Human Activity Recognition using a Semantic Ontology-Based Framework

Rosario Culmone
Computer Science Division,
School of Sciences and Technologies,
University of Camerino
Email: rosario.culmone@unicam.it

Paolo Giuliodori
Computer Science Division,
School of Sciences and Technologies,
University of Camerino
Email: paolo.giuliodori@unicam.it

Michela Quadrini
Computer Science Division,
School of Sciences and Technologies,
University of Camerino
Email: michela.quadrini@unicam.it

Abstract—**In the last years, the extensive use of smart objects embedded in the physical world, in order to monitor and record physical or environmental conditions, has increased rapidly. In this scenario, heterogeneous devices are connected together into a network. Data generated from such system are usually stored in a database, which often shows a lack of semantic information and relationship among devices. Moreover, this set can be incomplete, unreliable, incorrect and noisy. So, it turns out to be important both the integration of information and the interoperability of applications. For this reason, ontologies are becoming widely used to describe the domain and achieve efficient interoperability of information system. An example of the described situation could be represented by Ambient Assisted Living context, which intends to enable older or disabled people to remain living independently longer in their own house. In this contest, human activity recognition plays a main role because it could be considered as starting point to facilitate assistance and care for elderly. Due to the nature of human behavior, it is necessary to manage the time and spatial restrictions. So, we propose a framework that implements a novel methodology based on the integration of an ontology for representing contextual knowledge and a Complex Event Processing engine for supporting timed reasoning. Moreover, it is an infrastructure where knowledge, organized in conceptual spaces (based on its meaning) can be semantically queried, discovered, and shared across applications. In our framework, benefits deriving from the implementation of a domain ontology are exploited into different levels of abstraction. Thereafter, reasoning techniques represent a preprocessing method to prepare data for the final temporal analysis. The results, presented in this paper, have been obtained applying the methodology into AALISABETH, an Ambient Assisted Living project aimed to monitor the lifestyle of old people, not suffering from major chronic diseases or severe disabilities.**

Keywords–Pattern Recognition; OntoAALISABETH Domain Ontology; Semantic Reasoning; Complex Event Processing (CEP).

I. INTRODUCTION

In the last years, the extensive use of smart objects embedded in the physical world in order to obtain information has increased rapidly. In other words, such sensor network allows to monitor and record physical or environmental conditions, especially interactions of users with the physical world. In order to reach this aim, the network is composed of a large and heterogeneous sources. The issue is that typical technologies for recording data do not allow to describe the relation of a sensor with the network. Furthermore, data can be incomplete, unreliable, incorrect, and could happen that one type of information is expressed by using different type of physical measures. So, to process these data generated by several heterogeneous sources, it turns out to be important both the integration of information and the interoperability of applications. In such scenario, these data are stored in a data repository, usually a Database (DB), which often shows a lack of semantic information and relationships among components of the system. So, acquired data from smart objects need to be treated according to their semantics. For this reason, in order to successfully monitor a situation, it is typically necessary to integrate stored data with static data and background knowledge. Ontologies represent a tool to connect these aspect. In fact, they provide a shared understanding of a domain, hence allowing semantic interoperability. This is the approach that we have presented into the initial proceeding [1].

An example of the described situation could be represented by an Ambient Assisted Living (AAL) context. The AAL program, promoted by the European Commission [2], intends to enable older or disabled people for the purpose of remaining living independently longer in their own house with an improved quality of life [3][4]. So, in the user domestic environment a wide network of smart objects is installed, whose task is to provide the possibility to monitor the user lifestyle. In order to reach this aim, the smart home (SH) relies on many different types of objects: from clinical devices for the user's health to indicators of presence, from temperature and humidity measurements to fridge and door opening sensors.

In this context, also at national and regional level [5], there are many on-going experiments among which AALISABETH (Ambient-Aware LIfeStyle tutoring for A BETter Health) [6], a project running in Regione Marche. This project has the objective to develop a new technology, based on the use of non-invasive sensor network, for monitoring the lifestyle of old people (65+), not suffering from major chronic diseases or severe disabilities. In particular, the main goal of this project is to detect a set of abnormal behaviors that could bring to the onset of the most common diseases. In particular, the same activity can acquire different meaning depending on the time of day. In order to reach this goal, a set of sensors has been selected and interconnected through an heterogeneous communicating network that wires the AALISABETH SH. Data collected from such variety of devices, weather environment, wearable or clinical sensors, are store in a proper database. In order to answer to the requirements of previous portrayed project, we develop a novel methodology, which is able to detect particular behaviours, compare them evolving with time, or determine the order in which events occurred. In situation monitoring, geo-spatial information is also of great importance, since it enables to locate events in the real world.

