

f-SPARQL extension and application to support context recognition

C. De Maio, G. Fenza, D. Furno, V. Loia

Department of Computer Science & CORISA (Consorzio di Ricerca Sistemi ad Agenti)

University of Salerno

e-mail: cdemaio@unisa.it, gfenza@unisa.it, dfurno@unisa.it, loia@unisa.it

Abstract — Context aware computing as well as wearable and ubiquitous computing often attain with pattern recognition on incoming sensor data. Recognizing more (useful) contexts requires more information about the context, and thus more sensors and better recognition algorithms. In order to enable logic inference on incoming data, the proposed work assumes that incoming data are represented by means of semantic languages (e.g., RDF, OWL, etc.). Nevertheless, in a context aware computing purely logic-based reasoning on context may not be enough. So, the work introduces soft computing techniques to approximate context recognition. Specifically, this paper introduces an approach to context analysis and recognition that relies on f-SPARQL[1] tool, that is a flexible extension of SPARQL. In particular, in this work a JAVA implementation of f-SPARQL and the integrated support for fuzzy clustering and classification are discussed. This tool is exploited in the architecture that foresees some task oriented agents in order to achieve context analysis and recognition in order to identify critical situations. Finally, a simple application scenario and preliminary experimental results have been described.

Keywords-component; fuzzy clustering; fuzzy classification; semantic web; semantic sensor data; context aware computing; f-SPARQL

I. INTRODUCTION

One of the aims of context aware computing is to build intelligent machines that provide automatic and autonomous support in critical activities. Accurate and comprehensive recognition of a context is an important step towards that goal. In order to understand context and develop applications that can be used for context recognition the first important step is to define context. Dey [2] defines context as information which for a simple information source is obvious and can be considered low level context but for several sources the definition of relevant information or high level context is not obvious. In the proposed work, we analyze how high level contexts can be generated by mashing the output of several different information sources and our intention is to define contexts as clusters of the data. Furthermore, clustering algorithms can be exploited in the classification of contexts and in the automatic exploitation of knowledge from data. As pointed out in [3], [4] many current context aware applications make unrealistic assumptions about the quality of the available context information. Typically these approaches, often based on ontologies [5], [6], make the inherent assumption that the lower-level context information, typically coming from sensors, is error-free. The idea proposed in this work consists

on the one hand to exploit semantic formalism to annotate sensor data for increasing interoperability as well as provide contextual information essential for situational knowledge. On the other hand, our proposed work is based on fuzzy clustering of semantic sensor data which can deal with very noisy, corrupted and uncertain information in a robust manner. Hence, it can be considered a generic approach to context recognition. Specifically, the paper presents an architecture for context recognition, aimed to support context aware in some application domains, that exploits synergy among intelligent agent technology, semantic web languages and soft computing techniques. In particular, the work defines an integrated environment aimed at identifying different level of contexts and their correlations. The goal is to define context-aware system whose quality of retrieved contexts relies on the acquisition of sensor data by means of a robust theoretical approach. So, the overall workflow is composed of two main steps: context training, by means of fuzzy context data analysis and interest context elicitation, by means of exploitation of semantic reasoning. The first phase of the workflow uses our extension of f-SPARQL 0. Furthermore we assume that sensor data are represented according to Semantic Sensor Web [7].

The paper is organized as follows: Section II presents an overview of some related works available in literature; Section III presents our extension of f-SPARQL; Section IV introduces the overall architecture and describes the complete workflow. Then, Section V describes a case study with the description of an application scenario and preliminary experimental results. Finally, conclusions and future works close the paper.

II. RELATED WORKS

In literature there are many contributions on context recognition applied to different fields. The context-recognition approach proposed by Nam et al. [8] and Young et al. [9] focus on filter fusion to support a robust face recognition under uneven illumination (image processing). Stäger et al. [10] provide an empirical design process for audio-based context recognition. Ma et al. [11] also propose an approach to recognize auditory scenes. Clarkson [12], [13] describes a context recognition based on audio-visual information from a wearable camera and microphone. Furthermore Schmidt [14] deals with development and prototyping of context-aware computing for ubiquitous computing environments.

To discover main trends or unusual patterns in context recognition applications, techniques of machine learning are extensively applied. For instance, in [15] and [16] the authors

enhance context aware recommendation systems by using neural networks and majority voting classifiers. Furthermore, in literature some works use fuzzy logic to enhance the ability to information retrieval, that is to search the correct information with imperfect information. A number of fuzzy query languages have been proposed in the scientific research. Takahashi [17] proposed the Fuzzy Query Language that is an improvement of the relational domain calculus. It provides a theoretical basis for the development of a human-oriented interface with relational databases. Another query language is the fuzzy SQL published by Bosc and Pivert [18]. This language allows gradual predicates interpreted in the framework of the fuzzy set theory. Then, Kacprzyk and Zadrozny [19] proposed FQUERY which extends Microsoft's database system Access in order to answer imprecisely specified questions. Finally, we cite the query language dealt in the paper that has inspired the present work, that is, F-SPARQL [1]. It is a flexible extension of SPARQL that introduces fuzzy set theory by allowing the occurrence of fuzzy terms and fuzzy relations. On the basis of fuzzy set theory and α -cut of fuzzy number, a set of translation rules is developed, converting f-SPARQL into crisp ones, so as to take advantage of existing implementations of SPARQL.

