its strength point is the capability to describe contextual knowledge in an immediate way.

- *Object oriented* models. This approach exploits the main benefits of the object oriented paradigm (e.g. encapsulation, reusability, inheritance) to model context. It provides API to manipulate and access information, and hides the internal representation to clients. An advantage of this approach is the capability to fit distributed system requirements.

- *Logic based* models. This approach describes context by means of facts, expressions and rules with a high degree of formality. In this way, it is able to derive new facts or to manage the existing ones via using inference.

- *Ontology based* models. Ontologies are a formal representation of a set of concepts within a domain and the relationship between these concepts. Ontology based models have high capability of expressiveness and sharing between heterogeneous systems.

Strang and Linnhoff-Popien concluded their survey assessing that ontology-based models are the approach that best fit all the requirements imposed by context-aware systems. Indeed, it is extensively acknowledged that ontologies are a promising way to specify context information. For instance, Wang *et al.* [WZGP04] pointed out that there are three advantages exploiting ontologies in context-aware systems: (i) they enable a common set of concepts about context between system's components (knowledge sharing); (ii) they enable reasoning mechanisms to extract new information from context data (logic inference); and (iii) composition of a personal ontology is supported from a large set of reusable Web ontologies (knowledge reuse). Moreover, Bettini *et al.* [BBH+10] envisioned ontologies as a powerful technology to model context information due to their capability of: (i) supplying rich expressiveness of the language; (ii) providing a formal semantics to context data, enabling context sharing; (iii) and being supported by reasoning tools that can be used both to check for consistency and to infer more abstract contexts such as situations.

### 2.1.2. Semantic Rules

It has been recognized that OWL has expressive limitations, particularly with respect to what can be said about properties [HoPa04]. For instance, there is no composition constructor. Thus, an extension of OWL to overcome these limitations is needed. Rule languages can do it. Horrocks and Patel-Schneider [HoPa04] proposed the OWL Rules Language (ORL), adding rules as a new kind of axiom in OWL. However, using such rules can lead to undecidability. Motik *et al.* [MoSS04] proposed a decidable extension of OWL with rules, where each variable in the rule is required to occur in a non-DL-atom in the rule body. However, starting from 2004, a proposal [SWRL04] for an extension of OWL with rules has been submitted to the W3C. The proposed language is the Semantic Web Rule Language (SWRL) and it combines the OWL Language with the Rule Markup Language [RMI01] a markup language to express rules in XML.
To deal with uncertainty in the Semantic Web, extensions of OWL have been proposed by several researchers, as reported in Stoilos *et al.* [SSSK06]. Indeed, a mechanism to represent vague and imprecise knowledge and information is highly desirable. In particular, rule languages that take uncertainty into account have been introduced. Pan *et al.* [PSS+06] proposed f-SWRL that extends SWRL enabling fuzzy rules such as 'being healthy is more important than being rich to determine if one is happy'. More specifically, condition atoms in a rule can include a weight that represents the importance of the atom in the rule itself. Wang *et al.* [WMYM08] enhanced f-SWRL enabling the representation of the importance of membership degrees. Recently, a Rule Interchange Format (RIF) [RIF05a] has become a W3C candidate recommendation to enable an interchange format among existing rule systems. The working group has designed a family of languages, called dialects, in order to cover the broad categories of rule systems: first-order logic, logic-programming, and action rules [RIF05b]. However, no RIF's dialects provide a support to manage fuzzy rules [WMYZ10]. Some non-standard extensions have been proposed, such as RIF Fuzzy Rule Dialect (FRD) based on fuzzy sets [WMYZ10] or RIF Uncertainty

Rule Dialect (URD) to represent directly uncertain knowledge [ZhBo08].
Thus, in this thesis we have decided to still continuing to use classical semantic web formalisms, fully based on well-established standards: the Web Ontology Language (OWL) to describe the semantic database and the Semantic Web Rule Language (SWRL) to express the semantic rule base.

#### 2.1.2.1. SWRL - Semantic Web Rule Language

SWRL combines the OWL DL and OWL Lite sublanguages of OWL with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language [SWRL04]. In other words, SWRL extends the set of OWL axioms and constructs to represent in the knowledge base also Horn-like rules.
Usually, rules are used to infer new knowledge from the facts declared in a data base. Such rules have the form of an implication between an antecedent *(body)* and consequent *(head)*:

**IF** *A* **THEN** *B*

where *A* and *B* are concepts expressed in the ontology. The meaning of such a rule is "if an individual is found to be an instance of *A*, then this implies that it is also an instance of *B*." These rules are often called *trigger rules* [BaNu03].
In SWRL both the body and head consist of zero or more atoms:

- A rule with an empty body is considered always true, thus the head is always satisfied.

- A rule with an empty head is considered always false, thus the body cannot be satisfied by any interpretation of the data base.

- In case of multiple atoms, they are considered as connected by the logical conjunction.

Atoms in these rules can be of the form C($x$), P($x$,$y$), built-in($x$,$y$,$z$), where C is an OWL class, P is an OWL property, built-in is a predefined function and $x$, $y$, $z$ are either variables, OWL individuals or OWL data values.
As an example, let us consider an ontology with the following facts:

*Person(Mario)*
*Work-Place(office)*
*is-located-in(Mario, office)*

This trivial ontology declares that *"Mario"* belongs to the class *Person*, *"office"* to the class *Work-Place*, and finally that *"Mario"* is linked to the individual *"office"* by means of the property *is-located-in*.
An inference rule to deduce that a Person who is located in a Work-Place is working, is the following:

*Person(?x) ^ Work-Place(?y) ^ is-located-in(?x, ?y)* →
*is-in-situation(?x, "working")*

The rule is expressed in the informal "human readable" syntax proposed in the SWRL specification [SWRL04], a syntax more concise and more easy to read by human beings. Variables are indicated by prefixing them with a question mark (e.g., *?x*). If this rules is applied to the previously defined ontology, the result is a new fact, i.e.,

*is-in-situation(Mario, "working")*

## 2.2. The Fuzzy Theory

There are some concepts that involve an intrinsic degree of uncertainty and vagueness. For instance, let us consider the set of *"people who are near to the conference room"* and the set of *"people who are registered to the conference"*. The latter set can be identified by simply accessing the conference register which maintains the conference members or by asking to the conference secretary. Instead, the other set is quite imprecisely defined and difficult to identify. Intuitively, it is possible to consider the distance at which a person is in relation to the conference room, and then place a threshold beyond which that person is not close anymore. Of course, it is very difficult to choose such a threshold, either basing on common sense or on subjective measurements. Moreover, at a distance slightly greater than this threshold, the person is not close to the conference room, whereas at a distance slightly less the person is perfectly close. In order to overcome such problems and handle uncertainty and vagueness encountered in the real physical world, Zadeh [Zade65] introduced the Fuzzy Set Theory in his seminal paper.

Fuzzy Logic [Zade75] aims to describe approximate reasoning based on fuzzy set theory. It can be considered as an extension of the classical Aristotelian logic.

Fuzzy systems are systems based on fuzzy set theory and/or fuzzy logic. More specifically, the term fuzzy system is employed to identify expert systems which have a linguistic rule base and an inference engine based on fuzzy logic.

In the following, each concept is described in more detail, with reference to the authoritative works of Zadeh [Zade96], Pedrycz and Gomide [PeGo98], Driankov *et al.* [DrHR96], Babuska [Babu98], Klir [Klir06].

### 2.2.1. Fuzzy Set Theory

The peculiarity of a fuzzy set is that objects belong to the set with a certain degree, called membership value. This value is a continuous number ranging from 0 (complete exclusion) to 1 (complete membership). A fuzzy set $A$ can be uniquely defined by a *membership function* of the form:

$$\mu_A : U \rightarrow [0,1]$$

where $U$ is called *domain, space*, or *universe of discourse*. For each $x \in U$, the value $\mu_A(x)$ expresses the membership degree of $x$ in the fuzzy set $A$.

It can be observed that such a definition of fuzzy set extends the traditional definition of set, in which the membership degree of the objects can assume only two values (belong or not belong). More formally, a *crisp set* $C$ (or *ordinary set*) is a fuzzy set where

$$\forall x \in U, \mu_C(x) = 0 \vee \mu_C(x) = 1$$

In principle, any kind of function of the form $\mu_A : U \rightarrow [0,1]$ can be used to represent a membership function. However, some basic functions are commonly used in the literature such as piece-wise linear functions (formed using straight lines), Gaussian distribution function, sigmoid curve, quadratic or cubic polynomial curves. In this thesis, all membership functions have a trapezoidal form, because experimental data showed that this choice enhances the ability of determining the correct user situation.

#### 2.2.1.1. Properties of Fuzzy Sets

The *support* S(*A*) of a fuzzy set *A* is a crisp set composed by all elements that belongs to *A* with a membership value greater than zero. More formally:

$$\mathrm{S}(A) = \{x \in U \mid \mu(x) > 0\}$$

The *core* C(*A*) of a fuzzy set *A* is a crisp set composed by all elements that belongs to *A* with a membership value equals to 1. More formally:

$$\mathrm{C}(A) = \{x \in U \mid \mu(x) = 1\}$$

The *height* hgt(*A*) of a fuzzy set *A* is the least upper bound of the membership values and it is defined as

$$\mathrm{hgt}(A) = \sup_{x \in U} \mu_A(x)$$

If the height of a fuzzy set is equal to 1, the fuzzy set is called *normal*, otherwise *subnormal*.
Given a fuzzy set *A* and a number $\alpha \in [0,1]$, the $\alpha$-cut of *A*, denoted by $A_\alpha$, is the crisp set defined as follows:

$$A_\alpha = \{x \in U \mid \mu_A(x) \geq \alpha\}$$

All introduced properties of fuzzy sets are illustrated in the Figure 5, in which a trapezoidal membership function is reported.

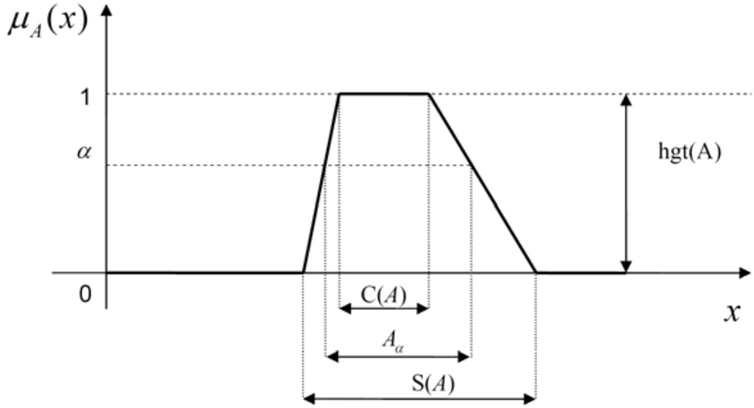



**Figure 5 – The main properties of fuzzy sets.**

### 2.2.1.2. Operations on Fuzzy Sets

Classical set theory defines univocally three basic operations on sets: *intersection*, *union* and *complement*. In fuzzy set theory, it is possible to define a class of functions that satisfy certain requirements for each operation.

*Intersection of fuzzy sets.* Given two fuzzy sets *A* and *B*, their intersection $C = A \cap B$ is defined by the membership function

$$\mu_C(x) = f(\mu_A(x), \mu_B(x))$$

where $f(\mu_A(x), \mu_B(x))$ is a t-norm, i.e., a binary function that satisfies the following properties: commutativity, monotonicity, associativity. Moreover, the identity element for a t-norm must be the number 1, i.e., $f(a,1) = a$.

*Union of fuzzy sets.* Given two fuzzy sets $A$ and $B$, their union $C = A \cup B$ is defined by the membership function

$$\mu_C(x) = f(\mu_A(x), \mu_B(x))$$

where $f(\mu_A(x), \mu_B(x))$ is a s-norm, i.e., a binary function that satisfies the following properties: commutativity, monotonicity, associativity. Moreover, the identity element for a s-norm must be the number 0, i.e., $f(a,0) = a$.

*Complement of a fuzzy set.* Given a fuzzy set $A$, its complement $\bar{A}$ is defined by the membership function

$$\mu_{\bar{A}}(x) = f(\mu_A(x))$$

where $f(\mu_A(x))$ is a function that satisfies the following requirements:

1. $f(0) = 1$ and $f(1) = 0$;
2. $a < b$ implies $f(a) > f(b)$;
3. $f(f(a)) = a$.

Typical operators used to compute intersection, union, and complement of fuzzy sets are $\mu_C(x) = \min(\mu_A(x), \mu_B(x))$, $\mu_C(x) = \max(\mu_A(x), \mu_B(x))$, and $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$, respectively.

### 2.2.2. Fuzzy Logic

Fuzzy logic aims to formalize the "approximate reasoning" that human beings use in everyday life, providing a formal method that exploits the fuzzy set theory. Fuzzy logic can be viewed as an extension of classical logic, where bi-valued logic (true or false) is extended to a multi-valued logic (degree of truth between true and false). Hence, in fuzzy logic truth values can assume any value on the interval [0, 1].
In the following, the main concepts of fuzzy logic are described.

A *Linguistic variable* is defined by Zadeh [Zade74] as a quintuple $(X, T(X), U, G, M)$ where:

- $X$ is the name of the *base variable*, which is a variable in the classical sense (e.g., distance);
- $T(X)$ is the set of *linguistic terms* of X (e.g., T(X) = {low, high});
- $U$ is the universe of discourse of the base variable;
- $G$ is a syntactic rule for generating the linguistic terms based on the primary term (e.g., very low, not high);
- $M$ is a semantic rule for associating a meaning to each linguistic term, i.e., a corresponding fuzzy set to represent the term itself.

A *fuzzy proposition* is a logical proposition assigned to fuzzy sets. If a proposition *P* is assigned to the fuzzy set *A* (written $P : x \in A$), the degree of truth of *P* is given by

$$T(P) = \mu_A(x),$$

where *x* is a variable defined in the universe *U*.
An example of fuzzy proposition is the proposition "*P*: *distance* is *low*", where *distance* is a variable and *low* is a fuzzy set. The truth degree of *P* is the membership value of the variable *distance* to the set *low.*
Fuzzy propositions can be connected and manipulated with inference rules in order to derive new knowledge. A process of approximate reasoning combines fuzzy propositions by means of fuzzy connectives and inference rules, as it is described in what follows.

*Negation.* Given a proposition $P : x \in A$, the degree of truth of not *P* is $T(\neg P) = \mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

*Conjunction.* Given two propositions $P : x \in A$ and $Q : y \in B$ the degree of truth of P and Q is $T(P \wedge Q) = \min(\mu_A(x), \mu_B(y))$. In general, $T(P \wedge Q) = f(\mu_A(x), \mu_B(y))$, where $f$ is a t-norm.

*Disjunction.* Given two propositions $P : x \in A$ and $Q : y \in B$ the degree of truth of P and Q is $T(P \vee Q) = \max(\mu_A(x), \mu_B(y))$. In general, $T(P \vee Q) = f(\mu_A(x), \mu_B(y))$, where $f$ is a s-norm.

*Implication.* Given two propositions $P : x \in A$ and $Q : y \in B$ the degree of truth of $P \to Q$ is $T(P \to Q) = \min(\mu_A(x), \mu_B(x))$. In general, $T(P \to Q) = f(\mu_A(x), \mu_B(x))$, where $f$ is any fuzzy relation which satisfies $f(0,0) = f(0,1) = f(1,1) = 1$ and $f(1,0) = 0$ (boundary conditions of classical implication). It is worth to note that the choice to implement the fuzzy implication with the *min* function does not comply with all the boundary conditions, in particular $\min(0,1) = 0$. Nevertheless, the case in which $\mu_A(x) = 0$ and $\mu_B(y) = 1$ is quite exceptional in real world applications. Hence, in this thesis, the *min* function is considered as the standard form of fuzzy implication, as common in the literature.

*Generalized modus ponens.* The generic inference model in classical logic is the modus ponens, which takes the following form:

$$\begin{array}{l} P \\ \underline{P \to Q} \\ Q \end{array}$$

If there is a rule (in the form of logical implication) $P \to Q$, and $P$ is true, then also $Q$ is true. In fuzzy logic, modus ponens has been generalized to allow truth degrees, and the generalized modus ponens takes the following form:

$$\begin{array}{l} P' \\ \underline{P \to Q} \\ Q' \end{array}$$

Where $P : x \in A$, $Q : y \in B$, $P' : x \in A'$, and $Q' : y \in B'$. The degree of truth of $Q'$ is

$$T(Q') = \sup_x f(\mu_{A'}(x), \min(\mu_A(x), \mu_B(y))),$$

where $f$ is a t-norm and the *min* operator implements the fuzzy implication. Commonly, the *min* operator is used to implement also $f$.