For these reasons, our methodology integrates an ontology for representing contextual knowledge with rule-based and a Complex Event Processing (CEP) engine for supporting the timed reasoning. It is an infrastructure where the knowledge, organized in conceptual spaces (based on its meaning) can be semantically queried, discovered, and shared across applications. The ontology is introduced because it is able to provide a shared understanding of a domain, hence allowing semantic interoperability. In addition, it has the ability to reuse knowledge and integrate several knowledge domains. Moreover, it is built following a pyramidal structure in order to distinguish two types of knowledge, static and dynamic. The former describes the domain, while the latter models the context acquisition, in particular sensor data, in order to describe the AAL domain, to organize data according to their semantic meaning and to select them during the pre-processing phase. On the other hand, CEP engine has been introduced in the proposed framework due to the expressiveness limitation of ontology, which lacks of temporal reasoning. In fact, traditional methods have not focused on reasoning over time and space, which is necessary to capture some of the important characteristics of streaming data and events. Moreover, the benefits of the framework can be noted in the Section IV, where a concrete case of use is presented. More specifically, the framework has been tested in a standard flat populated by an elderly person that has a regular behaviour.

The paper is structured as follows: Section II examines the related literature concerning the topics addressed in this work. Section III explains the motivation of the proposed methodology, in particular it provides a detailed description of the framework architecture. Section IV is entirely dedicated to implementation of framework with its different components, starting from data source, going trough semantic and ending with pattern recognition. At the end of this section a concrete example is provided, which allows to validate the proposed methodology. Section V contains the conclusion and some possible future development.

II. Related Work

Human activity discovery and recognition play an important role in a wide range of applications in the AAL domain in order to facilitate assistance and care for elderly. Moreover, they represent an active and ambitious research area because of the large amount of noise in data and the difficulty of modelling situations [7]. In addition, each human has different ways to perform the activity, but also people can do several activities at the same time, or different places may be needed to perform a particular activity.

In this scenario, methods to recognise human activity have been widely studied for long time and several approaches have been developed. They can be divided into three main categories: statistical, probabilistic and logic. The former is based on machine learning techniques, including both supervised and unsupervised human behaviour recognition. Moreover, there is a wide range of algorithms and models for human recognition based on statistical approach. For instance, Fleury et al. [7] classify human behaviors using a Support Vector Machines (SVM), however, Sharma et al. [8] proposes the designing of an artificial neural network (NN) for the classification of Human activity data received from an accelerometer sensor. This kind of technique leads to good performance, but the results are not easily interpretable [9]. Another suitable choice is a probabilistic approach. An illustration is given by Van Kasteren [10], in which Dynamic Bayesian networks are used to recognise activities. In particular, temporal probabilistic models have been shown to give a good performance in recognizing activities from sensor data, as shown in Patterson's work [11]. Because of the intrinsic nature of activities, a hybrid approach is often used, which combined boosting and learning an ensemble of static classifiers with Hidden Markov models (HMMs) to capture the temporal regularities and smoothness of activities. Moreover, in these two different approaches, it is difficult to take into account a previous high-level knowledge. This aspect could be easily introduced in a logical approach using an ontology, but the main lack of logical methods is the difficulty to manage uncertainty. To overcome any limit, the three approaches have been combined, by Getoor and Taskar [12], in Statistical Relation Learning (SRL) that integrates elements of logic and probabilistic models.

In our work, as introduced before, we propose a methodology to discover human activity that integrates ontology with CEP engine based on data stored in a database produced by a wide range of sensors. So, the proposed approach includes different areas of research: domain ontology, mapping database to ontology, semantic data pre-processing, pattern matching and discovery in data words.

Ontologies are commonly used to explicitly formalize and specify a domain of knowledge [13]. Furthermore, they improve the automation of integration of heterogeneous data sources, providing a formal specification of the vocabulary of concepts and their relationships as described in the Gagnon's work [14]. A wide literature on the use of ontologies for information integration over various domains is available. In particular, an ontology for smart home is defined by Bonino et al. [15] for formally expressing what they call the "domotic environment" (e.g., sensors, gateways and network), but also for supporting reasoning mechanisms. The reasoner allows to support automatic recognition of device instances and to verify the formal correctness of the model.