In this work we have defined an approach for context recognition, aimed to support context aware in several domains. In particular, we propose an extension of f-SPARQL able to perform both approximate querying and fuzzy clustering and classification of semantic sensor data.

III. F-SPARQL CLUSTERING AND CLASSIFICATION

This section deals with our implementation of f-SPARQL according to 0. In particular, we firstly discuss some additional features with respect to 0 and secondly present an extension of f-SPARQL for enabling fuzzy clustering and classification of semantic data.

A. f-SPARQL: Implementation

In this section, we discuss modification and implementation of f-SPARQL [1].

Our implementation is very close to approach described in 0. f-SPARQL has been implemented in JAVA, it can be interpreted as a wrapper to ARQ implementation of SPARQL. Pratically, it translates a query f-SPARQL into a standards crisp query language SPARQL, so that the query can be evaluated through the ARQ engine, and in the end, on obtained results are applied some transformations to achieve different goals.

In particular we have addressed two issues that were not discussed in the article 0, such as:

- how to define fuzzy sets;
- how to make classification.

To solve both issues we have been inspired by jFuzzyLogic (<http://jfuzzylogic.sourceforge.net/html/index.html>), a java package that implements the specifications of the Fuzzy Control Language (FCL).

Regarding the first issue, as defined in jFuzzyLogic, Piece Wise Linear, Trapezoidal, Triangular, Gaussian and Singleton function, are implemented. It is easy to add new ones, simply by implementing an abstract class, which requires the implementation of two methods: one that returns the degree of membership given a value x , and the other function returns the y value given a α -cut. To f-SPARQL grammar was added the keyword “FCL”, which has a similar meaning to *PREFIX*, but it identifies a FCL-like file in which Fuzzy Term are defined to be used later in the *FILTER* clause.

LISTING 1. EXAMPLE OF DEFINITION OF KEYWORD FCL

```
FCL service:<./fcl/service.fcl>
...
```

In Listing 1, we have defined an example of definition of prefix *service*, which locates the file “service.fcl”. So if in this file is described the fuzzy term “good” for the fuzzy set “estimate”, then we can use the *FILTER* clause as shown in Listing 2.

LISTING 2. EXAMPLE OF DEFINITION OF FILTER CLAUSE

```
FILTER (?estimate > very very
service:estimate:good WITH 0.1)
...
```

Furthermore, differently to original syntax of f-SPARQL, *WITH* parameter that identifies the α -cut value has been set for each fuzzy term.

The second issue concerns the individual score evaluation (i.e. how the individual meets the filters set), in order to achieve the classification. Again, for the resolution of the logical operators AND and OR, we have been made reference to FCL specifications, giving to the user the ability to set the desired method. Therefore, in order to accomplish this scope, we have added the keyword “*OPTION*” into f-SPARQL grammar. An example is shown in Listing 3. If the *OPTION* clause is omitted the default values of AND and OR operators are minimum and maximum, respectively.

In addition, the f-SPARQL package provides other several features, such as the possibility to depict functions graphs, or to realize the ontological clustering of individuals, as described in follow section.

LISTING 3. EXAMPLE OF DEFINITION OF KEYWORD OPTION

```
#top-k FQ# WITH ALL
OPTION AND:PROD OR:ASUM
...
```

B. f-SPARQL: Clustering and Classification

Our f-SPARQL implementation is based on Fuzzy c-means (FCM) algorithm [20]. In particular, we have considered N-dimensional centroids, where each dimension represents a Data Property corresponding to a feature on which we want to carry out clustering.

The clustering process involves two steps:

1. Identification of k centroids on dataset;
2. Classification of individuals according to centroids identified.

In order to accomplish the first step, in f-SPARQL we have defined a query, namely “#CQ#”. An example of this kind of query is shown in Listing 4:

LISTING 4. EXAMPLE OF #CQ# QUERY

```
#CQ# 2 2 0.1 50
PREFIX pre:<http://www.corisa.it/testDataset#>
ONPROPERTY pre:a pre:b
FROM <file://C:/UniSA/test.rdf>
```

The first line (i.e., #CQ# 2 2 0.1 50) defines the methodology to detect centroids, specifying the number of centroids to found, the fuzziness value, the accuracy to achieve and the maximum number of iterations if during the execution, the desired accuracy is not reached, respectively. In the example (see Listing 4), we want to cluster individuals in two classes, so we set 2 centroids, with fuzziness value 2, accuracy value 0.1, and a maximum number of iterations equal to 50. The *PREFIX* clause (as in the standard SPARQL query) associates short label with a specific URI. *ONPROPERTY* clause defines Data Property on which we want clustering. Furthermore, we can use the syntax “*ONPREOPERTY* *” in order to consider all data properties for clustering, when all individuals contain numeric values. Finally in the *FROM* clause we can specify the training dataset, on which we would retrieve the centroids.