### 2.2.3. Fuzzy Systems

Fuzzy systems, called also *fuzzy inference systems* or *fuzzy rule-based systems*, are a particular class of expert systems which exploit a linguistic rule base and the fuzzy inference process to determine their outputs.
The overall structure of a generic fuzzy system is depicted in Figure 6, as suggested by many authors in the literature (e.g., Babuska [Babu98] or Cordòn [Cord01]).

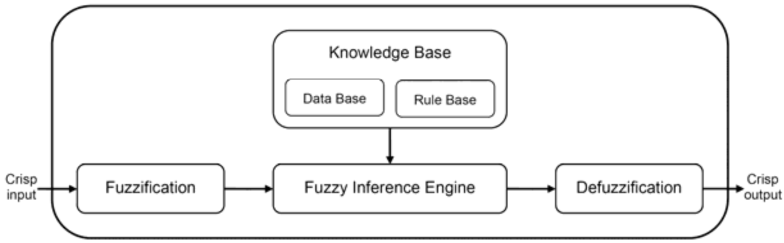



**Figure 6 – Generic architecture for a fuzzy system.**

The main components of a fuzzy system are:

1. A *knowledge base,* comprising a *Data Base* (which contains the membership functions of the linguistic terms) and a *Rule base* (which contains a collection of fuzzy rules, which can be activated simultaneously).
2. A *fuzzy inference engine*, which derives the fuzzy outputs of the system by combining the input fuzzy sets following the relations defined in the rule base.

3. *Fuzzification* and *defuzzification* modules, which enable the system to handle crisp inputs and to generate crisp outputs, respectively.

Depending on the particular structure of the fuzzy rules, it is possible to distinguish two types of fuzzy systems:

1. *Mamdani fuzzy systems,* also called linguistic fuzzy systems, in which both antecedent and consequent are fuzzy propositions. Rules are of the form:

$$r: \text{if } X_1 \text{ is } A_{i1} \text{ and } X_2 \text{ is } A_{i2} \text{ and ... and } X_n \text{ is } A_{in} \text{ then } Y \text{ is } B_i$$

2. *Takagi-Sugeno-Kang (TSK) fuzzy systems,* in which the antecedent is a fuzzy proposition and the consequent is a crisp function of the variables in the antecedent. Rules are of the form:

$$r: \text{if } X_1 \text{ is } A_{i1} \text{ and } X_2 \text{ is } A_{i2} \text{ and ... and } X_n \text{ is } A_{in} \text{ then } Y_i = f_i(X_i),$$

where $f(\ )$ is a (usually) linear function which combines the system inputs.

Mamdani systems have the advantage of transparency, i.e., they are easily interpretable by humans, whereas TSK systems have the advantage of low computational costs and high accuracy.

## 2.3. Genetic Algorithms

Genetic algorithms belong to a class of methods that aim to solve optimization problems by imitating the principles of natural evolution.

They were proposed by Holland [Holl75] in his seminal book. Genetic algorithms operate on a *population* which encodes randomly generated solutions for the problem. Each solution is called *chromosome* and it is usually represented by a string. A *gene* is a piece of a chromosome (usually, a bit or a short sequence of bits) that encodes a particular element of the solution. For instance, if the optimization problem concerns finding a set of parameters to tune a system, each parameter may be represented by a gene. The population evolves toward better solutions of the problem by applying operators such as *crossover* or *mutation* among the chromosomes. The goodness of each solution is evaluated by a *fitness* function. Genetic Algorithms have been widely employed to automatically tune some parameters in fuzzy systems, in the so called Genetic Fuzzy Systems. Indeed, the automatic definition of a fuzzy system can be seen as an optimization problem, characterized by a large search space, and asking for suitable algorithms [Herr08].

In what follows, genetic algorithms are described in their main characteristics and in their application to fuzzy systems, with reference to the authoritative works of Michalewicz [Mich94], Mitchell [Mitc98], and Cordòn [Cord01].

### 2.3.1. Characteristics of Genetic Algorithms

A Genetic Algorithm starts at time $t = 0$ with a randomly generated population of $n$ chromosomes, each representing a candidate solution of the problem. Chromosomes can be initially determined also by exploiting the knowledge of a domain expert. Then the algorithm proceeds in steps called generations. More specifically, at each step a new population is derived based on the antecedent, i.e., $P(t+1) = f(P(t))$. The new population is composed by the best chromosomes of the previous population, selected by means of an evaluation function called fitness function. On the selected

chromosomes are applied two genetic operators, in order to mate them *(crossover operator)* and slightly modify them *(mutation operator)*. The new obtained chromosomes compose the new population. Optionally, some of the best chromosomes of the previous population can be moved to the new population without changes. The population continues to evolve until a stopping criterion is met, e.g. when the maximum number of generations is reached or when the fitness function returns a value that overcomes a threshold.
Figure 7 describes the steps of the genetic algorithm presented above.

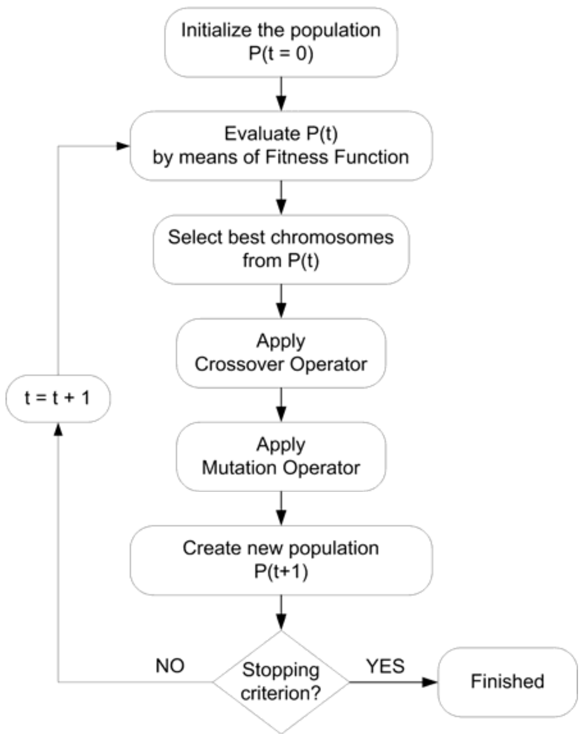



**Figure 7 – A simple genetic algorithm.**

The implementation of the genetic operators tightly depends on the coding schema of the chromosome and on the specific problem. Crossover operators choose some individual in the population and

mate them, i.e., mix the genetic material of the parents chromosomes to form a new chromosome. Mutation operators aim to introduce new genetic material in the population, by randomly altering the value of some genes in the chromosomes.
Genetic Algorithms have been proved to be a robust method to search optimal solutions in complex spaces [Cord01].

### 2.3.2. Genetic Fuzzy Systems

In order to design a Fuzzy System, an important task to be considered is the definition of the knowledge base. The knowledge base is composed by (i) a *data base,* containing the linguistic terms and their membership functions; and (ii) a *rule base,* containing a set of linguistic rules that consider the linguistic terms defined in the data base. The knowledge base has to be derived from the knowledge of a domain expert, because the fuzzy system is not able to learn. In order to overcome this drawback, genetic algorithms can be employed. A *genetic fuzzy system* is a system that exploits genetic algorithms to automatically generate or optimize the knowledge base of a fuzzy system. The architecture of such a system is depicted in Figure 8. The *genetic design* of a genetic fuzzy system involves the codification of the parameters of the knowledge base into a suitable genetic representation. Once the knowledge base is optimized, the *fuzzy processing* can start, by computing outputs as a standard fuzzy system.

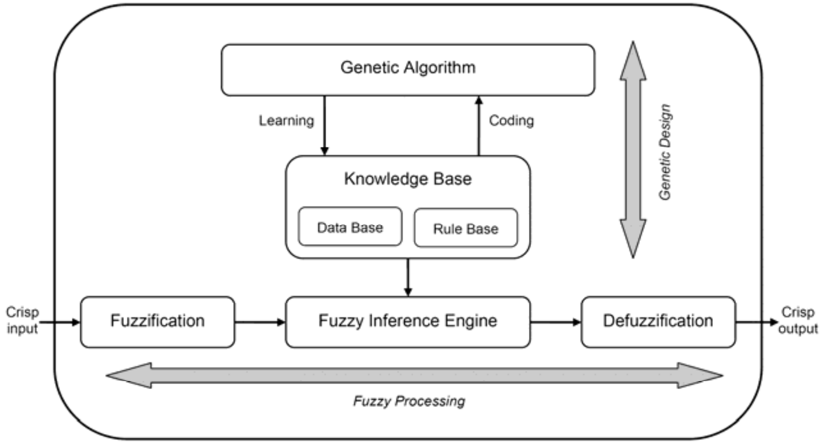



**Figure 8 – Architecture of a Genetic Fuzzy System.**

According to the different parts of the knowledge base that are automatically determined by genetic algorithms, a taxonomy of Genetic Fuzzy Systems can be proposed [Herr08]:

- *Genetic tuning:* if the knowledge base already exists, and genetic algorithms are employed to improve the performance of the system, without changing the rule base.

- *Genetic learning:* the knowledge base is learnt in its components by exploiting genetic algorithms. Hence, the process can include the design of an adaptive inference engine.

Genetic tuning comprises the case in which the process automatically adjusts the shapes of the membership functions in the data base, but leaves unchanged the number of fuzzy terms in each fuzzy partition. This is the approach that is proposed in Chapter 6. Indeed, genetic fuzzy systems allows a deeper control of the optimization process than other methods of automatic tuning of the knowledge base (such as neuro-fuzzy systems), as stated by Cordòn *et al.* [CGH+04].

# Chapter 3

# Related Work in Context- and Situation-Awareness

In this Chapter, we focus our attention on previous approaches to handle user context and situation. The first section proposes a survey of the main literature in sensing and exploring user situations, whereas the other section is narrowed in the field of recommendation systems. In particular, approaches for dealing with uncertainty in contextual data and for exploiting the context history are reviewed.

## 3.1. Approaches to Determine the User Situation

Since the Weiser's vision of ubiquitous computing [Weis95], researchers have proposed their own personal solutions to push computers into the background and enable a new way to interact with them. Context has been recognized as a fundamental key to develop new services that can adapt to the circumstances in which they are used [CCDG05]. Indeed, as reported in the most accepted definition proposed by Dey [Dey01], *context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.* Recently, a new level of abstraction is emerged, i.e., situation. Situation can be viewed as logically aggregated pieces of context [AnHa08]. Such pieces have to be invariant through the whole time in which the situation occurs [Weiß06]. Situation awareness is important because allow targeting precisely the demand of the user at a certain time [Weiß06]. However, mobile systems have to deal with uncertain, rapidly changing, partially true data from multiple and heterogeneous sources [KMK+03]. Thus, reliable methods to acquire and process uncertain context data to infer situations are particularly desirable. Korpipää *et al.* [KMK+03] proposed a framework to managing uncertainty in raw data and infer higher-level context abstractions with a related probability. The framework uses a blackboard-based approach to enable communications among entities in the system. All context sources publish their data in the blackboard, which acts as a centralized module to process contextual data and deliver high-level information, i.e. the user situation, to the application. Fuzzy sets are employed to convert unstructured raw data into a representation defined in a context ontology by means of predefined fuzzy labels. A confidence value is associated to contextual data to describe the context uncertainty. Situations are recognized by means of naïve Bayes classifier, which learns conditional probabilities for each situation from training data. Mäntyjärvi and Seppänen [MäSe03] proposed to represent context information by applying fuzzy membership functions. In particular, raw data from sensors are converted in context information by means of fuzzy quantization. Such information is then employed as input for fuzzy rule-based controllers to adapt applications according to the context. For instance, if the user is moving and the loudness of environment is silent, then volume of application is turned down to minimum level. However, no semantic description of context is considered. Ranganathan *et al.* [RaAC04] model uncertainty in situation awareness by attaching a confidence value to all pieces of contextual information. Confidence values measure the probability or the membership value of the event corresponding to the contextual information being true. Indeed, authors proposed three methods to infer the user situation, (i) using probabilistic logic, (ii) using fuzzy logic, or (iii) using Bayesian networks. In the probabilistic and fuzzy approaches, developers have to write their own rules to infer situation, whereas in the Bayesian approach developers have to define the network specifying the relations among contextual information. Gu *et al.* [GuPZ04] proposed a context-aware middleware to support context reasoning in order to derive the user situation. Uncertainty is faced in two manners. First, they propose to extend the context ontology to allow additional probabilistic markups. Second, they adopt Bayesian network to support the inference process of the user situation. In [CXC+05], user situation is assessed as a combination of context information which is expressed by a fuzzy linguistic variable. More specifically, a situation is represented by a set of 3-element tuples, where each tuple contains a certain contextual information (e.g., the current network rate), a linguistic value that characterizes that situation (e.g., high), and finally the fuzzy membership degree of the contextual information to the linguistic value. Thus, the recognized situations contain a list of fuzzy degrees referred to several linguistic values and it is difficult to compare situations with each other and to rank them. Haghighi *et al.* [HKZG08] proposed an approach for situation modeling and reasoning under uncertainty based on fuzzy theory. Situations are expressed by multiple contextual conditions joined in a fuzzy rule, where the consequent represents the degree of confidence in the occurrence of a situation. Moreover, developers can specify weights to represent the relative importance of each contextual condition for inferring a situation.

The main problem with these approaches is that the relationship between contextual information and situations is static, and cannot adapt to the changing behavior of the user. Indeed, Byun and Cheverst [ByCh03] pointed out that when context awareness is reached by means of predefined rules, users have to reconfigure the system when their behavior changes, resulting in a frustrating and annoying task. In order to automatically recognize the user situation related to the user behavior, the authors proposed to exploit context history. Adaption is provided by fuzzy decision trees, which takes uncertainty in the raw data into account. Hagras *et al.* [HDCL07] proposed a novel learning technique to adapt the system to the continuous changing in the user behavior. The technique is an unsupervised data-driven one-pass approach for extracting type-2 fuzzy membership functions and rules from the context history of the user. However, the authors do not separate the situation determination phase from the system response phase based on the inferred situation. Indeed, the sensed contextual information is immediately employed to adapt the system to the user needs, which are application specific. For instance, a particular configuration of some sensors such as internal light level, bed pressure, internal temperature can lead to activate the window blinds. Thus, the concept of situation is lacking in the system. Finally, Anagnostopoulos and Hadjiefthymiades [AnHa10] introduced advanced semantics in the context representation, combining the fuzzy logic approach with the semantic one. In particular, advanced representation schemes concern specialization, mereonomy, mutual exclusion and compatibility. By means of a neuro-fuzzy classification engine, the system learns to map sets of contextual information to particular situations and builds the corresponding fuzzy rules. However, the proposed system deals only with physical contextual information, such as orientation of the mobile device, illumination level, humidity, and not with other virtual contextual information, such as user calendars or geographical maps. Moreover, explicit means of representing user situations are also needed. The mapping between contextual information and situations should be customized by the user.

## 3.2. Use of Context in Recommendations

One of the first approaches that recognized explicitly the importance of context in recommendations was Herlocker and Konstan [HeKo01]. In particular, the authors proposed task-specific recommendations, where the task is identified as a set of example items related to the task itself.

For instance, if the user provides a hammer as example item in a shopping recommender, the system can recommend buying nails. Such associations are identified automatically by the system, using data about user interest-ratings, i.e., associating items that have similar ratings. Naganuma and kurakake [NaKu05] proposed a task-oriented service navigation system that supports users in finding appropriate services by browsing rich task ontology. This ontology contains a variety of structured tasks in the real world and their links to appropriate services that may be able to solve a user's task. In [LFWK08], the authors extended this system by taking the user situation into account, in order to suggest tasks and services actively, without the need for initial user input. However, this approach does not consider the inescapable uncertainty that affects contextual data in order to infer the correct user situation. Indeed, situations are recognized by applying dynamic assertional classification of contextual entities such as the location, the time and the neighbor people.