Other interesting works presenting ontologies for AAL activities are those by Mocholi et al. [16] and Gu et al. [17]. More specifically, in the last work the authors present ontology-based context model using OWL, which ontology is divided into upper and domain-specific ontologies. The former is a high-level ontology and it is able to describe general context knowledge about the physical world, whereas the latter defines the details of general concepts and their properties in each subdomain. Instead, for mapping an external database to a local ontology, we refer to techniques suggested by Sedighi [18] and Barrasa et al. [19]. In addiction, tools that automatically generate OWL ontologies [20] from database schemas have been also presented, for instance by Cullot et al. [21] and Rodriguez-Muro et al. [22]. Furthermore, ontologies may also support a semantic approach to applications involving Business Process Management (BPM) techniques and analyses of processes based on a list of recorded events, i.e., Process Mining. In this case, a possible procedure is to enrich the event logs coming from external data sources by using ontology based data integration, as observed by Tran Thi Kim and Werthner [23]. A similar methodology used to integrate semantic annotation to the event log is illustrated in a BPM

context by Ferreira and Thom [24], where semantic reasoning is used to automatically discover patterns from the recorded data.
In the field of activity recognition, time interval restrictions become essential. Cases of dealing with complex events are rapidly increasing. To address this issue, ontologies are used as a basis to preserve information and relationships among events. Thereafter, they are temporally managed by a Complex Event Processor (CEP), yielding to a semantic complex event processing technique as proposed by Taylor and Leidinger [25], where ontologies are used used only for event definition and CEP tool for stream processing.

III. METHODOLOGY

In this section, we will describe our methodology, together with motivation that drives its development. Furthermore, after the description of the architecture of the framework, we will expound the components of the system in details.

A. Motivation

The main goal of AALISABETH project is to detect a set of abnormal behaviours that could bring to the onset of most common diseases. To reach this aim, a technology based on the use of non-invasive sensor network was developed. In particular, recognise human activities by monitoring which home appliance is in use and how long user spends time on appliances is our goal. So, in the private environment a wide network of smart objects is installed. More specifically, a set of sensors has been selected and interconnected through an heterogeneous communicating network that wires the AALISABETH Smart Home (SH). Then, the wide amount of heterogeneous data generated from such network is stored in a data repository, appropriately developed as described in the following sections. They show a fine-grained nature, carrying generally their value, the originating device, the data type, the timestamp and so on. However, the acquired data show a lack of information, both the semantics and the relationships among the inert and living objects that furnish the smart home. Furthermore, one should focus not only on the single values of data, but rather on its meaning within the context. In order to take into account such relationships and formalize the knowledge of the whole context, we propose a methodology based on the employment of a specific domain ontology and it makes use of a CEP engine. In particular, the ontological modeling allows to explicitly specify the key concepts and their properties for a given domain, initially the resulting ontologies are essentially knowledge models. Furthermore, data generated by real domain can be loaded in this model, so an ontology model allows us to merge both static and dynamic information. The granularity features of acquired data are a stumbling block for the contained semantic information, which may be eventually lost. Also, a further verifiable aspect is data redundancy; that is, there can be several devices, which apparently output different results, but they provide the same information. Hence, the ontology is introduced to somehow circumvent such technical aspects and to form a bridge from the real-world system and its formal representation. In fact, it is able to merge the static knowledge and the dynamic parts by means of classes and their instances, rebuilding the whole context. Therefore, the advantages of a semantic technique are exploited twice. Once the ontology-based method has provided a conceptualization and specific

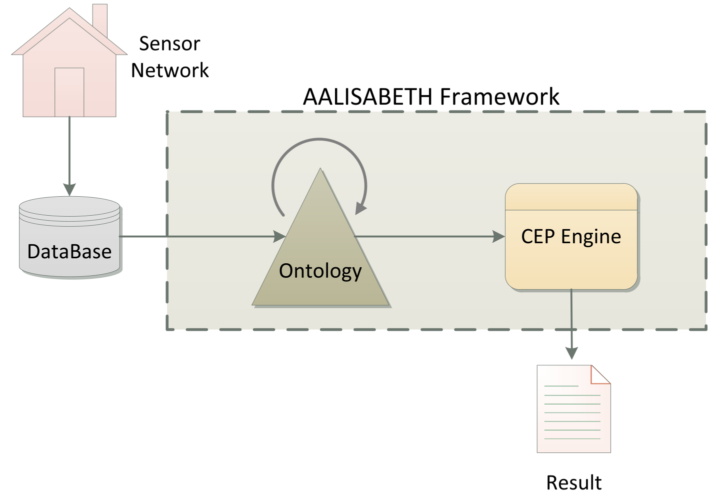

Figure 1. A simplified Architecture Model.

description of the real-world system, such formalization drives the analysis phase. In our specific case, it is needed to look for well-determined set of data. It is worth noting that such research has to be performed according to the own semantics of the desired set. This requirement represents the main reason why an ontology-based technique is introduced.