ResultSet in output consist of centroids represented as ontological individuals, as shown in Listing 5:

LISTING 5. EXAMPLE OF RESULT SET

```
(?http://www.corisa.it/testDataset#b ="1.036"
^^xsd:double)
(?http://www.corisa.it/testDataset#a ="5.225"
^^xsd:double) -> [Root]
(?http://www.corisa.it/testDataset#b ="2.944"
^^xsd:double)
(?http://www.corisa.it/testDataset#a ="0.774"
^^xsd:double ) -> [Root]
```

Once obtained centroids, we are able to extract the fuzzy rules used for the fuzzy classification of individuals.

First, will need to define Fuzzy Sets and Fuzzy Term, so, for each cluster, we define functions from which to derive the membership degree. In particular, we define:

- as many fuzzy sets as the centroids’ dimensions;
- each fuzzy set will be described by many fuzzy terms as the number of clusters.

The resulting linear functions are obtained by calculating for each individual of training data set, the similarity with respect to each centroid dimension.

Let $sim_{ijp} = \frac{1}{1 + d_{ijp}}$, the similarity value, where d_{ijp} is

the Euclidean distance with respect to dimension p and the index i and j correspond to i -th individual and j -th centroid, respectively.

Following previous example in Figure 1. are shown the two fuzzy set, a and b, obtained.

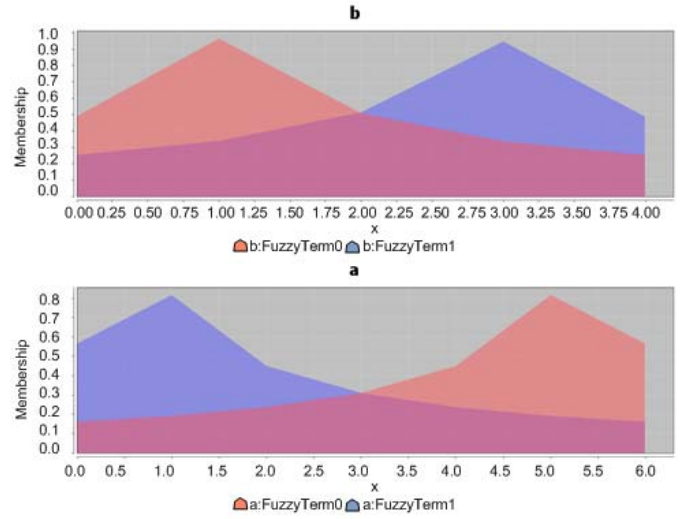


Figure 1. Example of Fuzzy set

These functions can be automatically designed using a static class *FCLwrite* (see Listing 6), which will create an FCL-like file.

LISTING 6. EXAMPLE OF FCLWRITE CLASS USAGE

```
fcl.FCLwriter.FCLwrite ("./ fcl / write.fcl"
centroidResult, anOntology, 2, "individual");
```

This takes as input:

- the path of the file to be created;
- the result set with centroids;
- a training data set, as an OntModel or ResultSet;
- the fuzziness value;
- the name of the output variable.

The output FCL file is described in Listing 7 highlighting in addition to fuzzy sets and fuzzy term, a block of rules (RULEBLOCK) was created concerning the clauses which must be fulfilled in order to assign an individual to a specific cluster. These rules will be translated into f-SPARQL filters enabling us to perform the fuzzy classification of individuals. In particular, will define a specific type of query composed of two parts, the first part in which are defined the classification rules, and the second part that specifies the individuals to classify. In the sample query, shown in Listing 8, we want to classify, with respect to the two input centroids, only those individuals whose value for the Date Property “a” is greater than 4. Furthermore, we want the result to be sorted in a decreasing way relatively belonging degree to cluster “0”. In particular we identify in the first part of query, the clause *CENTROIDS*, which will specify the value of fuzziness and the classification rules as filters f-SPARQL. On the other hand, the second part of query can be specified according to f-SPARQL.

LISTING 7. EXAMPLE OF FCL FILE CONTENT

```
// Data Property's NameSpace
DATAPROPERTY
  b := http://www.corisa.it/testDataset#
  a := http://www.corisa.it/testDataset#
END_DATAPROPERTY

// Define fuzziness value
FUZZINESS
  2.0;
END_FUZZINESS

// Define output variable 'individual'
VAR_OUTPUT
  individual;
END_VAR

// Fuzzify centroid dimension 'b'
FUZZIFY b
  TERM FuzzyTerm0 := (0.0, 0.631),(1.0, 0.706), ...
  TERM FuzzyTerm1 := (0.0, 0.282),(1.0, 0.392), ...
END_FUZZIFY

// Fuzzify centroid dimension 'a'
FUZZIFY a
  TERM FuzzyTerm0 := (0.0, 0.563),(1.0, 0.815), ...
  TERM FuzzyTerm1 := (0.0, 0.161),(1.0, 0.191), ...
END_FUZZIFY

// Centroids translateds in rules
RULEBLOCK
  RULE 0:IF((b IS FuzzyTerm0)AND(a IS FuzzyTerm0))
    THEN individual IS c0 ;
  RULE 1:IF((b IS FuzzyTerm1)AND(a IS FuzzyTerm1))
    THEN individual IS c1 ;
END_RULEBLOCK
```