[Weiß06] proposed a system that exploits situation awareness to provide user with the desired information and services. In this approach, a situation describes a user demand that occurs at a certain time and it is formed by a sequence of contexts defined as logical expression, such as *LocationOfTheUser(stadium)* or *TypeOfMovement(Fast).* Both situation inference and service selection are based on ontologies to infer first a set of situations and then a set of services which may be relevant in these situations. However, also in this case, no uncertainty aspects are considered. Moreover, at a given time, a user may be in zero, one or many situations but no final ranking is given to help user in choosing the best fitting situation or to list the recommended services in an apt order. Recently, Petry *et al.* [PTVS08], proposed ICARE, a recommendation system that returns references to experts in a requested domains using contextual information. More specifically, the system improves its recommendations by using the user's and expert's context, privileging those experts who better fit user's current needs. Examples of contextual information employed are expert availability, approachability, social distance, etc. Contextual rules are defined to set appropriate weights in order to decide which contextual information should be favored given a user context. Hence, the recommendations are different for each user, according to his context. However, ICARE does not consider any uncertainty aspects in the contextual information. Moreover, the system does not act proactively but waits for the user requests in order to provide the desired recommendations.

### 3.2.1. Dealing with the Uncertainty in Recommendations

Fuzzy logic has been proved as a promising approach to manage the natural uncertainty that affects contextual data.

Cena *et al.* [CCG+06] employed fuzzy logic in a context-aware tourism recommender. The system exploits personalization rules to suggest services (e.g. restaurants, places to visit, etc.) tailored to the user profile and context. User profile is a very important piece of the system, and it is built by (i) explicit data of user (such as age, gender, general interests, etc.); (ii) inferred data by means of fuzzy rules based on domain knowledge (such as propensity to spend, specific interests, etc.); and (iii) user current needs and wishes, by observing the sequence of user interactions with the system (such as printed pages, on-line booking, etc.). Based on the user interests maintained in the profile and the user position, the system computes an overall score for each service and recommends services in an order depending on the score. Thus, the context is limited mainly to the user location that acts as a filter to recommend near services. Moreover, proactivity of the recommendations is not provided, but only envisioned as future work.

Park *et al.* [PaYC06] proposed a context-aware music recommendation system that employs fuzzy Bayesian networks and utility theory. In particular, a fuzzy system is exploited to preprocess contextual data from various sensors and the Internet, in order to have quantized

inputs for the Bayesian network. Based on these inputs, the network can infer the user context and assigns a probability. Finally, recommendations are proposed depending on a final score, which is computed taking the inferred context and user preferences into account. In this approach, no semantic aspects of the contextual information are considered. Moreover, the inference process is entirely based on the Bayesian network, resulting in a not easily understandable and customizable mechanism for average users.
Min *et al.* [MiKC08] introduced a smart phonebook that recommends a contact list according to the user situations. The authors employ Bayesian Networks to infer three kinds of high-level contexts in which the user can be involved: a social context, i.e., the degree of friendship with other peers, and two personal contexts, i.e., the user emotional state and the quantity of commitments that the user has to carry on. Once these contexts are recognized, specific rules are fired in order to recommend the contact list that best fits the user situation. However, rules associate piece of contexts directly with the items to be recommended. Indeed, as an example, the system recommends calling a friend if it is his/her birthday. Thus, the abstraction level introduced by using the concept of situation is lacking in the system.

### 3.2.2. Enhanced Recommendations with Context History

Context history has been identified as an important piece of information to determine the user situation. Context history is strictly related to the activity that the users are going to perform, and to the resources which they might be interested in [ByCh03]. However, the use of context history in recommendation systems is considered a relatively under-explored area [HSKK09]. Mayrhofer [Mayr05] has proposed to use context history to predict the current situation. Here, sensor data are classified into higher-level context identifiers, and then the next possible contexts are predicted by using an algorithm based on

Markov models. Byun and Cheverst [ByCh04] have exploited context history to induce rules for adapting the system to the user behavior. In particular, fuzzy decision trees have been employed to handle the vagueness in sensed data and to represent the level of uncertainty in the suggestions to the user. Si *et al.* [SKMA05] have developed a platform that can learn user behaviors from context history, in order to provide the most relevant services in the current situation. Here, Bayesian Networks are used to correlate contexts and services, by modeling the relationship between the sensor data and the selected services in the context history. Yap *et al.* [YaTP05] have proposed to dynamically choose the set of contextual information on which the resource recommendations can be based. Support Vector Machines techniques are applied to the context history in order to learn a relevance coefficient of each contextual information.

Shin *et al.* [SLYL09] proposed to exploit fuzzy logic and context history for more accurate recommendations. Fuzzy set theory is employed to handle raw contextual data and abstract them with a set of concepts, such as "warm" for a temperature of 25° C. When the user selects an item, the current contextual data are recorded and associated with the item. During the time, these associations compose the user context history as an aggregated context model. In this way, the system can compute the importance of each contextual data with respect of the selected items and establish a relation between them. Thus, a similarity measurement between the current context and context history is computed, and the appropriate items are recommended. In this approach, fuzzy logic is employed only to represent context information in a more abstract way, by applying fuzzy membership functions. Moreover, the system needs a context history large enough to allow computing the relation between contextual data and selected items. This can lead to a well-know problem for new users, called *cold start* problem, in which the system cannot recommend any item because has no reference data for the user. Finally, the concept of situation is lacking in the system.

Hong *et al.* [HSKK09] have suggested the use of context history to automatically extract the user preferences about services. In particular, by means of decision trees and association rules, the system is able to associate user context with services and even predict the next services the user might need.

## Chapter 4

# A Mobile Resource Recommendation with Fuzzy Logic and Semantic Web

In this Chapter, we propose a situation-aware resource recommender (SARR) for mobile users, which allows locating resources while the users need them, by taking the current situation into account [LFWK08; GVCF07]. In the proposed approach, recommendations are delivered in a proactive way, considering possibly uncertain contextual information. This is achieved by the integration of a fuzzy logic engine and a web semantic engine. More specifically, the semantic engine infers one or more current situations by exploiting domain knowledge expressed by ontologies and semantic rules. If multiple possible situations are inferred, the fuzzy engine computes a certainty degree for each situation, taking the intrinsic vagueness of some conditions of the semantic rules into account. Thus, the system can associate a rank with the recognized situations depending on such certainty degrees. Each situation is therefore associated with specific tasks, on the basis of domain knowledge expressed in terms of a task ontology [OWL04; SWRL04]. Finally, the specific current task together with contextual information is used to recommend a set of resources, identified by means of a Label (or Tag)-based file system [TTH+09].

## 4.1. The Tagging Paradigm

To organize personal resources independently of their location, a bookmark management system is usually employed. The function of bookmarks is to offer an associative memory for personal usage. Conventional bookmarks contain a URL (or local path) and a title of the resource [AbBC98]. Bookmarks reduce the cognitive and physical loads of managing URL addresses, and facilitate the return to groups of related resources. Users collect bookmarks to create their own personal information space and share it with others [KaMa01]. However, organizing bookmarks is labor-intensive, requires a lot of time, and is difficult to do. In fact, typically users do not organize bookmarks [AbBC98]. Further, web usability studies show that bookmark lists are far from representing an effective personal information space [KaMa01].

A number of researches have been performed to enhance bookmark functionality. In [NOST02], context-dependent bookmarks have been discussed, with a method based on the automatic extraction of representative keywords of resources. The automatic extraction of representative keywords is applicable only to documents. In the case of applications, descriptors should convey the user intention, which is difficult to extract without semantically representing high-level concepts.

Recently, the tagging paradigm for information organization has become popular, especially in the context of collaborative tagging systems for managing shared bookmarks or public digital images. Resources are tagged by annotating them with simple descriptors. Conjunctions of tags can be used to narrow down the search space and, at the limit, to identify a limited set of resources such as a folder path. Information organization based on tags is capable of overcoming many problems inherent in hierarchical file systems [VGD+08; BGSV06]. Information about tags can also be represented in an ontology, with the

advantage that extensions of the data model and integration with other semantic-aware applications are easy to realize.
The number of tags is likely to grow with the increase of the collection of resources. Hence, information represented by tags cannot be efficiently exploited without a proper user interface (using a laptop or desktop device), or without a further level of semantics, which helps the system take the current intention of the users into account. Key to support mobile users with an efficient access to resources is an intelligent platform that mediates between services and users by observing the user activity.

## 4.2. Overall Architecture

The Situation-aware Resource Recommender (SARR) runs 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 Figure 9. In the server side, there are two main modules, i.e., the *fuzzy engine* and the *semantic engine*. The fuzzy engine module is in charge of assessing conditions that are inherently vague, such as mobility and proximity state of users. The domain model and the behavior of the system are instead handled in the semantic engine module, which infers the current situation of the users and suggests the most useful resources for that situation. The *observer* module is in charge of controlling the state of the fuzzy and the semantic engines, allowing the interoperability between these modules.
The control flow of the server-side application is steered by the *application controller* module, which acquires the data collected by the *contextual data sources* block. Whenever a new value is acquired, it is transmitted to the observer, which triggers the fuzzy engine module. This module verifies whether the value belongs with certainty degree higher than 0 to a fuzzy set in the partition corresponding to the

linguistic variable which the value refers to. If the certainty degree is higher than 0, the observer inserts the corresponding property into the ontology and triggers the semantic engine. If the semantic rules infer more than one situation, the observer asks the fuzzy engine to assess the final certainty degree for the recognized situations. The certainty degree of a situation is important for considering the order with which resources are recommended. If more than one situation is recognized, all the related resources are recommended, with an order depending on the certainty degrees.

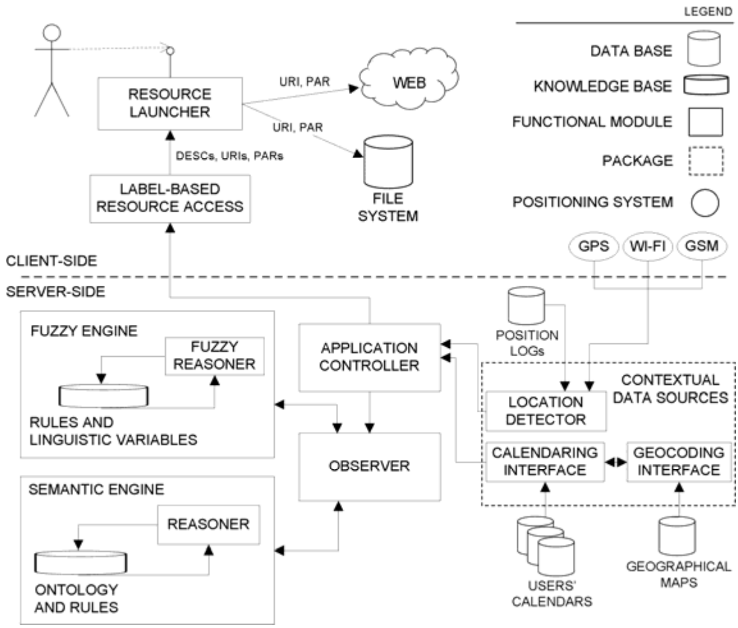



**Figure 9 – Overall architecture of SARR.**

The *contextual data sources* package comprises a set of interfacing modules for different data sources, such as geographical maps, users' calendars and positions.

In particular, numerical data concerning user's movements, i.e., position, time and speed, 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 [SCGL05]. Regardless of the available technologies, the location detector provides a generalized interface in terms of user movement data and its accuracy. To this aim, the GPX (GPx eXchange format) standard abstraction is used [GPX02]. GPX is a lightweight XML data format, which allows describing waypoints, tracks and routes. More specifically, as regards the location detector, in GPX a collection of time-spatial points is considered as a track. A piece of a simplified GPX track is shown in Figure 10.

```
<trkpt lat="43.92765" lon="10.915965">
  <time>2009-10-19T7:47:11Z</time>
  <speed>4.671609504842717</speed>
</trkpt>
```

**Figure 10 – An example of GPX document representing the user movement.**

The *geocoding interface* module is a basic service that provides associated geographic coordinates (expressed as latitude and longitude) from other intelligible textual location data, such as street addresses, or zip codes (postal codes). Intelligible location data comes from the user's calendar, where meetings or other events are recorded by the user. The data format used in this module is imported from the Google Maps API [Maps05], a web mapping service application. This allows a great interoperability with the client-side simulator, i.e., an auxiliary web application that has been used in the experiments.

The *calendaring interface* module offers time-management services. This module allows users to insert, via mobile application, the events or appointments for each day, which are used as a reference by the system. In particular, the application controller uses the user daily timetable to schedule the specific events to monitor. The calendaring interface module is based on the Google Calendar API [Cale06], a web application that can be synchronized with the most common mobile devices.

On the client side, the *label-based resource access* [BGSV06] module is supplied by the *application controller* module with a set of labels and contextual parameters. This information is used to locate and adapt recommended resources. More specifically, the label-based resource-access module provides an abstraction of the file system with tag semantics [BGSV06]. In a traditional file system, a resource is only located within its exact (most specific) path, but not implicitly contained in higher-level directories. For instance, considering pictures, which can be organized by author, genre or date, it allows only one access path, such as "year/author/album" but not "author/album/year". On the contrary, the label-based file system allows for large flexibility, since it allows treating information objects, such as bookmarks, addresses, e-mails and applications, uniformly with respect to metadata [BGSV06]. Hence, the specification of a resource becomes a set of labels rather than a URI.

Finally, the selected resource is identified in terms of description, URI and parameters, and can be started by the *resource launcher* module, which is directly connected to the local or web resources.

In the following, we consider the design of the server-side application, focusing on the semantic and fuzzy engines.

## 4.3. The Semantic Engine Module

To recommend resources inherent in the current user task, the system takes the current user situation into account. According to [Weiß06], the term "situation" is a business level concept that allows targeting precisely and at different levels of granularity the demand of the user at a certain time. In our system, each situation is devoted to identify a collection of user tasks. In a task-navigation paradigm [LFWK08], the user is supported to find appropriate resources by relying on a task ontology, which represents common sense knowledge about her/his usual activities. In order to suggest in a proactive manner tasks and resources actively, i.e., without the need for initial input from the user, the context is a fundamental vehicle. Context refers to any relevant information that can be used to characterize a user [Dey01]. Therefore, a situation can be modeled as a collection of context information that is invariant as long as the situation occurs [Weiß06]. For instance, the situation "meeting" can be inferred from a set of contextual information such as "user is stationary", "user is located in the scheduled place at the scheduled time", "user is close to the meeting organizer", and so on.

Another important advantage of using contextual information is the possibility of deriving contextual parameters to adapt the identified resource to the current demand of the user. Hence, the full goal of the ontology is to identify a set of resource descriptors together with a set of contextual parameters. Furthermore, to make the ontology independent of the specific applications and related path installations, and of the number, type and sequence of parameters, two abstraction mechanisms have been introduced in the system, by means of the following respective modules: the *label-based resource access,* which allows the exact localization of an application or a document, described more generically as a resource in the ontology, and the *resource launcher,* which enables the forwarding of the gathered parameters, and the launching of the selected application.

The semantic engine module exploits two ontologies: the first ontology *(situation ontology)* allows connecting contextual information to situations, and the second one *(task ontology)* allows connecting situations to tasks, and then to specific resources. The ontologies have been developed by using the Web Ontology Language (OWL, [OWL04]), a W3C standard well supported in most semantic engines.

To develop the ontologies we adopted the following iterative and incremental process [Lópe99]. First, we interviewed some domain experts to model some user scenarios and to understand basic domain concepts and relationships among these concepts. Each interview allows producing the narration of a story. After the interview, the narration is formalized, producing a register of sentences, as in the excerpt of Figure 11 referred to the situation ontology. This register is then processed, with a textual analysis approach. Textual analysis is a process of analysis of a domain which helps to identify the fundamental ontology elements: classes, relations, properties and values. In Figure 11, nouns, verbs and attributes are highlighted with a different underlining to identify classes, relations, properties and values. To identify an upper ontology, which is valid for many application scenarios, in the first interviews a bottom-up development process has been employed, starting with the definition of the most specific classes, with subsequent generalization of these classes into more general concepts. Figure 12 shows basic concepts (e.g., *User, Calendar,* etc.) and basic relationships (e.g. *owns, contains,* etc.) identified for the situation ontology. Here, concepts and relationships are represented by oval shapes and directed edges, respectively. In particular, general concepts such as *Time* and *Place* are inherited from publicly available ontologies [Time06; DCKF03] according to the best practices of reusing domain ontologies. In the figure, external ontologies are enclosed in dashed rectangular shapes.