LISTING 8. EXAMPLE OF QUERY DEFINITION

```
#CQ#
CENTROIDS {
  FUZZINESS: 2.0 .
  c0:((?b >= write:b:FuzzyTerm0 WITH 0.0)&&
    (?a >= write:a:FuzzyTerm0 WITH 0.0)).
  c1:((?b >= write:b:FuzzyTerm1 WITH 0.0)&&
    (?a >= write:a:FuzzyTerm1 WITH 0.0)).
  ORDER BY DESC(c0)
}
FCL write:<./fcl/write.fcl>
SELECT ?individual ?b ?a
FROM <file:./ontology/test.rdf>
WHERE {
  ?individual<http://www.corisa.it/testDataset#b>
  ?b WITH 1.0 .
  ?individual<http://www.corisa.it/testDataset#a>
  ?a WITH 1.0 .
  FILTER (?a > 4) .
}
```

IV. CONTEXT AWARE FRAMEWORK

This work proposes a context aware framework that combines semantic web technologies and soft computing techniques to support critical activities in several domains such as broadcasting of traffic or weather news, healthcare and so on. In particular the framework exploits agent paradigm to distribute tasks among the different components of the identified architecture. In Figure 2. an architectural overview of the context aware framework is depicted also involving the expected activities for the execution of the workflow. Hence, in

the following subsections a detailed description of architectural components and workflow activities is provided.

A. Architectural overview

As mentioned above this work proposes an approach to context recognition based on semantic modeling of sensor data, and f-SPARQL clustering and classification. In particular, we propose a qualitative querying on sensor data observations, usually characterized by a large amount of information difficult to manage and analyze, in order to extract useful knowledge. Figure 2. shows an architectural overview of the context aware framework with all components. Specifically, it is characterized by three main layers: *Context Sensing Domain*, *Context Aware Devices Domain* and *Context Recognition Domain*.

1) Context Sensing Domain:

In the Context Sensing Domain, different wireless sensor networks can be considered for the acquisition of context information according to different types of sensors. Specifically, we distinguish sensors that are placed on the human body and measure physical parameters of a person and sensors for the detection of environmental data. The first type of sensors, usually integrated in a wireless sensor network named Body Sensor Network (BSN), includes ECG Sensors, for a transthoracic (across the thorax or chest) interpretation of the electrical activity of the heart over a period of time; Blood Pressure Sensors, that is, non-invasive sensors designed to measure human blood pressure; Artificial Pancreas, to help people with diabetes automatically control their blood glucose level; EMG Sensors, for evaluating the electrical activity produced by skeletal muscles; Inertial Sensors, that detects and measures acceleration, tilt, shock, vibration, rotation, and multiple degrees-of-freedom (DoF); Pulse Oximetry Sensors for the detection of blood oxygen level and so on. A recent platform software framework for the design of Body Sensor Network (BSN) applications is represented by SPINE¹ (Signal Processing in Node Environment). On the other hand in the second category falls sensors for observations related to temperature, humidity, pressure, wind speed, precipitation rate and so on. This layer provides context data that are acquired, semantically annotated, analyzed and queried by Context Sensing Agents in the *Context Recognition Domain*.

2) Context Aware Devices Domain

People's actions can often be determined by their situation; in effect, there are certain things we do in different situations. The aim of manage contextual information refers to exploit this fact. Queries on contextual information can produce different results according to the context in which they are issued. Systems might learn contextual information in several ways: for example, acquiring data by sensors or simply by using global positioning (GPS) or both. In this layer fall all those devices that exploit context aware applications to provide the user with useful information in different situations. In particular, thanks to context recognition we aim to notify users with context-aware applications

¹ <http://spine.tilab.com/>

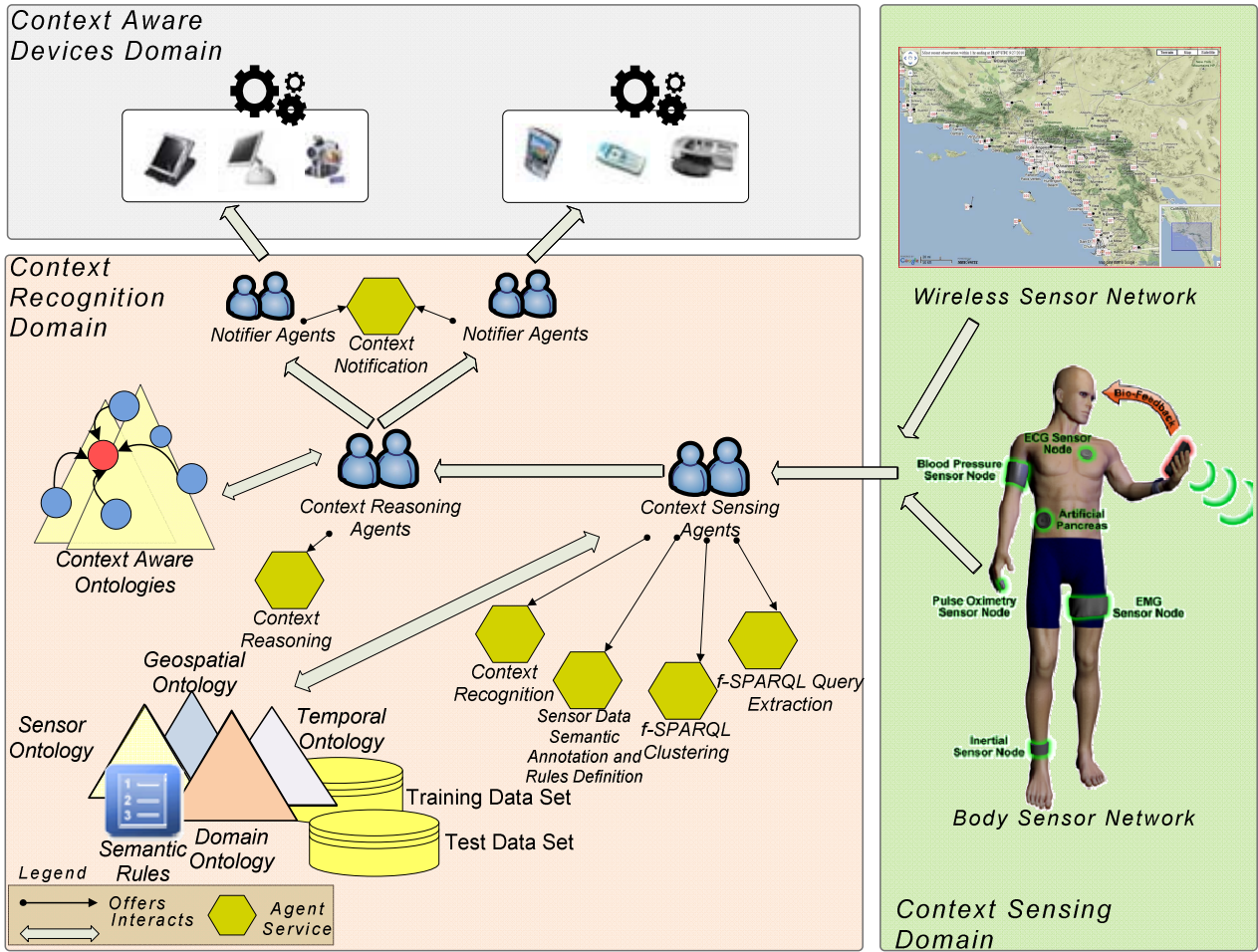


Figure 2. Context Aware Framework

installed on their devices, with meaningful context information. Some devices involved in this layer may be handheld devices, console in central command, video cameras, and so on. This layer receives interest context notifications by Notifier Agents in the *Context Recognition Domain*.

3) Context Recognition Domain

Context Recognition Domain is the main layer of the whole proposed framework. In particular it performs the activity of context recognition using f-SPARQL and context reasoning using semantic technologies. Furthermore this domain includes knowledge base consists of ontologies and databases on which elaboration and updating depend quality of system feedbacks. This layer is built on the agent-based platform JADE and the agents communication protocols are FIPA compliant. In more detail, the expected activities (services) in this layer are performed by the following task-oriented agents:

- **Context Sensing Agents:** these agents acquire raw data from sensors and, relatively to the training activity of workflow, perform the activities of Sensor Data Semantic Annotation and Rules Definition, f-SPARQL Clustering and f-SPARQL Query Extraction. Sensor

Data Semantic Annotation and Rules Definition concerns the activity of adding semantic annotations and inference rules to sensor data. Furthermore, on sensor data set these agents perform the activity of f-SPARQL Clustering in order to determine relevant patterns in data, also named prototypes or centroids. Finally, in the third step of the training phase of the workflow, these agents perform the activity of f-SPARQL Query Extraction, where approximate queries for a qualitative selection of data are designed thanks to clustering results obtained in the previous step. On the other hand, Context Sensing Agents perform activity of Context Recognition when the architecture is in exercise. Specifically, the Context Recognition activity deal with the execution of f-SPARQL queries extracted in the training phase. This activity allows to select data according to relevant patterns resulting from clustering and considering their inherent uncertainty. Therefore, each query allows to select data belonging to a context (cluster) with a certain membership degree.

- *Context Reasoning Agents*: these agents perform the activity of Context Reasoning in the execution phase of the workflow. In particular, this activity follows the activity of Context Recognition and relates to the exploitation of semantic rules in order to refine clustering and classification results. As a result, more meaningful contexts are extracted from the whole process.
- *Notifier Agents*: these agents perform the activity of Context Notification in the execution phase of the workflow. This activity follows the Context Reasoning activity and simply deals with the notification of context recognition process result, that is, interest context according to specific user or application needs.

B. Workflow

This section presents the workflow's phases in the proposed context aware framework. The whole workflow involves two main phases: Training Phase and Execution Phase. The first phase of the workflow concerns with Sensor Data Semantic Annotation, f-SPARQL Clustering and f-SPARQL Query Extraction activities. The second phase of the workflow refers to Context Recognition, Context Reasoning and Context Notification activities. Through the section all the aforementioned activities will be discussed in more detail.