$s_1$ : User owns a Calendar;

$s_2$ : Calendar contains Events;

$s_3$ : Event can be of two types, whose value can be business or private;

$s_4$ : Event is attended by a User;

$s_5$ : Event is organized by (hence, is attended by) User, is located in a Place;

$s_6$ : Event is scheduled at a Time;

$s_7$ : User has a name;

$s_8$ : User owns a Device;

$s_9$ : User is located in a Place, has a mobility, is close to a User;

LEGEND : class, relation, property, value;

**Figure 11 – Excerpt of the register of sentences.**

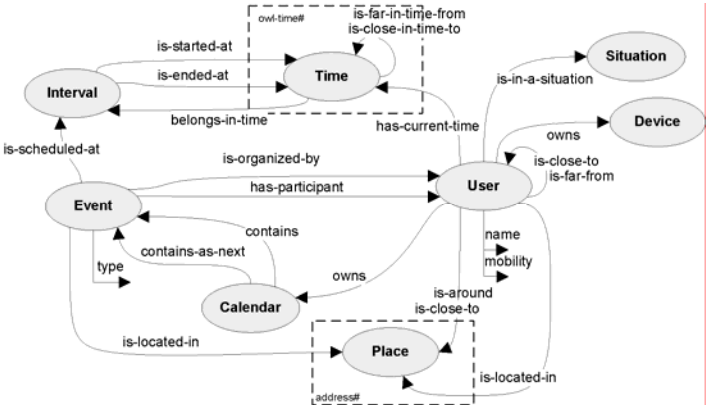



**Figure 12 – The situation ontology.**

Similarly, we developed also the task ontology. Figure 13 shows the upper task ontology. Here, the dashed edge named "required" represents a property that is not implemented in the ontology, but is conceived only for a better understanding.

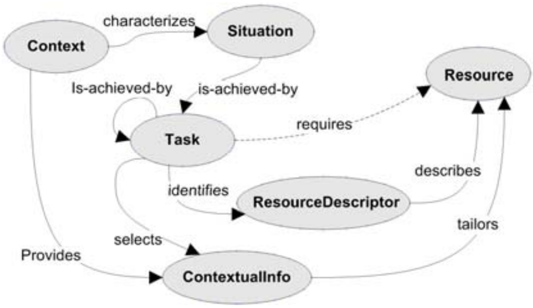



**Figure 13 – The task ontology.**

In addition to the ontologies, a set of rules is employed to infer the situation. Rules are expressed in the Semantic Web Rule Language (SWRL, [SWRL04]), 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 it is named OWL DL [OWL04].

Figure 14 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 (based on two-valued logic) and fuzzy, represented in Figure 14 in bold and italic bold, respectively. The condition *"is a participant"* is derived from the user's calendar, and is 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. Although web ontology is the most promising assets for context modeling for ubiquitous computing [StLi04], the classical semantic web formalisms do not allow the representation of uncertainty [Harm06].

As regards fuzzy logic, there has been a significant theoretical work in extending Description Logics with fuzzy set theory [LuSt06; BoSt08]. Considering also the semantic web perspective, an OWL ontology to represent fuzzy extensions of the OWL language has recently been proposed [BoSt09]. With this approach, some reasoning can be performed by using standard OWL reasoners. The ontology can be extended to other fuzzy statements. Ongoing works concern the development of a plug-in for a well-known visual editor [Prot11], and the implementation of some optimization techniques to reduce the running time.

In our approach, to deal with uncertainty still continuing to use classical semantic web formalisms, we have coupled the semantic engine with a fuzzy engine. Thus, the semantic engine does not handle directly the uncertainty. This approach allows achieving several advantages (see [BoDG09] for more details): (i) there is no need to agree with a non-standard fuzzy ontology to use; (ii) a number of resources is available for standard ontologies, such as ontology editors [Prot11] and public ontologies [Swoo04] to reuse; (iii) existing, well-known, widely-used crisp reasoners [Pell10], and APIs [Jena10] can be used.

```
  owns(?user1, ?aCalendar)
^ contains(?aCalendar, ?anEvent)
^ is-located-in(?anEvent, ?aPlace)
^ is-started-at(?anEvent, ?anEventTime)
^ has-current-time(?user1, ?user1Time)
^ has-participant(?anEvent, ?user2)
^ is-close-in-time-to(?user1Time, ?anEventTime)
^ is-close-to(?user1, ?aPlace)
^ is-far-from(?user1, ?user2)
^ PreMeeting(?aSituation)
 ⇒ is-in-a-situation(?user1, ?aSituation)
```

(a)

```
IF   user2 IS A PARTICIPANT to the scheduled event
AND  user1Time IS CLOSE IN TIME TO the scheduled time
AND  user1 IS CLOSE TO the scheduled place
AND  user1 IS FAR FROM user2
THEN user1 IS IN A SITUATION of pre-meeting
```

(b)

**Figure 14 – An example of SWRL rule:**
**(a) human readable syntax and (b) natural language.**

The observer module is responsible for integrating the semantic engine and the fuzzy engine. More specifically, when the semantic rules are characterized by fuzzy conditions, the observer asks the fuzzy engine for their evaluation. The fuzzy engine returns a certainty value in [0, 1] for each uncertain condition. If the certainty value is larger than zero, the condition is considered to be true in the semantic inference. Otherwise the condition is considered to be false. When the semantic engine infers a situation, the fuzzy engine, based on these fuzzy conditions, computes a certainty degree for this situation. Thus, the semantic engine using a two-valued logic determines which situation occurs, whereas the fuzzy engine establishes, once a situation has occurred, its certainty degree.

## 4.4. The Fuzzy Engine Module

In the system, the fuzzy model is described using the Fuzzy Control Language (FCL) specification [Cing10]. FCL is a standard for Fuzzy Control Programming published by the International Electrotechnical Commission (IEC). Figure 15 shows an example of linguistic variable defined using FCL. It is worth noting how each term is defined in terms of vertices of a trapezoidal fuzzy set.

```
// Define linguistic variable
FUNCTION_BLOCK distance

  // Define base variable
  VAR_INPUT
     distance : REAL; // meters
  END_VAR

  // Define linguistic values
  FUZZIFY distance
     TERM veryLow := (0,1) (0,1) (200,1) (400,0);
     TERM low := (0,1) (0,1) (400,1) (1400,0);
     TERM high := (20,0) (700,1) (1500,1) (1500,1);
  END_FUZZIFY

END_FUNCTION_BLOCK
```

**Figure 15 – An example of linguistic variable expressed in FCL.**

Each fuzzy condition is expressed by using linguistic variables declared in FCL. In the system, we have defined a set of linguistic variables to express a series of common contextual conditions. For instance, the condition *"user1* is close to *the scheduled place"* depends on the linguistic variable *distance.* More specifically, the linguistic variable distance can assume the linguistic values *veryLow, low,* and *high,* as defined in Figure 15. The state of each fuzzy variable is monitored by the observer module, which, for each variation of the values of the linguistic variables, updates the corresponding properties in the semantic model. The values of the variables are collected from contextual data sources. To design the linguistic variables, a representative set of contextual data is used. Figure 16 shows an example of GPS track, provided by a smart phone. A user moves from $Q$ to $P$ to participate to a meeting event. In the fuzzy engine, spatial and temporal proximities are expressed as linguistic variables, let us say $\Delta s$ and $\Delta t$, respectively. The number and meaning of the possible linguistic values for these variables are application-dependent. In our case study, we partitioned the universe of definition of these variables with trapezoidal membership functions, appropriately extracted from experimental data. The use of trapezoidal membership functions helps constrain the number of activated conditions, thus limiting the number of concurrently inferred situations. We adopted the linguistic values [*veryLow, low, high*] and [*low, medium, high*] for $\Delta s$ and $\Delta t$, respectively.

**Figure 16 – An example of GPS track, provided by a user smart phone.**

In particular, let $(\mathbf{s},t)$ be the reference location and time for the event scheduled in the user's calendar. Let $(\mathbf{s}_1,t_1)$ and $(\mathbf{s}_2,t_2)$ be the current location and time of user1 and user2, respectively, provided by their mobile devices. Let $\Delta t_1 = |t_1 - t|$, $\Delta s_1 = \|\mathbf{s}_1 - \mathbf{s}\|$ and $\Delta s_{12} = \|\mathbf{s}_1 - \mathbf{s}_2\|$ be the current user temporal proximity and distances. Hence, the fuzzy rule corresponding to the semantic rules in Figure 14 is

```
IF   is LOW AND
     is LOW AND
     is HIGH
THEN situation1 is Pre-Meeting
```

In the fuzzy engine, we implemented the logical AND and the implication operators as minimum. To allow an efficient integration with the semantic engine, fuzzy rules are processed in the fuzzy engine in a two-stage way. In the first stage, the observer module periodically synchronizes the properties of the semantic model, considering the certainty degrees of the fuzzy conditions in the antecedent part of the fuzzy rules. For each condition with a certainty degree larger than zero, the observer inserts the corresponding property in the ontology and triggers the semantic engine. Hence, at this stage each fuzzy condition is monitored separately in the fuzzy engine. In the semantic engine, the corresponding crisp property is processed in the overall semantic model. Hence, the semantic engine can infer one or more situations. Once the semantic rules have inferred the current situations, in the second stage, the observer asks the fuzzy engine to assess the final certainty degree for the recognized situations. The certainty degree of a situation is important for considering the order with which services are recommended. If more than one situation is recognized, all the related services are recommended, with an order depending on the certainty degrees.

## Chapter 5

# Combining Fuzzy Logic and Semantic Web

In this Chapter, 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 [OWL04] and is enriched with a truth degree depending on a level of certainty. Situations are inferred by means of semantic rules [SWRL04], which take the fuzziness of the contextual antecedents into account, and are ranked depending on their fuzzy values.

Unlike in the previous Chapter, 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 [FNFK05], where the user is supported to find appropriate services and documents by relying on the task ontology.

## 5.1. 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 Figure 17.

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 [SCGL05]. 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, [SWRL04]).

Thus, designers can express how the system should 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.

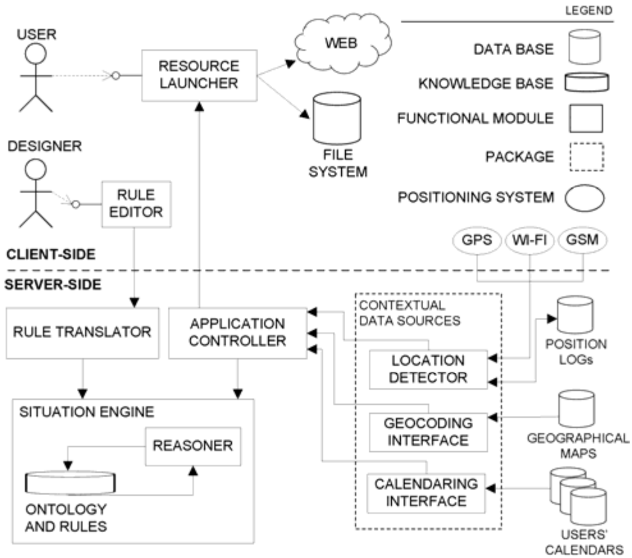



**Figure 17 – Overall architecture of the situation-aware resource recommender.**

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.

## 5.2. 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 [OWL04]), 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*. Figure 18-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 Figure 18-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.

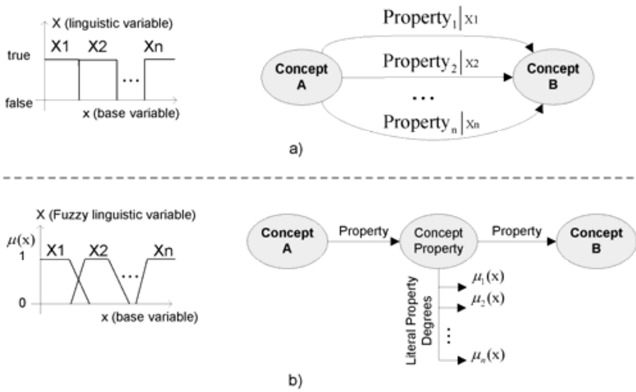



**Figure 18 – An OWL-compliant fuzzy extension of a property.**

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 Figure 19, 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 [Time06; DCKF03], 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).

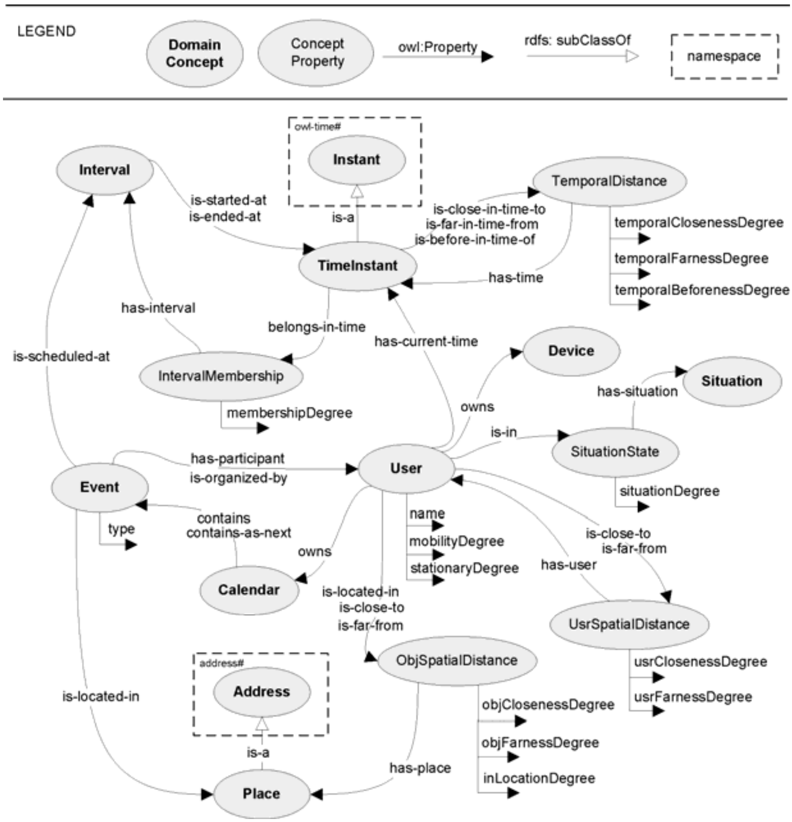



**Figure 19 – The upper situation ontology.**

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 [OWL04].

Figure 20 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 Figure 20 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.

```
owns(?user1, ?aCalendar)
^ contains-as-next(?aCalendar, ?nextEvent)
^ is-scheduled-at(?nextEvent, ?anInterval)
^ is-started-at(?anInterval, ?eventStartTime)
^ mobilityDegree(?user1, ?user1mobilityDegree)
^ has-current-time(?user1, ?user1Time)
^ is-before-in-time-of(?user1Time, ?temporalDistanceReification)
^ has-time(?temporalDistanceReification, ?eventStartTime)
^ temporalBeforenessDegree(?temporalDistanceReification,
                  ?beforenessDegree)
^ type(?nextEvent, "business")
^ Pre-Meeting-on-Movement(?inferredSituation)
^ is-in(?user1, ?situationStateReification)
^ has-situation(?situationStateReification, ?inferredSituation)
^ swrlb:add(?sum, ?user1mobilityDegree, ? beforenessDegree)
^ swrlb:subtract(?difference, ?user1mobilityDegree, ?beforenessDegree)
^ swrlb:abs(?absDifference, ?difference)
^ swrlb:subtract(?doubleMinimum, ?sum, ?absDifference)
^ swrlb:divide(?computedDegree, ?doubleMinimum, 2)
  ⇒  situationDegree(?situationStateReification, ?computedDegree)
```

(a)

```
IF user1 IS A PARTICIPANT to the scheduled event
AND user1 IS moving
AND user1Time IS BEFORE IN TIME the scheduled event start-time
AND event HAS TYPE business
THEN user1 IS IN A SITUATION OF pre-meeting-on-movement
```

(b)

**Figure 20 – 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 Figure 21, the upper task ontology is represented.

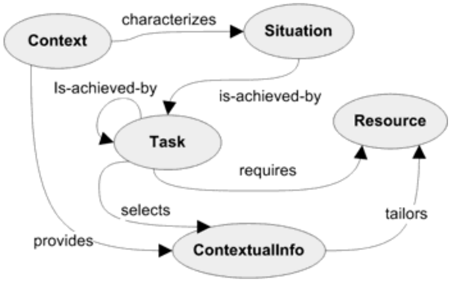



**Figure 21 – The upper task ontology.**

## 5.3. 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 Figure 20-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 user 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 [MäSe03]. In order to evaluate 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, [Cing10]), 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 Figure 22.