1) Training Phase

This phase encompasses the activities of Sensor Data Semantic Annotation and Rules Definition, f-SPARQL Clustering and f-SPARQL Query Extraction. In particular, a synergic application of semantic technologies and soft computing techniques is carried out in order to go from raw sensor data to the extraction of approximate queries to support context recognition.

a) Sensor Data Semantic Annotation and Rules Definition

This first activity of the training phase concerns with the semantic modeling of raw sensor data extracted from different sources. This modeling is achieved according to standardization efforts of the Open Geospatial Consortium (OGC; www.opengeospatial.org) and Semantic Web Activity of the World Wide Web Consortium (W3C; www.w3.org/2001/sw/) to provide enhanced descriptions and meaning to sensor data. Specifically, sensors observations are often in binary or proprietary formats; therefore, metadata play an essential role in managing sensor data. So the adding of a semantically rich meaning to sensors would provide spatial, temporal, and thematic information essential for sensor data discovery, analysis and reasoning. In this work, we use RDFa a W3C proposed standard (www.w3.org/2006/07/SWD/RDFa/) in order to represent semantic metadata within an XML language from which it is possible to extract RDF triples using a simple mapping [7]. Furthermore, Jena inference engine is used to derive additional RDF assertions by using axioms and rules associated with the reasoner (<http://jena.sourceforge.net/inference/>). These rules are written by domain experts according to specific sensor observations. So, the main goal of this activity is to prepare data sets for

analysis aimed at the identification of relevant contexts carried out in the next steps. In more detail, to each sensor observation will be certainly associated a temporal information. So each set of semantically annotated sensor observations will have an associated timestamp, therefore, it is possible to uniquely determine the observations tuple (feature set) for each time instant or interval. Furthermore, all sets of semantically annotated sensor observations will be stored in appropriate RDF storages supporting SPARQL querying. Finally, the sets of sensor observations modeled and stored as mentioned above as well as inference rules will represent inputs for the other steps in the workflow.

b) f-SPARQL Clustering

In this activity of the training phase our proposed extension of f-SPARQL is exploited in order to extract relevant context from a first set of semantic sensor observations. In particular, our approach refers to the extraction of a representative individual (tuple of sensor observations) of each relevant context (cluster) from a first block of sensor data (training data set) using our proposed Java implementation of f-SPARQL based on fuzzy c-means clustering. Specifically, our idea is firstly to mine centroids representing relevant contexts from the training data set and secondly establish a fuzzy classification of individuals in the data set according to the previously identified centroids. This process, carried out using f-SPARQL, allow to determine a fuzzy set for each feature (property) of an extracted centroid and a fuzzy term for each cluster, that is, membership functions. Furthermore, membership functions and classification rules accordingly built will be also stored in a file representing fuzzy control accessible through the invocation of Java methods. More details regarding this activity can be found in Section III.

c) f-SPARQL Query Extraction

This activity refers to the automatic design of f-SPARQL queries from clustering results. Thanks to the proposed extension of f-SPARQL, it is quite simple to achieve this task. In fact, since it is possible access membership functions and rules representing cluster ranges resulting from previous activity, the steps to carry out in this activity consists in the translation of classification rules in f-SPARQL query filters. In particular, our Java implementation of f-SPARQL enables the achievement of this translation and the subsequent design of f-SPARQL queries by invoking appropriate Java methods. These queries will allow for individuals to obtain the Euclidean distances relatively to centroids, and then make the classification. Section III provides further details for this activity.

2) Execution Phase

This phase of the workflow refers to the run time of the proposed context aware framework. In particular, it involves Context Recognition, Context Reasoning and Context Notification activities.

a) Context Recognition

In this activity, the input data set is represented by a second block of semantic sensor observations, that is, the so called test data set. In particular, for this new data set it is not necessary execute again training activity but it is enough to carry out f-

SPARQL queries resulting from training phase. Since f-SPARQL queries are designed according to clustering results, in this phase, their execution enable us to properly select individuals belonging to a specific cluster with a certain degree. In other words, in this activity we recognize context in which falls each new individual and at what level. Furthermore, only individuals exceeding a certain degree of membership to a relevant context are filtered and further processed in the next activity of Context Reasoning.

b) Context Reasoning

This activity is based on the observation that clustering only considers numerical distances between relevant patterns and observed data, therefore, the accuracy in the context recognition can not be always optimal. So, the main aim here is to include a level of inference considering other context information not considered by clustering. In particular, this activity considers the application of context aware ontologies and the execution of semantic rules designed in a pre processing phase of the training workflow. Results of this activity will be send to *Notifier Agents* that execute the last activity of this phase, that is, Context Notification.

c) Context Notification

This activity refers to the notification of context awareness result deriving from the execution of the workflow. In particular, a further formatting of context information may be applied in this activity. Specifically, different context aware devices may be interested to different formats or protocols for notifications. So in this phase a definitive refinement on context aware result will be applied in order to satisfy the different needs of users and/or applications.

V. CASE STUDY

The generic context aware framework proposed in this work is suitable to be applied in several domains. The aim of this section is to discuss a simple case study in the domain of healthcare. So through the section an application scenario and preliminary experimental results in the aforementioned domain will be argued.

A. Application Scenario

The application scenario deals with context aware prediction of heart attack risk or stroke risk in some cases also accompanied by risk of fall from fainting. So it is crucial to recognize the symptoms as well as the speed of intervention. In our simplified scenario we consider a BSN with four sensors, that is, a Blood Pressure Sensor, an Artificial Pancreas, a Pulse Oximetry Sensor and an Accelerometer. The observations relating to the first three sensors will be used in the training phase of the workflow to identify the following contexts: normal situation, heart attack risk situation and stroke risk situation. On the other hand, the accelerometer sensor observation will be exploited in order to refine clustering result by adding the information concerning user position (i.e. sitting, standing). In fact, in most cases heart attack and stroke also causes falls due to fainting. Hence, in TABLE I. the classification rules for each recognized context extracted in the training phase of workflow. These classification rules will be translated in f-SPARQL filters as described in Section III and

enable us to perform approximate queries on semantic sensor observations. Furthermore, semantic rule-based reasoning can support clustering result refinement or can aid to infer adding information, for instance, by exploiting the context information concerning the user position.

TABLE I. CLASSIFICATION RULES

Rule	Context
IF Blood Pressure is <i>normal</i> AND Diabete is <i>normal</i> AND Blood Oxygen is <i>normal</i>	<i>Normal Situation</i>
IF Blood Pressure is <i>high</i> AND Diabete is <i>high</i> AND Blood Oxygen is <i>low</i>	<i>Heart Attack Risk Situation</i>
IF Blood Pressure is <i>very low</i> AND Diabete is <i>high</i> AND Blood Oxygen is <i>low</i>	<i>Stroke Risk Situation</i>

Listing 9 depicts an example of semantic rule definition for Fall From Fainting Risk Situation according to Jena inference support. So we can conclude that there are two level of notification in the proposed context aware framework, a first level deriving from f-SPARQL querying and a second level deriving from a combination between f-SPARQL querying results and semantic reasoning.

LISTING 9. EXAMPLE OF SEMANTIC RULE DEFINITION

```
[FallFromFaintingRiskRule:
  (?sit1 rdf:type ca:HeartAttackRisk),
  (?sit1 ca:timestamp ?sit1Timestamp),
  (?sit1 ca:relevantIndividual
    ?SensorObservationX),
  (?sit2 rdf:type ca:StandingSituation),
  (?sit2 ca:timestamp ?sit2Timestamp),
  (?sit2 ca:relevantIndividual
    ?SensorObservationY),
  Equal(?sit1Timestamp,?sit2Timestamp),
  makeTemp(?sit)
  ->
  (?sit rdf:type ca:FallFromFaintingRisk),
  (?sit ca:timestamp ?sit1Timestamp),
  (?sit ca:relevantIndividual
    ?SensorObservationX),
  (?sit ca:relevantIndividual
    ?SensorObservationY),
  (?sit ca:involves ?sit1),
  (?sit ca:involves ?sit2) ]
```

B. Experimentaental results

This section presents the results of the analysis conducted for the evaluation of the context aware framework proposed in this work. The proposed framework was developed in Java with the support of JADE for the agent architecture. Specifically, in this preliminary experimentation we evaluated the following performance measures:

- *Accuracy*: assessed by the number of “false positives” detected by the system;
- *Completeness*: evaluated as the ratio between the total number of risk situations detected by the system and the total number of real risk situations.

Furthermore, data sets used for the experimentation are shown in TABLE II. where we can distinguish between a semantic data set used for the training phase of the workflow (i.e. BSN_Training_Data_Set.rdf) and a semantic data set employed for the execution phase of the workflow (i.e.

BSN_Test_Data_Set.rdf). The preliminary experimentation performed by evaluating the aforementioned performance measures and taking into account the number of situations of risk present in the test data set described above provided the results shown in TABLE III.

TABLE II. DATA SETS DESCRIPTION

<i>Data Set</i>	<i>Num. Observations</i>	<i>Num. Risk Situations</i>
BSN_Training_Data_Set.rdf	1743	622
BSN_Test_Data_Set.rdf	8070	3632

TABLE III. PERFORMANCE MEASURE EVALUATION

<i>Accuracy</i>		
<i>Num. Risk Situations</i>	<i>Num. "False Positive"</i>	<i>accuracy (%)</i>
3632	388	89,31
<i>Completeness</i>		
<i>Num. Risk Situations</i>	<i>Num. "True Positive"</i>	<i>completeness (%)</i>
3632	3520	96,91

Therefore, we can conclude that these first experimental results indicate the goodness of the proposed approach in terms of accuracy and completeness. Obviously, further experimental results considering other measures and metrics as well as comparison against other similar methods are necessary to establish its real applicability in several critical domains.