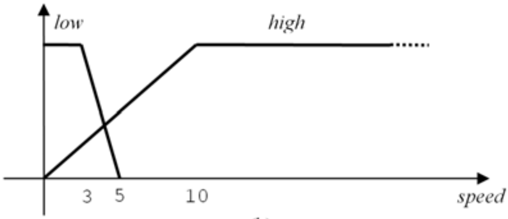
```
FUNCTION_BLOCK speed

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
```

(a)




(b)

**Figure 22 – Definition of linguistic labels in FCL.**

## 5.4. A Simple Integration of Fuzzy Logic into SWRL

In our implementation, we expressed fuzzy rules, such as the one described in Figure 20-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 [LuSt07])

$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}.R_1(\mathbf{z_1})[s_1],...,R_l(\mathbf{z}_l)[s_l], s = f(s_1,...,s_l)$$

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;
4. $\mathbf{z}_i, \mathbf{z}'_j$ are tuples of constants or variables in $\mathbf{x}$ or $\mathbf{y}$;
5. $s, s_1,...s_l$ 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 $R(x)[s]$ the *head,* $\exists \mathbf{y}.R_1(\mathbf{z_1})[s_1],...,R_l(\mathbf{z}_l)[s_l]$ the *body* and $s = f(s_1,...,s_l)$ the *scoring atom*. We also allow the scores $[s],[s_1],...[s_l]$ 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. $s = f(s_1,...,s_l)$.

As an example, in the following we show the high-level encoding of the fuzzy rule in Figure 20-b:

is−in−a−situation(?user1, ?aSituation)[s] $\leftarrow$ 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),
$s = \min(s_1, s_2)$

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 (s1) and being before a meeting (s2) (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) [KlMP00].

A rule base $\Re$ 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 $\Re$ 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 $\Re$ 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 meet on the score combination functions (see *e.g.*, [Stra05], 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

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

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

$$R_1(\mathbf{z_1})[s_1],\ldots,R_l(\mathbf{z}_l)[s_l], s = f(s_1,\ldots,s_l).$$

We also impose that a rule base $\Re$ 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 head 2[2], i.e.

$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}_1.\phi_1(\mathbf{x},\mathbf{y}_1)[\mathbf{s}_1]$$
$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}_2.\phi_2(\mathbf{x},\mathbf{y}_2)[\mathbf{s}_2]$$
$$\ldots$$
$$\ldots$$
$$\ldots$$
$$R(\mathbf{x})[s] \leftarrow \exists \mathbf{y}_n.\phi_n(\mathbf{x},\mathbf{y}_n)[\mathbf{s}_n]$$

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

$$R_1(\mathbf{x})[s] \leftarrow \exists \mathbf{y}_1.\phi_1(\mathbf{x},\mathbf{y}_1)[\mathbf{s}_1]$$
$$R_2(\mathbf{x})[s] \leftarrow \exists \mathbf{y}_2.\phi_2(\mathbf{x},\mathbf{y}_2)[\mathbf{s}_2]$$
$$\ldots$$
$$\ldots$$
$$\ldots$$
$$R_n(\mathbf{x})[s] \leftarrow \exists \mathbf{y}_n.\phi_n(\mathbf{x},\mathbf{y}_n)[\mathbf{s}_n]$$
$$R(\mathbf{x})[s] \leftarrow R_1(\mathbf{x})[s_1],\ldots,R_n(\mathbf{x})[s_n], s = g(s_1,\ldots,s_n)$$

where $R_1,\ldots,R_n$ 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, $g(s_1,\ldots,s_n) = \max(s_1,\ldots,s_n)$, but in general, any so-called *s-norm* [KlMP00] (used to combine disjunctive information) may be appropriate as well. This transformation

[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.

guarantees then that $\mathfrak{R}$ 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,

   is−before−of(?userTime, ?aTime)[s2]

   becomes

   is−before−of(?userTime, ?aTime, s2)

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

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

   which defines a predicate $P_f(s_1,...,s_l,s)$ such that $s = f(s_1,...,s_l)$, using the built-in arithmetic operations of the rule language.
3. Now, we replace each rule

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

   with the crisp rule

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

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

min(s1, s2, s3) ⟵ built-in(s3 = min(s1, s2))

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),
min(s1, s2, s)

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 [DaPi02]), which allows to represent an *n*-ary relation via unary and binary relations. So, for instance, for the relation

is−before−of(?userTime, ?aTime, s2)

we create a new class

is−before−ofRelation(?aTimeReification)

with two additional properties

is−before−ofValue(?aTimeReification, ?aTime)
is−before−ofDegree(?aTimeReification, s2)

and, thus, is−before−of(?userTime, ?aTime, s2) will be replaced with

is−before−of(?userTime, ?aTimeReification),
is−before−ofValue(?aTimeReification, ?aTime),
is−before−ofDegree(?aTimeReification, s2) .

[3] `http://www.w3.org/TR/swbp-n-aryRelations/`

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

is−in−a−situation(?user1, ?aSituation, s)

as before, we create a new class

is−in−a−situationRelation(?aSituationReification)

with two additional properties

is−in−a−situationValue(?aSituationReification, ?aSituation)
is−in−a−situationDegree(?aSituationReification, s)

then add

is−in−a−situation(?user1, ?aSituationReification),
is−in−a−situationValue(?aSituationReification, ?aSituation)

to the rule body and replace the head with

is−in−a−situationDegree(?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 [Kalm84]):

min(s1, s2, s3) $\leftarrow$ sum(sm, s1, s2)
substract(diff, s1, s2),
abs(absdiff, diff),
substract(sd, sm, absdiff),
divide(s3, sd, 2),

is−in−a−situationDegree(?aSituationReification, s) $\leftarrow$
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 do not go further into the reification procedure as it is pretty common and well-known in the Semantic Web literature.

## Chapter 6

# Adapting the Situation Recognition to the User Behavior

In the previous Chapters, we proposed a general architecture for a recommender system enhanced with situation awareness. We defined the linguistic variables used in the fuzzy layer considering a generic user without taking specific user habits into account. Of course, the definition of these variables through the corresponding membership functions is a critical step of the overall recommending process [PoLH09]. Indeed, the shape and position of these functions strongly affect the computation of the degrees of certainty. Thus, to increase the recognition rate, shapes and positions should be adapted to the user habits. Currently, some systems already allow a personalization degree, but the users have to input and update their preferences manually in order to receive personalized services. A more efficient technique for personalization would be to deduce user habits automatically from the context history. Indeed, the employment of context history can be extremely effective in enabling personalization and adaptation by discovering recurrent patterns in the data [ByCh03].

In this Chapter, we briefly describe the architecture of our resource recommender. Then, we show how the definition of the linguistic variables can be tuned to the specific user via a genetic algorithm (GA) by using the context history. In the Chapter 7, we discuss using real business cases how this personalization increases the performance of our system, allowing recognizing each situation with a higher precision than the system developed for a generic user.

## 6.1. Overall Architecture

The overall system architecture of the resource recommender is shown in Figure 23. Here, we will illustrate only the main blocks of this architecture. The interested reader can find a more detailed description in [CCLM10], where we have also shown a comparison with other recently proposed recommenders. In the server side, the semantic engine and the fuzzy engine are the main modules. The semantic engine infers one or more current situations by exploiting domain knowledge modeled by ontologies (expressed in the Web Ontology Language – OWL [OWL04]) and semantic rules (expressed in the Semantic Web Rule Language – SWRL [SWRL04]). Figure 24 shows basic concepts and relationships identified for the situation ontology. In Figure 25 we provide an example of semantic rule expressed in natural language by using the ontology of Figure 24. In the rule, we have represented the conditions which typically are affected by a degree of uncertainty in italic bold. These conditions are modeled by using fuzzy propositions expressed in terms of linguistic variables and linguistic values in the fuzzy engine. These propositions are therefore connected by a logical AND implemented by using the minimum operator in order to form a fuzzy linguistic rule. Once fired, this rule can compute a certainty degree for the situation inferred by the corresponding semantic rule in the semantic engine.

The interoperability between the fuzzy engine and the semantic engine modules is guaranteed by the observer. More specifically, the observer module transmits to the fuzzy engine each contextual value which is affected by uncertainty. Then, the fuzzy engine checks whether the value belongs to a fuzzy set in the linguistic variable at some degree. If this occurs, the observer communicates to the semantic engine that the corresponding condition in the semantic rule can be considered true, thus triggering the semantic inference process. Obviously, the value can belong to more than one fuzzy set and therefore more conditions in different semantic rules are considered true, thus firing more than one semantic rule and possibly inferring more than one situation. Since the fuzzy engine computes a degree of certainty for each situation, taking the intrinsic vagueness of some conditions of the semantic rules into account, the system can associate a degree of certainty with each situation inferred by the semantic engine. Each situation is therefore associated with specific tasks on the basis of domain knowledge expressed in terms of a task ontology. Finally, the specific current task together with contextual information is used to recommend a set of resources, identified by means of a Label (or Tag)-based file system.

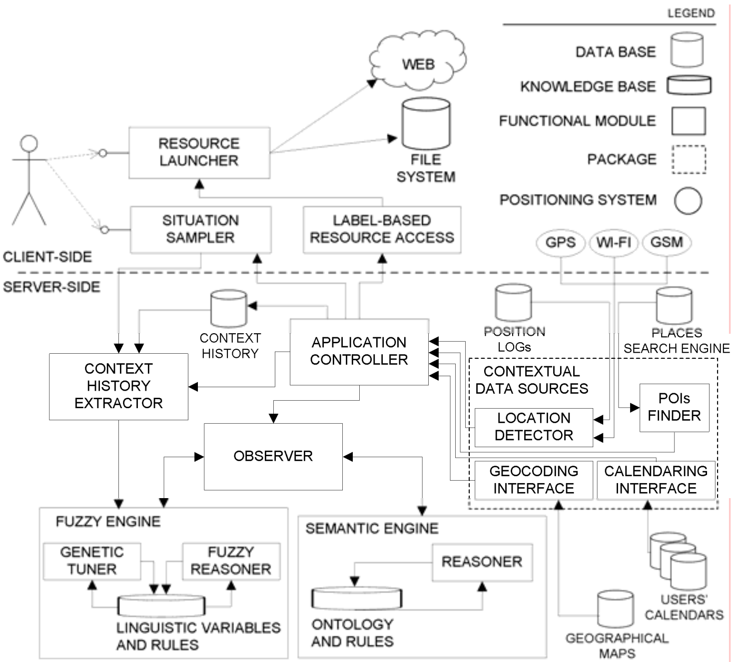


**Figure 23 – The overall system architecture.**

The resources are recommended in the same order of the situations with which they are associated: from the resources associated with the situation characterized by the highest degree of certainty to the ones associated with the situation with the lowest degree of certainty.

The application controller module handles the execution flow of the server-side application, managing the activities of the other modules and acquiring data collected by the contextual data sources package. Contextual data concern geographical maps, user calendar, user position, user speed, Point Of Interests (POIs) for the user. In particular, the application controller drives the process of recording the acquired contextual data over the time to build the context history for the user.

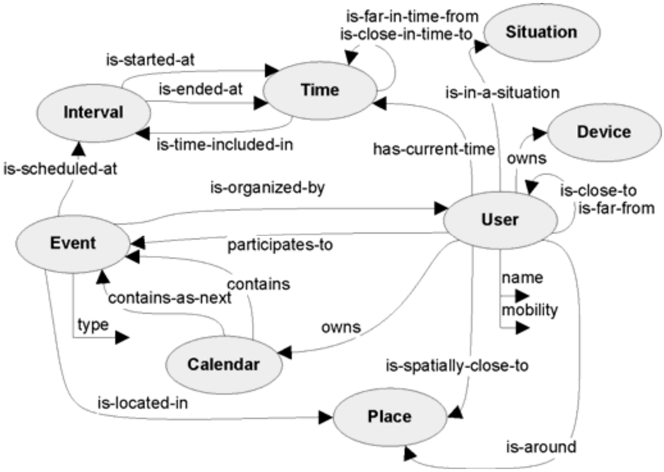



**Figure 24 – The Situation Ontology.**

The context history extractor module is aimed at producing the training set needed for the tuning of the linguistic values so as to adapt the fuzzy engine to the specific user. In particular, the module associates a set of tracks of contextual data (context history) with the corresponding situations. In the fuzzy engine, the genetic tuner implements a GA that optimizes the membership functions associated with the linguistic values, as detailed in the next section.

On the client side, the label-based resource access [BGSV06] module provides a reference to tagged resources. Indeed, the application controller identifies the recommended resources by using abstract descriptors (labels) in place of their URIs, with the aim of being independent of the resources and enabling reusability of the ontology [CCLM09; CCLM10]. The reference to the resource is employed by the resource launcher module. Finally, the situation sampler module allows tracking the instants of time and the situations for the context history extractor, during the tuning phase. The tuning phase can be started manually by the user or automatically by the client application depending on a performance index that is monitored on the client side.

```
IF user1 PARTICIPATES-TO meeting1
AND meeting1 HAS-TYPE "business"
AND user1 HAS-MOBILITY "stationary"
AND user1Time IS-TIME-INCLUDED-IN meeting1Time
AND user1 IS-SPATIALLY-CLOSE-TO meeting1Place
THEN user1 IS-IN-A-SITUATION "ongoing-meeting"
```

**Figure 25 – An example of a semantic rule in natural language.**

## 6.2. The Genetic Tuner

Each rule in the fuzzy engine is expressed by using linguistic variables. For each linguistic variable, we define a set of linguistic values and associate a fuzzy set with each of these linguistic values. The fuzzy sets describe the meaning of the linguistic values. This meaning is generally fixed by considering a generic user. Actually, different users have different behaviors. Thus, it can be a very hard task to find a meaning which satisfies all the possible users. As an example, let us consider the rule shown in Figure 25. To infer a degree of certainty for the situation "ongoing-meeting", the spatial closeness to the meeting place has to be evaluated. To this aim, a linguistic variable is defined with two linguistic values: close and not-close. To define a meaning for these two values for a generic user is not however a trivial task. Indeed, a very precise user would usually note in his calendar the complete address (street and number) of the meeting place, whereas a less precise user might note the street name only. To timely recognize the closeness of the user to the point of interest, the fuzzy set corresponding to the linguistic value close should be representative of both users. Considering the difference between the user habits, to achieve this objective is practically impossible. Indeed, a fuzzy set characterized by a narrow support would not allow detecting the closeness to the meeting place for the less precise user, whereas a fuzzy set with a wide support would detect too early the closeness for the very precise user.

To lessen this drawback and therefore improve the performance of the resource recommender, the specificity of each user has to be taken into account. This can be performed by employing the context history of the specific user for adapting the meaning of the linguistic values used in the rules of the fuzzy engine. To this aim, we can adopt a GA.

GAs have been so widely used to tune membership functions of linguistic values in fuzzy rule-based systems that a specific term, genetic fuzzy systems, has been coined in the literature [Herr08]. Although in the last years different algorithms and procedures have been proposed to learn membership functions from data [KaAl05], in this paper, we adopt a very simple approach. On the other hand, our aim is only to show that the use of a tuning mechanism for adapting the resource recommender to the user habits and behaviors can considerably increase its accuracy and responsiveness.

Let us consider the generic linguistic variable $X_j$ shown in Figure 26. We assume that each linguistic value is represented by a trapezoidal membership function, $A_{j,t}$, whose support is $[a_{j,t}, d_{j,t}]$ and whose core is $[b_{j,t}, c_{j,t}]$. Further, for each fuzzy set $A_{j,t}, t = 1 \ldots T_j - 1$, we suppose that $c_{j,t} = a_{j,t+1}$ and $d_{j,t} = b_{j,t+1}$. Finally, $a_{j,1} = b_{j,1}$ and $c_{j,T_j} = d_{j,T_j}$ coincide with the left and right extremes of the universe, respectively. Thus, the strong partitions made of these membership functions can be represented by $T_j - 1$ pairs $(a_{j,t}, b_{j,t})$. Let *M* be the number of linguistic variables which have to be tuned. The overall data base can be defined by the chromosome shown in Figure 27.