VI. CONCLUSIONS AND FUTURE WORKS

This work proposes a generic approach for context recognition. The approach is hybrid since it involves the use of semantic technologies (i.e. semantic annotation of sensors data, f-SPARQL, reasoning) and the use of soft computing techniques (i.e. fuzzy clustering and classification). Through a sequence of phases the system trains itself by means of f-SPARQL unsupervised clustering aimed at recognizing of specific contexts. The use of ontologies is fundamental for the usability of the system, since they enable the semantic description of sensor data, the inter-relationships among concepts and the f-SPARQL querying. On the other hand, fuzzy clustering of semantic sensor data can deal with very noisy, corrupted and uncertain information in a robust way. Final result is a generic context aware framework able to provide notifications which appropriately meet context requirements. The experimental results address the conclusion that in order to achieve accurate and complete context recognition there is a need to properly model and identify the context requirements in the different phases. Future works foresee the application of the proposed framework in other application domains (e.g. broadcasting of traffic or weather news) and further extend f-SPARQL in order to include other useful operators and functionality and then enhance this interesting querying tool.

REFERENCES

- [1] Cheng, J.; Ma, Z.M.; Yan, L. (2010): f-SPARQL: A Flexible Extension of SPARQL. Proc. Intl. Conference on Database and Expert Systems Applications (DEXA), LNCS 6261, Springer.
- [2] Dey, A.K. and Abowd, G.D., "Towards a better understanding of context and context-awareness". In the CHI 2000 Workshop on The What, Who, Where, When, Why and How of Context-Awareness.
- [3] Henriksen, K., J.Indulska: "Modelling and using imperfect context information". In: Pervasive Computing and Communications Workshops(PERCOMW), 2004, 33-37.
- [4] Dey, A., Mankoff, J., Abowd, G., Carter, S.: "Distributed mediation of ambiguous context in aware environments". In: UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology, ACM Press, 2002, 121-130.
- [5] Korpipaa, P., Mantyjarvi, J.: "An ontology for a mobile device sensor-based context awareness". In: Proc. Context03. LNAI no. 2680, Springer-Verlag, 2003, 451-459.
- [6] Chen, H., Finin, T., Joshi, A.: "An ontology for context-aware pervasive computing environments". Knowledge Engineering Review 18, 2004, 197-207 Special Issue on Ontologies for Distributed Systems.
- [7] Sheth, A.P., Henson, C., Sahoo, S.S.: "Semantic sensor web". IEEE Internet Computing, Vol. 12, No. 4. (July 2008), pp. 78-83.
- [8] M. Y. Nam, M. R. Bashar, and P.-K. Rhee, "Adaptive feature representation for robust face recognition using context-aware approach," Neurocomputing, vol. 70, no. 4-6, pp. 648-656, Jan. 2007.
- [9] N. Young, M. Bashar, and P. Rhee, "Adaptive context-aware filter fusion for face recognition on bad illumination," in Knowledge-Based Intelligent Information and Engineering Systems. Berlin, Germany: SpringerVerlag, 2006, pp. 532-541.
- [10] M. Stäger, P. Lukowicz, and G. Tröster, "Power and accuracy trade-offs in sound-based context recognition systems," Pervasive Mob. Comput., vol. 3, no. 3, pp. 300-327, Jun. 2007.
- [11] L. Ma, D. Smith, and B. Milner, "Context-awareness using environmental noise classification," in Proc. Eurospeech, 2003, pp. 2237-2240.
- [12] Clarkson, B. & Pentland, A. 1999. "Unsupervised clustering of ambulatory audio and video". In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 6: 3037-3040.
- [13] Clarkson, B., Mase, K. & Pentland, A. 2000. "Recognizing user context via wearable sensors". In the digest of papers of the 4th Intl. Symposium on Wearable Computers: 69-75.
- [14] Schmidt, A. 2002. "Ubiquitous computing - computing in context". Ph.D. Thesis, Lancaster University, UK.
- [15] Biancalana, C., Flamini, A., Gasparetti, F., Micarelli, A., Millevolte, S., and Sansonetti, G. Enhancing traditional local search recommendations with context-awareness. In Proceedings of User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, 2011., pp. 335-340.
- [16] Biancalana, C., Gasparetti, F., Micarelli, A., Miola, A., and Sansonetti, G. Context-aware movie recommendation based on signal processing and machine learning. In Proceedings of the Second Workshop on Context-Aware Movie Recommendation, CAMRa '11, Chicago, USA, 2011.
- [17] Takahashi, Y.: "A Fuzzy Query Language for Relational Databases". In: Bosc, P., Kacprzyk, J.: Fuzziness in Database Management Systems. Physica Publisher, Heidelberg, 1995, pp. 365-384.
- [18] Bosc, P., Pivert, O.: "SQLf Query Functionality on Top of a Regular Relational Database Management System". In: Pons, O., Vila, M. A., Kacprzyk, J.: Knowledge Management in Fuzzy Databases. Physica Publisher, Heidelberg, 2000, pp. 171-190.
- [19] Kacprzyk, J., Zadrozny, S.: "Data Mining via Fuzzy Querying over the Internet". In: Pons, O., Vila, M. A., Kacprzyk, J.: Knowledge Management in Fuzzy Databases. Physica Publisher, Heidelberg, 2000, pp. 211-233.
- [20] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", New York, Plenum Press, 1981.