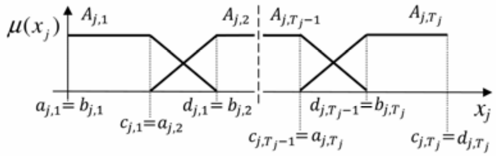



**Figure 26 – A generic partition of a linguistic variable.**

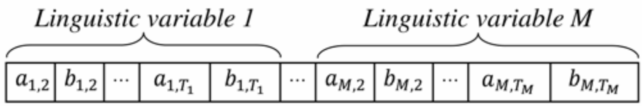



**Figure 27 – The chromosome coding.**

We aim to tune the membership functions so as to increase the capability of the system to recognize the desired situation. To this aim, we maximize the following fitness function *f*. Let $s_1,\ldots,s_S$ be the possible situations the recommender system can recognize. Let $s_t$ be the target situation. Then, *f* is defined as:

$$f = \sum_k \left( \mu_{s_t}(k) - \max_{r \neq t} \left( \mu_{s_r}(k) \right) \right) \tag{1}$$

where $\mu_{s_t}$ and $\mu_{s_r}$ are the certainty degrees with which the fuzzy engine recognizes the target situation $s_t$ and each situation $s_r$ different from $s_t$ for the sample *k* in the training set. The training set is built on the basis of samples of the context history. Each sample is made of the contextual variables that allow inferring the situation, together with the user situation itself. To give a glimpse of the context history, let us consider again the semantic rule reported in Figure 25.

Here, the context history is made of: (i) the mobility of the *user1*; (ii) the temporal inclusion of the *user1* time in the *meeting1* time; (iii) the spatial closeness of the *user1* position to the place of the *meeting1*. These contextual values are periodically recorded and associated to the respective user situations. Once the training set is large enough (a few hundreds of samples for each situation), the GA can be executed. We would like to highlight that for each observed situation, we store approximately an average of 400-450 samples (about one per minute).

The initial population of the GA is made of 50 chromosomes. Each individual of the population is randomly generated within the universe of the base variables. We adopt a BLX-$\alpha$ crossover operator with $\alpha$ = 0.5 [EsSc93], an adaptive feasible mutation operator [VaBa09] and stochastic uniform selection [Bake87]. The algorithm stops when the average fitness of the population, over 2000 generations, varies less than $10^{-6}$. At the end of the GA execution, the membership function parameters are tuned by using the values of the chromosome with the highest fitness value.

## Chapter 7

# Evaluation Case Studies

In order to show the effectiveness of the proposed approach for situation assessment, we applied the framework in the field of resource recommendations. The resource recommender has been applied to three real business cases, in order to show the effectiveness of the proposed approach. Business cases concern (i) a *pharmaceutical consultant* in typical business situations, (ii) an *off-site university student*, who performs a daily travel to go to university and return, and (iii) a *peddler,* who participates to markets and fairs in order to sell goods.

By means of a series of interviews with domain experts, a knowledge model for each business case has been developed. In particular, the upper context ontology has been extended with domain-specific ontologies, identifying the concepts and relations among concepts that better describe the business case. For instance, Figure 28 shows the comprehensive context ontology for an off-site university student. The domain-specific context ontology related to this case study contains specific concepts such as *Canteen, Classroom, Course,* etc. Sub-concepts are represented by white oval shapes and white directed edges indicate inheritance.

Moreover, different situations for each case study have been identified, and the related semantic rules have been defined. An example of a semantic rule is shown in Figure 14. For the pharmaceutical consultant, the situations of interest are: (i) *Meeting-Planning,* when the user is planning the calendar of business appointments; (ii) *Pre-Meeting On Movement,* when the user is going to have a meeting; (iii) *Ongoing-*

*Meeting,* when the user is involved in a meeting; (iv) *Post-Meeting,* when the user has just finished a meeting; (v) *Hospital-Conference,* when the user is giving a scientific talk in a hospital; (vi) *Call-for-Tenders,* when the user is attending a public auction; (vii) *Meal,* when the user is having a meal during the lunch break. For the off-site university student, the situations identified are: (i) *Pre-University-Day,* when the user is leaving his apartment and he is going to take the train; (ii) *Uni-Traveling,* when the user is heading to the university; (iii) *Studying,* when the user is waiting the beginning of the lectures; (iv) *Attending-Courses,* when the user is attending lectures; (v) *Meal,* when the user is having a meal during the lunch break in the student canteen; and (vi) *Home-Traveling,* when the user is going back to home. Finally, for the peddler, the situations of interests are: (i) *Retailing,* when the user is selling goods in markets or fairs; (ii) *To-Market,* when the user is heading to the market place in order to start selling; (iii) *To-Home,* when the user is heading to home at the end of the working day; (iv) *Procurement,* when the user is purchasing goods at a supplier shop; (v) *Warehouse-Management,* when the user is managing all the aspects related to his own warehouse.

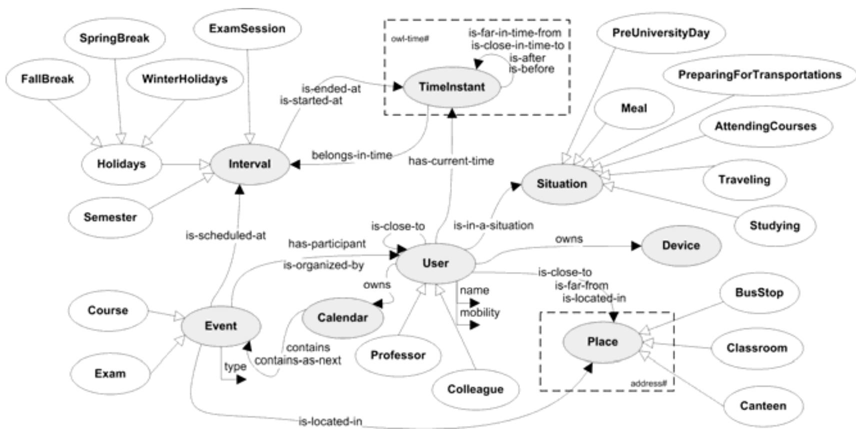



**Figure 28 – The comprehensive context ontology for an off-site university student.**

Afterwards, for each situation a set of possible tasks and related resources have been identified, starting from the actual demands of interviewed experts. Thus, the upper task ontology has been extended with domain-specific concepts and relations. Figure 29 illustrates a simplified excerpt of the comprehensive task ontology for a peddler.
In order to tune the linguistic variables of each semantic rule, the genetic approach has been employed. Firstly, starting from five real tracks for each case study, 42 training tracks have been generated. Each track contains the user movements for a whole day and the related context history, as explained in Chapter 6.

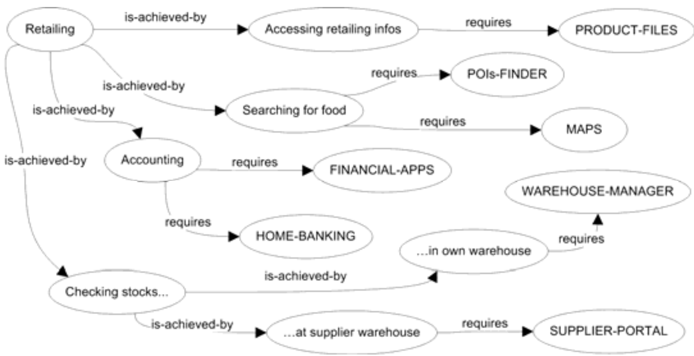



**Figure 29 – An excerpt of the task ontology defined for the situation "Retailing".**

To produce real tracks, we used an *Apple iPhone 2G* smart phone, permanently connected to the Internet and to the GPS signal, and equipped with InstaMapper[4], a free service that enables to track a phone in real time. Training tracks have been generated by means of a client-side simulator, i.e., an auxiliary web application, based on Google Maps API [Maps05]. The simulator generates new tracks based on instructions provided by the user, such as the geographical coordinates of the starting point, the number of events during the day, the distance between each event, etc. Moreover, it can simulate user

[4] InstaMapper, http://www.instamapper.com/, accessed Jan 2011.

movements under different circumstances, such as different means of transportation (walking, by bicycle or by car), different traffic conditions (without/with traffic jam), or different weather conditions (sunny day, cloudy, rainy, etc). Noise is also introduced to make contextual sources very close to real world signals. Figure 30 shows the user interface of the simulator during a batch generation of the tracks. In particular, some conditions can be noted in the configuration area, such as *weather: 'calm'* and *transportation: 'by car'*.

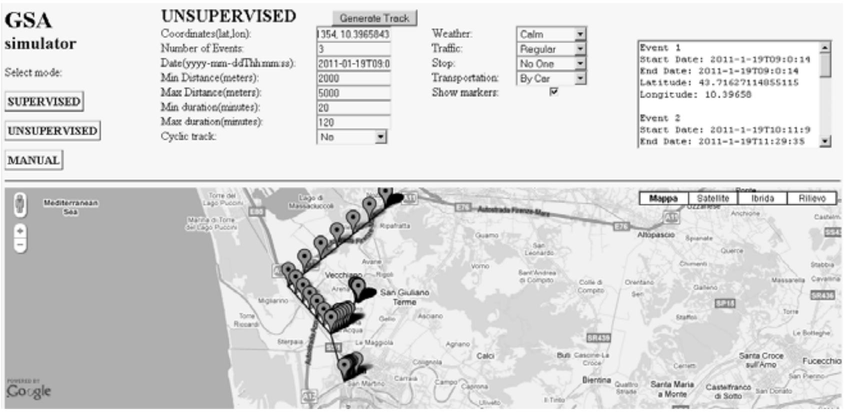



**Figure 30 – The client-side simulator to generate tracks.**

A domain expert for each domain of the case studies defined a set of linguistic variables, which finally has been tuned by the GA. For instance, let us consider the case study of the pharmaceutical consultant. The linguistic variables involved in the case study are: (i) *spatial closeness,* which represents the distance of the user from a place expressed linguistically as close and not-close; (ii) *temporal relativity,* which denotes the order between two instants of time and is expressed linguistically as before and after; (iii) *time inclusion,* which assesses whether an instant of time belongs to a temporal interval and is expressed linguistically as included and not-included; and (iv) *user mobility,* which represents the speed of the user and is expressed linguistically as stationary and not-stationary. Figure 31 shows the linguistic variables defined by the domain expert and tuned by the GA, respectively.

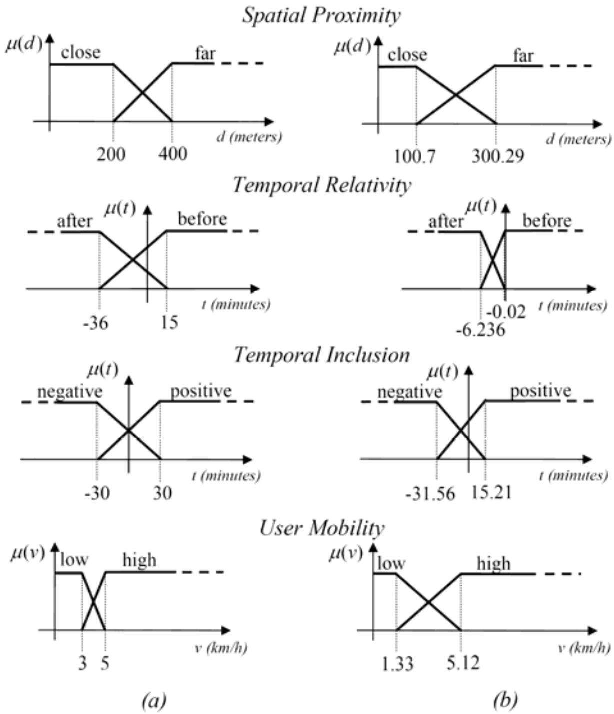



**Figure 31 – Linguistic variables for the case study of the pharmaceutical consultant defined by the domain expert (a) and tuned by the GA (b).**

After the tuning process, the system has been tested by a user for each case study. In particular, for each user, a week timetable has been considered, consisting of 77 events for the pharmaceutical consultant, 96 events for the off-site student, and 136 events for the peddler.

To asses the reliability and timeliness of the recommender, the following performance index has been considered. Let us assume that for each type $E_i$ of event, we have $N_i$ occurrences $o_{i,p}$. For each occurrence $o_{i,p}$, we record the instant of time $t_{i,p}$ at which that occurrence occurs, and the time $t'_{i,p}$ at which the recommender recognizes the occurrence. Let us define the responsiveness of the recommender to the type $E_i$ of event as:

$$Resp(E_i) = \frac{\sum_{p=1}^{N_i} \left| t'_{i,p} - t_{i,p} \right|}{N_i} \tag{2}$$

Table 2 shows the responsiveness of the recommender for each type of event occurred during the testing. Experimental results show that the recommender tuned by the GA considerably outperforms the recommender configured by the domain expert. In particular, with the proposed technique the recommender is able to increase the responsiveness on average of almost 30%, with peaks of 45% for some specific type of events.

To assess the capability of our approach to adapt the meaning of the linguistic terms to the user behavior, we have applied the GA to a different pharmaceutical consultant in similar business situations. Figure 32 shows the linguistic variables after the GA optimization. By comparing Figure 32 with Figure 31.b, we can observe that the abscissas corresponding to the crossing points between *after* and *before*, and *included* and *not-included* in, respectively, linguistic variables *temporal relativity* and *time inclusion* are considerably smaller for the second consultant than for the first. This can be explained by the different habits of the two consultants. Indeed, the second consultant is typically latecomer whereas the first consultant is generally punctual.

**Table 2. Responsiveness of the system.**

| **Case Study** | **Situation (Event)** | **Responsiveness (sec.)** | |
|---|---|---|---|
| | | ***Recommender defined by the domain expert*** | ***Recommender tuned by the GA*** |
| Pharmaceutical Consultant | Pre-Meeting (begin) | 50.52 | 36.01 |
| | Pre-Meeting (end) | 86.9 | 48.24 |
| | Ongoing-Meeting (begin) | 111.8 | 84.83 |
| | Ongoing-Meeting (end) | 32.17 | 23.28 |
| | Post-Meeting (end) | 27.53 | 23.15 |
| | Hospital-Conference (begin) | 117.45 | 68.86 |
| | Hospital-Conference (end) | 41.61 | 35.79 |
| | Meal (begin) | 102.75 | 69.25 |
| | Meal (end) | 59.3 | 49.29 |
| Off-Site Student | Uni-Traveling (end) | 26.56 | 19.33 |
| | Attending-Courses (begin) | 41.56 | 30.67 |
| | Attending-Courses (end) | 40.44 | 27.54 |
| | Meal (begin) | 110.07 | 69.17 |
| | Meal (end) | 116.69 | 79.25 |
| | Home-Traveling (start) | 54.93 | 35.25 |
| Peddler | To-Market (begin) | 3.51 | 2.62 |
| | To-Market (end) | 11.4 | 7.5 |
| | Retailing (begin) | 2.24 | 1.85 |
| | Retailing (end) | 5.62 | 3.67 |
| | Procurement (begin) | 8.04 | 5.41 |
| | Procurement (end) | 14.43 | 9.58 |
| | To-Home (begin) | 11.64 | 8.11 |
| | To-Home (end) | 36.68 | 27.52 |

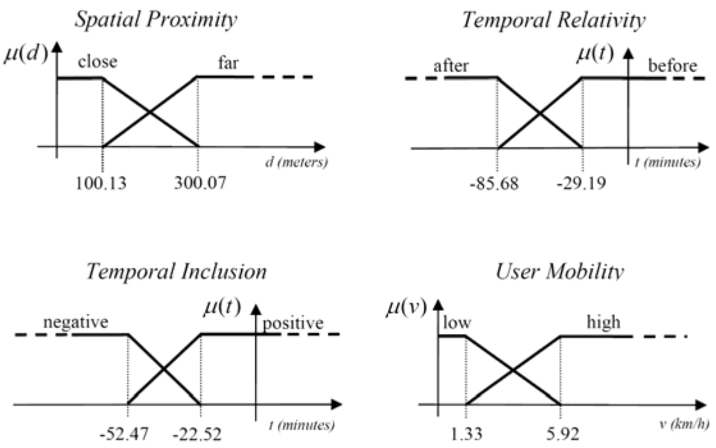



**Figure 32 – Linguistic variables tuned by the GA on the behavior of another pharmaceutical consultant.**

Moreover, we have tested also the user acceptance of the recommender. In particular, five pharmaceutical consultants and three off-site students have been asked to evaluate the recommender. To this aim, an auxiliary web application, based on Google Maps API [Maps05], has been developed. The application provides an online simulation interface that is used by the user himself, as shown in Figure 33. The user can choose his position in the map and provides information about his speed and the current date. The test is composed by two phases, which comprise a predefined number of recommendations, i.e., 10 iterations. First, after the user has input the data about his position, the application proposes a list of all resources in the smart phone, without a particular order and relation with the user situation. Thus, the user is invited to choose the resource that he needs, given the figured situation. In the second phase, the recommender is enabled and, after the user has input the data about his position, the application exploits the inferred situation to filter the recommended resources. Hence, the user is invited to choose the desired resource, guided by the predefined task ontology of the case study. Time required for the selections of both phases is registered and compared. Results of the tests are reported in Table 3. It is worth noting that for all experiments the selection time of a resource is markedly reduced. Moreover, in the interviews, users have asserted that they selected resources that were not foreseen.

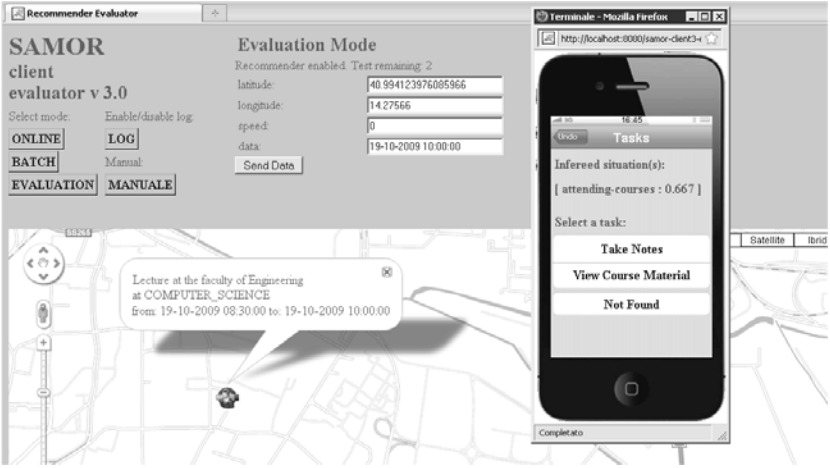



**Figure 33 – The auxiliary web application to evaluate the recommender.**

**Table 3. Selection time for a resource.**

| **Case Study** | **User** | **Selection time without recommender (seconds)** | **Selection time with recommender (seconds)** |
|---|---|---|---|
| Pharmaceutical Consultant | #1 | 15.0 | 8.6 |
| | #2 | 13.7 | 7.1 |
| | #3 | 13.2 | 9.8 |
| | #4 | 20.1 | 12.4 |
| | #5 | 16.7 | 8.5 |
| | **Average** | **15.74** | **9.28** |
| Off-Site Student | #1 | 10.7 | 5.6 |
| | #2 | 6.2 | 5.4 |
| | #3 | 17.1 | 13.6 |
| | **Average** | **11.33** | **8.20** |

Finally, we have evaluated the response time of the system for each recommendation. In a system equipped with an Intel Core 2 Duo Processor 2.2 GHz, with 3 GB DDR2 of RAM, the average response time of the system is 0.932 seconds, which guarantees a soft real time response to the user needs.

## 7.1. Comparison with other Context-Aware Recommenders

Recently, other context-aware recommenders have been proposed in the literature. Luther *et al.* [LFWK08] have integrated a situational reasoning engine into a mobile service recommendation, using an approach based on the standard representation language OWL. Weißenberg *et al.* [Weiß06] have proposed a demand-driven personalized service recommender, based on user profiles, semantic service, context- and situation-awareness. Goix *et al.* [GVCF07] have introduced a rule-based approach for inferring situations of mobile users, considering context data collected from heterogeneous and distributed sources. Unlike our recommender, all the three approaches do not consider the inescapable uncertainty that affects contextual data in order to infer the correct user situation. It follows that these approaches cannot adequately manage concurrent situations. Further, they cannot handle the gradual recognition of a situation. Since the papers which introduce the three recommenders propose no evaluation of them in terms of responsiveness, no comparison is possible with respect to this dimension. However, when we added the fuzziness to our recommender, we verified that the gradual recognition of situations had considerably increased the capability of our recommender to react proactively. Thus, we expect that our recommender may outperform the other recommenders in terms of responsiveness.

Also, in the situation inference process, the four recommenders use different approaches. In Luther *et al.*'s recommender, the situation inference is performed by applying dynamic assertional classification of contextual entities such as the location, the time and the neighbor people. Classification is carried out directly into OWL DL by subsumption. In Weißenberg *et al.*'s recommender, situation inference is implemented by means of F-Logic [KiLW95], a formalism that allows complex rules with high-level of expressiveness. However, F-Logic is generally undecidable. In Goix *et al.*'s recommender, context is modelled by ContextML, a proprietary XML-based Context Markup Language, whereas situation inference is performed by RuleML [RMI01], an XMLbased standard language to tackle the much broader problem of rule interchange. In our recommender, as we have already pointed out, situation inference is carried out by means of SWRL, a OWL-specific standard language that provides a formally sound way of inferring information in OWL ontologies, offering an officially standardized rule formalism for the Semantic Web.

Finally, in our recommender, the context model is separated in upper and domain-specific ontologies. This approach enhances reusability in different domains, enabling a truly general-purpose recommender, easy to adapt to different use scenarios. Among the compared systems, only the Weißenberg *et al.*'s recommender employs a similar approach to context modelling.

# Chapter 8

# Conclusion

## 8.1. Conclusion

Recognizing a situation in which a user is involved leads to better identify his demand at a certain time. In this thesis, a robust and general approach for managing situation awareness is proposed. Situation is defined as a logical conjunction of contextual conditions. Domain knowledge is expressed by means of ontologies and semantic rules, in order to guarantee portability, integration and extensibility. In this way, software agents that administrate their own contextual sources can easily communicate each other. Moreover, the overall system can rely on a formal representation avoiding inconsistency of the knowledge base. Contextual conditions can be affected by uncertainty, due to inaccuracy in sensor measurements or human imprecision in expressing concepts. For instance, the condition *user is close to a place* can represent a person who is a few dozen meters from a place as well as a person who is a few kilometers from the same place, depending on the personal feel of the observer. Fuzzy logic theory is employed to effectively manage the uncertainty, enabling a richer expressiveness in the contextual conditions. Thus, a rule base which combines fuzzy and semantic technologies has been developed.

More specifically, two overall architectures are proposed. In the first one, (Chapter 4), fuzzy logic and web ontology have been efficiently integrated thanks to an intermediate module based on the *observer* pattern. The architecture is based on two important modules, i.e., the *fuzzy engine,* which analyzes real-world inaccurate information, and the *semantic engine,* which contains the resource recommendation ontology and the related semantic rules. The other architecture (Chapter 5) aims to combine 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. Unlike the previous architecture, here fuzziness is directly managed within the semantic rules and the semantic inference engine rather than by a specific fuzzy inference engine.

However, in both cases, using predefined rules to infer situations leads to not completely satisfactory results. Indeed, users have different habits that may affect the way in which situations arise. Moreover, the same user can change his behavior over time, e.g., becoming a latecomer when he is always been a punctual person. Systems based on predefined rules force users to waste time in reconfiguring such rules, in order to adapt the system to their personal habits. Hence, we have proposed another architecture (Chapter 6) in which context history is considered as a powerful source of information about user behavior. By means of genetic algorithms, the rule base is automatically tuned to fit the actual behavior of the user, increasing the accuracy and responsiveness of the situation assessment. More specifically, we have presented a method based on a GA to adapt a resource recommender to the behavior of the specific user. This allows increasing the accuracy of the recommender in determining the user situation, thus improving the effectiveness and reliability in suggesting the correct resources to the user. The recommender exploits fuzzy linguistic variables to manage the inherent vagueness of some contextual parameters. The GA tunes the meaning of these linguistic variables on the basis of context history collected by tracking the behavior of the user when interacting with the mobile device.

Finally, real evaluation case studies concerning resource recommendations have been provided. The study has been focused on (i) a *pharmaceutical consultant* in typical business situations, (ii) an *off-site university student*, who performs a daily travel to go to university and return, and (iii) a *peddler,* who participates to markets and fairs in order to sell goods. A prototype has been implemented and configured for the case studies and simulation results enable to assess the reliability and effectiveness of the proposed approach. In particular, the results have shown that the GA enhances the performance of the recommender, increasing its responsiveness and modeling capabilities. Moreover, the user acceptance of the system has been tested, confirming that our framework can significantly improve the interaction between users and machines. Lastly, a comparison with other systems proposed in the literature is provided, to give a concrete and comparative view of the system, and to assess its reliability and responsiveness.

## 8.2. Future Work

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 problem of building an accurate context history. Indeed, the first data to be recorded in the context history is the actual situation in which the user is involved during his interactions with the system. A possible simple approach is to ask the user to declare the beginning and the end of each situation as he uses the mobile device. The process should be done in the first week of adoption of the system, in order to allow collecting a sufficient amount of training data (a few hundreds of samples for each situation to be inferred). Of course, in this case, an initial effort of the user is required. Instead, another approach that we are investigating requires no intervention of the user. A number of indicators have to be selected as proxy for the user situation, depending on the real service provided to the user by the system. For instance, good indicators could be particular resources that users need only in determined situations. In this case, when the user selects a predefined resource, the system automatically registers the associated situation and updates the context history. More specifically, the system registers when the user starts a proxy resource for a situation, and then waits the start of a proxy resource for the next situation. When the second proxy resource has been launched, the system infers the transition from the first situation to the other one as the average of the time instants of the starts of the two proxy resources.

# References


[AbBC98] D. Abrams, R. Baecker, and M. Chignell, *Information archiving with bookmarks: personal web space construction and organization.* In Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '98), pp. 41−48, ACM Press, 1998.

[AnHa08] C. Anagnostopoulos, and S. Hadjiefthymiades, *Enhancing situation-aware systems through imprecise reasoning.* In IEEE Transactions on Mobile Computing, Vol. 7, No. 10, pp. 1153-1168, 2008.

[AnHa10] C. Anagnostopoulos and S. Hadjiefthymiades, *Advanced fuzzy inference engines in situation aware computing*. In Fuzzy sets and systems, Vol. 161, pp. 498-521 (2010).

[Babu98] R. Babuska, *Fuzzy modeling for control.* In Kluwer Academic Publishers, 1998. ISBN 0792381548.

[BaHS05] F. Baader, I. Horrocks, and U. Sattler, *Description Logics as Ontology Languages for the Semantic Web.* In Mechanizing Mathematical Reasoning, Lecture Notes in Artificial Intelligence, pp. 228-248, Springer, 2005.

[Bake87] J.E. Baker, *Reducing bias and inefficiency in the selection algorithm*. In Proc. of the Second International Conference on Genetic Algorithms and their application, pp. 14-21, 1987.

[BaNu03] Baader, F. and Nutt, W., Basic description logics, in The Description Logic Handbook, pp. 43-95, Cambridge University Press (2003), ISBN 0521781760

[BBH+10] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathanf, and D. Riboni, *A survey of context modelling and reasoning techniques.* In Pervasive and Mobile Computing, Vol. 6, pp. 161-180, 2010.

[BeHL01] T. Berners-Lee, J. Hendler, and O. Lassila, *The semantic Web.* In Scientific American, 2001.

[Bern01] T. Berners-Lee, *WWW past & future*, 2003. Available at http://www.w3.org/2003/Talks/0922-rsoc-tbl/. Accessed April 2011.

[BGSV06] S. Bloehdorn, O. Görlitz, S. Schenk and M. Völkel, *TagFS, tag semantics for hierarchical file systems*. In Proc. 6th Int. Conference on Knowledge Management (IKNOW' 06), 2006.

[BoDG09] F. Bobillo, M. Delgado and J. Gomez-Romero, *Crisp representations and reasoning for fuzzy ontologies.* In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 17, No. 4, 2009.

[Bors97] W.N. Borst, *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* PhD Thesis Enschede, The Netherlands, 1997. ISBN: 90-365-0988-2.

[BoSt08] F. Bobillo and U. Straccia, *FuzzyDL: An expressive fuzzy description logic reasoner.* In Proc. 2008 International Conference on Fuzzy Systems (FUZZ-IEEE 2008), Hong Kong, June 2008.

[BoSt09] F. Bobillo and U. Straccia, *An OWL Ontology for fuzzy OWL 2.* In J. Rauch *et al.* (Eds.): ISMIS 2009, LNAI 5722, pp. 151–160, Springer-Verlag, Berlin, Heidelberg, 2009.

[ByCh03] H.E. Byun and K. Cheverst, *Supporting proactive intelligent behaviour: the problem of uncertainty*. In Proc. of the Workshop on User Modeling for Ubiquitous Computing (UM03), Johnstown, PA, pp 17-25, 2003.

[ByCh04] H.E. Byun, and K. Cheverst, *Utilizing Context History to Provide Dynamic Adaptations.* In Applied Artificial Intelligence, Vol. 18, No 6, pp. 533-548, 2004.

[Cale06] Google Inc., *Google Calendar API.* Released on April 2006, available at http://code.google.com/apis/calendar/, accessed April 2011.

[CCDG05] J. Coutaz, J.L. Crowley, S. Dobson, and D. Garlan, *Context is key*. In Communications of the ACM, Vol. 38, No. 3, pp. 49-53, 2005.

[CCG+06] F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, and I. Torre, *Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide.* In AI Communications, Vol. 19, No. 4, pp. 369–384, 2006.

[CCLM09] A. Ciaramella, M.G.C.A. Cimino, B. Lazzerini, and F. Marcelloni, *Situation-aware mobile service recommendation with fuzzy logic and semantic web.* In Proc. IEEE International Conference on Intelligent Systems Design and Applications (ISDA'09), pp. 1037–1042, 2009.

[CCLM10] A. Ciaramella, M.G.C.A. Cimino, B. Lazzerini, and F. Marcelloni, *A situation-aware resource recommender based on fuzzy and semantic web rules.* In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 18, No. 4, pp. 411-430, 2010.

[CGH+04] O. Cordòn, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, *Ten years of genetic fuzzy systems: current framework and new trends*. In Fuzzy sets and systems, Vol. 141, No. 1, pp. 5-31, Elsevier, 2004. ISSN 0165-0114.

[ChIK99] E. Chavez, R. Ide, and T. Kirste, *Interactive applications of personal situation-aware assistants*. In Computers & Graphics, vol. 23, n. 6, pp. 903-915, Elsevier, 1999.

[ChKo00] G. Chen and D. Kotz, *A Survey of Context-Aware Mobile Computing Research*. In Dart-mouth Computer Science Technical Report TR2000-381 (2000).

[Cing10] P. Cingolani, *JFuzzyLogic, a Java package that implement Fuzzy Control Language (FCL) specification (IEC 1131p7)*, current version 2.0.7 released on April 2010, available at http://jfuzzylogic.sourceforge.net, accessed April 2011.

[Cord01] O. Cordón, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. In World Scientific Pub. Co. Inc., 2001. ISBN 9810240171.

[CXC+05] J. Cao, N. Xing, A.T.S. Chan, Y. Feng, and B. Jin, *Service adaptation using fuzzy theory in context-aware mobile computing middleware*. In Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05), IEEE Computer Society, 2005, ISSN 1533-2306.

[DaPi02] M. Dahchour and A. Pirotte, *The semantics of reifying n-ary relationships as classes.* In proc. of 5th International Conference on Enterprise Information Systems (ICEIS-02), pp. 580–586 (2002)

[DCKF03] L. Ding, H. Chen, L. Kagal and T. Finin, *Public Address Ontology*, released on October 2003, available at http://daml.umbc.edu/ontologies/ittalks/address, accessed June 2009.

[DeAb00] A.K. Dey and G.D. Abowd, *Towards a better understanding of context and context-awareness*. In Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, ACM Press, 2000.

[Dey01] A.K. Dey, *Understanding and using context.* In Personal and ubiquitous computing, Vol. 5, No. 1, pp. 4-7, Springer-Verlag, 2001.

[DKD+05] L. Ding, P. Kolari, Z. Ding, S. Avancha, T. Finin, and A. Joshi, *Using Ontologies in the Semantic Web: A Survey.* In TR-CS-05-07 UMBC, 2005.

[DrHR96] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An introduction to fuzzy control.* Springer Verlag, 1996. ISBN 3540606912.

[EsSc93] L.J. Eshelman, and J.D. Schaffer, *Real-Coded Genetic Algorithms and Interval-Schemata.* In Foundations of Genetic Algorithms 2, Morgan Kaufman Publishers, San Mateo, pp. 187-202, 1993.

[Figg04] S. Figge, *Situation-dependent services — a challenge for mobile network operators.* In Journal of Business Research, Elsevier, Vol. 57, Issue 12, pp. 1416−1422, 2004.

[FNFK05] Y. Fukazawa, T. Naganuma, K. Fujii, and S. Kurakake, *A framework for task retrieval in task-oriented service navigation system.* In R. Meersman, Z. Tari, and P. Herrero (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 876–885. Springer, Heidelberg, 2005.

[GPX02] Topografix, *GPS Exchange Format,* standard released on 2002. Available at http://www.topografix.com/gpx.asp, accessed April 2011.

[Grub93] T.R. Gruber, *A translation approach to portable ontology specifications.* In Knowledge Acquisition, Vol. 5, No. 2, pp. 199-223, 1993. ISSN 1042-8143.

[GuPZ04] T. Gu, H.K. Pung, and D.Q. Zhang, *A bayesian approach for dealing with uncertain contexts.* In Proc. of the Second International Conference on

Pervasive Computing, Vienna, Austria, Austrian Computing Society, May 2004.

[GVCF07] L. W. Goix, M. Valla, L. Cerami and P. Falcarin, *Situation inference for mobile users: a rule based approach.* In Proc. International Conference on Mobile Data Management (MDM '07), IEEE Computer Society, pp. 299−303, 2007.

[Hans06] F. A. Hansen, *Context-aware mobile hypermedia: concepts, framework, and applications,* PhD Thesis, University of Aarhus, 2006.

[Harm06] F. Van Harmelen, *Fuzzy logic and the semantic web.* In E. Sanchez (eds.), Capturing Intelligence, Elsevier, 2006.

[HDCL07] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez, *An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments.* In IEEE Transactions on Fuzzy Systems, Vol. 15, No. 1, pp. 41-55, 2007.

[HeKo01] J. L. Herlocker and J. A. Konstan. *Content-independent task-focused recommendation.* In IEEE Internet Computing, pp. 40-47, 2001.

[HePu06] K. Heinonen and M. Pura, *Classifying mobile services.* In Proc. Helsinki Mobility Roundtable, Working Papers on Information Systems, Sprouts, Vol. 6, Art. 42, 2006.

[Herr08] F. Herrera, *Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects.* In Evolutionary Intelligence, Vol. 1, pp. 27–46, Springer-Verlag, 2008.

[HKZG08] P.D. Haghighi, S. Krishnaswamy, A. Zaslavsky, and M.M. Gaber, *Reasoning about context in uncertain pervasive computing environments.* In Proc. Smart Sensing and Context: Third European Conference (EuroSSC 2008), Zurich, Switzerland, October 29-31, 2008.

[Holl75] J.H. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975. ISBN 0472084607.

[HoPa04] I. Horrocks and P. F. Patel-Schneider, *A proposal for an OWL rules language.* In Proc. of the Thirteenth International World Wide Web Conference (WWW 2004), New York City, NY, 2004.

[HSKK09] J. Hong, E.-H. Suh, J. Kim, and S.Y. Kim, *Context-aware system for proactive personalized service based on context history.* In Expert Systems with Applications, Elsevier, Vol. 36, No 4, pp. 7448-7457, 2009.

[Jena10] HP Labs Semantic Web Programme, *Jena – A Semantic Web Framework for Java.* Current version 2.6.4, released on December 2010, available at http://jena.sourceforge.net/, accessed April 2011.

[KaAl05] M. Kaya, and R. Alhajj, *Genetic algorithm based framework for mining fuzzy association rules.* In Fuzzy Sets and Systems, Vol. 152, No. 3, pp. 587-601, 2005.

[Kalm84] D. Kalman, *The maximum and minimum of two numbers using the quadratic formula.* In The College Mathematical Journal Vol. 15, No. 4, 329–330 (1984)

[KaMa01] R. Kanawati and M. Malek, *Cowing: a collaborative bookmark management system.* In Proc. International Workshop on Cooperative Information Agents (CIA '02), Lecture Notes in Artificial Intelligence, No. 2828, pp. 34−39, Springer-Verlag, 2001.

[KiLW95] M. Kifer, G. Lausen and J. Wu, *Logical foundations of object-oriented and frame-based languages.* In Journal of the ACM, Vol. 42, No. 4, pp. 741-843, 1995.

[Klir06] G.J. Klir, *Uncertainty and information: foundations of generalized information theory.* Wiley Online Library, 2006. ISBN 0471748676.

[KlMP00] E.P. Klement, R. Mesiar, and E. Pap, *Triangular Norms. Trends in Logic.* Studia Logica Library. Kluwer Academic Publishers, Dordrecht, 2000.

[KMK+03] P. Korpipää, J. Mantyjarvi, J. Kela, H. Keranen, and E.J. Malm, *Managing context information in mobile devices.* In IEEE Pervasive Computing, Vol. 2, No. 3, pp. 42-51, 2003.

[LFWK08] M. Luther, Y. Fukazawa, M. Wagner and A. Kurakake, *Situational reasoning for task-oriented mobile service recommendation.* In The Knowledge Engineering Review, Cambridge University Press, Vol. 23, pp. 7−19, 2008.

[LiSe00] H. Lieberman, and T. Selker, *Out of context: Computer systems that adapt to, and learn from, context*. In IBM System Journals, Vol. 39, pp. 617-632, 2000.

[López99] M. F. López, *Overview of methodologies for building ontologies.* In Proc. Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, August 2, 1999.

[LuSt06] T. Lukasiewicz and U. Straccia, *An overview of uncertainty and vagueness in description logics for the semantic web.* In Technical Report INFSYS RR-1843-06-07, Institut für Informationssysteme, Technische Universität, Wien, 2006.

[LuSt07] T. Lukasiewicz and U. Straccia, *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.

[Maps05] Google Inc., *Google Maps API.* Released on June 2005, available at: http://code.google.com/apis/maps/, accessed April 2011.

[MäSe03] J. Mäntyjärvi and T. Seppänen, *Adapting applications in handheld devices fuzzy context information.* In Interacting with Computers, Vol. 15, pp. 521-538, 2003.

[Mayr05] R. Mayrhofer, *An architecture for context prediction.* In Trauner Verlag, Schriften der Johannes-Kepler-Universität Linz, volume C45, April 2005.

[Mich94] Z. Michalewicz, *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Springer-Verlag, 1994. ISBN 0387580905.

[MiKC08] J.K. Min, H.T. Kim, and S.B. Cho, *Social and Personal Context Modeling for Contact List Recommendation on Mobile Device.* In Proc. of International Conference on Web Intelligence and Intelligent Agent Technology, pp. 381-384, IEEE, 2008.

[Mitc98] M. Mitchell, *An introduction to genetic algorithms*. The MIT press, 1998. ISBN 0262631857

[MoSS04] B. Motik, U. Sattler, and R. Studer. *Query answering for OWL-DL with rules.* In Proceedings of the 2004 International Semantic Web Conference (ISWC 2004), pp. 549-563, 2004.

[NaKu05] T. Naganuma and S. Kurakake, *Task knowledge based retrieval for service relevant to mobile user's activity.* In Proc. of 4th International Semantic Web Conference (ISWC 2005), Lecture Notes in Computer Science, Vol. 3729, pp. 959–973, Springer, 2005.

[Noki09] Nokia Research Center, *Location, Context and Mobile Services: The Vision*. In Nokia Technology Insights Series, electronically available at: http://research.nokia.com, Nokia Corporation, 2009.

[NOST02] S. Nakajima, S. Oyama, K. Sumiya and K. Tanaka, *Context-dependent web bookmarks and their usage as queries,* In Proc. International Conference on Web Information Systems Engineering (WISE '02), Systems Engineering, pp. 333-341, 2002.

[OWL04] W3C, *OWL Web Ontology Language Reference.* W3C Recommendation, 10 February 2004, electronically available at http://www.w3.org/TR/owl-ref/, accessed April 2011..

[OWL09] W3C, OWL 2 Web Ontology Language, W3C Recommendation 27 October 2009, electronically available at http://www.w3.org/TR/owl2-overview/. Accessed April 2011.

[PaYC06] H. S. Park, J.O. Yoo, and S. B. Cho, *A context-aware music recommendation system using fuzzy Bayesian networks with utility theory.* In International Conference on Fuzzy Sytems and Knowledge Discovery (FSKD'06), pp. 970–979, 2006.

[PeGo98] W. Pedrycz, and F. Gomide, *An introduction to fuzzy sets: analysis and design.* The MIT Press, 1998. ISBN 0262161710.

[Pell10] Clark&Parsia, *Pellet: OWL 2 Reasoner for Java.* current version 2.2.2, released on September 2010, available at http://clarkparsia.com/pellet/, accessed April 2011.

[PoLH09] C. Porcel, A.G. Lopez-Herrera, and E. Herrera-Viedma, *A recommender system for research resources based on fuzzy linguistic modeling.* In Expert Systems with Applications, Vol. 36, No 3, Part 1, pp. 5173-5183, 2009.

[Prot11] Stanford Center for Biomedical Informatics Research, Stanford University School of Medicine, *Protégé OWL and SWRL editor.* Current version 3.4.5, released on March 2011. Available at http://protege.stanford.edu, accessed April 2011.

[PSS+06] J.Z. Pan, G. Stoilos, G. Stamou, V. Tzouvaras, and I. Horrocks, *f-SWRL: A fuzzy extension of SWRL.* In Journal on Data Semantics, Special Issue on Emergent Semantics, Vol. 4090, pp. 28–46, 2006.

[PTVS08] H. Petry, P. Tedesco, V. Vieira, and A.C. Salgado, *ICARE: A context-sensitive expert recommendation system.* In Workshop on Recommender Systems (ECAI'08), pp. 53–58, Patras, Greece, 2008.

[RaAC04] A. Ranganathan, J. Al-Muhtadi, and R.H. Campbell, *Reasoning about uncertain contexts in pervasive computing environments.* In IEEE Pervasive Computing, Vol. 3, No. 2, pp. 62-70, 2004.

[RDF04] W3C, *Resource Description Framework (RDF),* 2004. Available at: http://www.w3.org/RDF/. Accessed April 2011.

[RIF05a] W3C, *RIF Working Group,* 2005. Available at http://www.w3.org/2005/rules/wiki/RIF_Working_Group. Accessed April 2011.

[RIF05b] W3C, *RIF Overview,* 2005. Electronically available at http://www.w3.org/2005/rules/wiki/Overview. Accessed April 2011.

[RMI01] The Rule Markup Initiative, *RuleML.* First preliminary version released on January 2001. Available at: http://ruleml.org/, accessed April 2011.

[SaYa06] E. Sanchez and T. Yamanoi, *Fuzzy ontologies for the semantic web.* H. L. Larsen *et al.* (eds.): FQAS 2006, LNCS, Vol. 4027, pp. 691−699, Springer, Heidelberg, 2006.

[ScAW94] B.N. Schilit, N. Adams, and R. Want, *Context-Aware Computing Applications*. In Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, 1994.

[ScBG99] A. Schmidt, M. Beigl, and H.W. Geller, *There is more to context than location*. In Proc. of Workshop on Interactive Applications of Mobile Computing, Computer & Graphics, Elsevier, 1999.

[SCGL05] G. Sun, J. Chen, W. Guo and K. J. R. Liu, *Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs.* In IEEE Signal Processing Magazine, Vol. 22, No. 4, pp. 12−23, 2005.

[Sema01] W3C, *Semantic Web Activity,* 2001. Electronically available at http://www.w3.org/2001/sw/, accessed April 2011.

[SKMA05] H. Si, Y. Kawahara, H. Morikawa, and T. Aoyama, *Stochastic approach for creating context-aware services based on context histories in smart home.* In Proc. of exploiting context histories in smart environments (ECHISE '05), pp. 3480-3495, 2005.

[SLYL09] D. Shin, J. Lee, J. Yeon, and S. Lee, *Context-Aware Recommendation by Aggregating User Context.* In IEEE Conference on Commerce and Enterprise Computing, pp. 423-430, 2009.

[SSSK06] G. Stoilos, N. Simou, G. Stamou and S. Kollias. *Uncertainty and the semantic web.* In IEEE Intelligent Systems, Vol. 21, No. 5, pp. 84–87, 2006.

[StBF98] R. Studer, V.R. Benjamins, and D. Fensel, *Knowledge Engineering: Principles and methods.* In Data & Knowledge Engineering, Vol. 25, pp. 161-197, Elsevier 1998.

[StLi04] T. Strang and C. Linnhoff-Popien, *A context modeling survey*. In Proc. First International Workshop on Advanced Context Modelling, Reasoning and Management (UbiComp '04), 2004.

[Stra05] U. Straccia, *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.

[Sure03] Y. Sure, *Methodology, tools & case studies for ontology based knowledge management,* Thesis Informatic Science (332pp.), Univeristy of Karlsruhe, 2003.

[Swoo04] University of Maryland, Baltimore County (UMBC), *Swoogle, semantic web search.* Released on 2004, http://swoogle.umbc.edu/, accessed April 2011.

[SWRL04] W3C, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML,* W3C Member Submission 21 May 2004, electronically available at http://www.w3.org/Submission/SWRL/. Accessed April 2011.

[Time06] W3C, *Public Time Ontology.* In W3C Working Draft 27 September 2006, http://www.w3.org/TR/owl-time. Accessed April 2011.

[TTH+09] Y. L. Tai, S. R. Tsai, G. H. Huang, C. M. Lee, L. J. Tsai, K. F. Ssu and S. J. Wey, *A label-based file system.* In Journal of Computers, Vol. 19, No. 4, January 2009.

[VaBa09] P. Vasant and N. Barsoum, *Hybrid genetic algorithms and line search method for industrial production planning with non-linear fitness function.* In Engineering Applications of Artificial Intelligence, Elsevier, Vol. 22, No 4-5, pp. 767-777, 2009.

[VGD+08] P. K. Vatturi, W. Geyer, C. Dugan, M. Muller and B. Brownholtz, *Tag-based filtering for personalized bookmark recommendations.* In Proc. of 17th ACM Conference on information and Knowledge Management (CIKM '08), ACM, pp. 1395−1396, 2008.

[Weis91] M. Weiser, *The Computer for the Twenty-First Century*. In Scientific American, Vol. 265, No. 3, pp. 94-104, 1991.

[Weis95] M. Weiser, *The Computer for the 21st Century.* In Scientific American, Vol. 272, pp.78-89, 1995.

[Weiß06] N. Weißenberg, *An ontology-based approach to personalized situation-aware mobile service supply*. In GeoInformatica, Vol. 10, No 1, pp. 55-90, Springer, 2006.

[WHFG92] R. Want, A. Hopper, V. Falcao, and J. Gibbons, *Active Badge Location System*. In ACM Transactions on Information Systems, Vol. 10, No. 1, pp.91-102, 1992.

[WMYM08] X. Wang, Z.M. Ma, L. Yan, and X. Meng. *Vague SWRL: A Fuzzy Extension of SWRL.* In Proc. of the 2nd International Conference on Web Reasoning and Rule Systems (RR2008), pp. 232-233, Springer-Verlag, 2008.

[WMYZ10] X. Wang, Z. Ma, L.Yan, and R. Zhao. *RIF-FRD: A RIF dialect based on fuzzy sets.* In Proc. of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 1912-1916, 2010.

[WZGP04] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung, *Ontology Based Context Modeling and Reasoning using OWL.* In Proc. of the Second IEEE Annual

Conference on Pervasive Computing and Communications Workshops, IEEE Computer Society, 2004.

[YaTP05] G.-E. Yap, A.-H. Tan, and H.-H. Pang, *Dynamically-optimized context in recommender systems.* In Proc. of the 6th International Conference on Mobile Data Management (MDM'05). ACM Press, New York, NY, USA, pp. 265-272, 2005.

[Zade08] L. A. Zadeh, *Is there a need for fuzzy logic?* In Information Sciences, No. 178, pp. 2751–2779, 2008.

[Zade65] L.A. Zadeh, *Fuzzy Sets.* In Information and Control, Vol. 8, pp. 338-353, 1965.

[Zade74] Zadeh, L.A., *On the analysis of large-scale systems*. Published in *Fuzzy sets, fuzzy logic, and fuzzy systems : selected papers by Lotfi A. Zadeh,* World Scientific Publishing Co., Inc. River Edge, NJ, USA,1996. ISBN 9810224222.

[Zade75] L.A. Zadeh, *Fuzzy logic and approximate reasoning,* In Synthese, Vol. 30, No. 3-4, pp. 407-428, 1975.

[Zade96] L.A. Zadeh, *Fuzzy Logic = computing with words.* In IEEE Transactions on Fuzzy Systems, Vol. 4, No. 2, pp.103-111, 1996.

[ZhBo08] J. Zhao and H. Boley. *Uncertainty Treatment in the Rule Interchange Format: From Encoding to Extension.* In Proc. of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008), Germany, 2008.