B. Architecture of the Framework

In order to address the situation previously described, we propose the framework depicted in Figure 1, in which main components of the architecture model are shown. As it is possible to note, the framework is essentially composed of two different interconnected components: Ontology and CEP Engine. More in detail, first of all, data collected in a MySQL relational database management system (RDBMS) are mapped in OntoAALISABETH, a domain ontology for AALISABETH project. In order to create a correspondence from records of DB to individuals of ontology, a d2rq language is used [26]. In particular, it is able to support conditional mappings, mapping of multiple columns to the same property, the handling of highly normalized table structures where instance data is spread over multiple tables, and the usage of translation tables in the mapping process. In this way, a correspondence between each element of DB and the ones of the previously implemented ontology is established. Thanks to the features of the ontology, data can be achieved efficient interoperability of information and they can be reorganized according to their semantic meaning because during the mapping phase it is possible to define how extract data. Moreover, by deploying the rules capabilities, we can divide and combine records. In fact, it is possible to build specific semantic rules in order to organize records.
The next task is represented by the simplification and aggregation of data so the fine-grained nature of data stored in the DB become a list of events that are provided by "virtual sensors". A virtual sensor represents a fictitious sensor, whose data are an appropriate aggregation of sensors data (from one or more different sensor type). For example, in order to detect the "toilet usage" action it is necessary to find a set of atomic actions and a set of locations to identify each particular activity. Furthermore, such events are occurred in specific time. More specifically, it is necessary to getting up, going to bathroom, using the

record_id	record_timestamp	host_id	obj_id	var_id	user_id	timestamp	data	int_value	real_value
199548	2015-02-02 01:28:58	1005	10011	20013	NULL	2015-02-02 01:28:58	2	NULL	NULL
199552	2015-02-02 01:29:07	1005	10011	20013	NULL	2015-02-02 01:29:07	2	NULL	NULL
199553	2015-02-02 01:29:12	1	107	17	2	2015-02-02 01:29:12	NULL	-51	NULL

Figure 2. A partial DB view.

toilet flush. So, the virtual sensor "toilet usage" collects data pertaining to involved sensor for such activity. The generation of these events is critical because, as described later, the pattern recognition is made by a specific event processor. In fact as far as time constraints are not taken into account, an ontology is sufficient to classify and organize data produced from both physical and virtual sensors. Moreover, it is able to achieve efficient interoperability of information systems. However, since our final aim is to obtain a specific time-dependent output, we need to introduce in our framework a component able to manage these time restrictions. This issue is solved by the use of a Complex Event Processing (CEP) engine, that is, a technique concerned with timely detection of compound events within streams of simple events [27]. In wider terms, the scope of this engine is to identify meaningful events.
Now we are entering into a more detailed description over every single component.

C. Components of the AALISABETH Framework

1) Database: The core of the AALISABETH system architecture is represented by the very heterogeneous sensor network, whose data produced are stored into a classical SQL Database, as introduced in Section I. Such database is developed in order to allow the integration between the different elements of embedded network and the supervision system. Moreover, its structure is organized to facilitate the extraction and interaction of contained information. In addiction, due to heterogeneous of devices, it is necessary that the database contains also information about characteristics of each device. So, it is made up a set of table, each of them contains particular information. As described above, data are collected by the sensor network into a classical SQL Database. The most important table is, clearly, the data table that contains all the records. In Figure 2 it is possible to see a small portion of that table. There are various useful columns, the following is a brief description:

- *Record_id*: the unique record id;
- *Record_timestamp*: the time of writing into the DB;
- *Host_id*: the host gateway that writes the record;
- *Obj_id*: the sensor device id;
- *Var_id*: the variable id of the measure;
- *User_id*: the user id acquired by sensors;
- *Timestamp*: the time of acquiring values;
- *Data*: a string column containing sensor measure;
- *Int_value*: an integer value of the measure;
- *Real_value*: a double value of the measure.

In the list of columns, it is possible to note that there are three different type of measure (a string, an integer, a double) this is due to the different nature of sensor devices and the physical value to measure. Also for this reason we have introduced the ontology, in order to filter and standardize data.

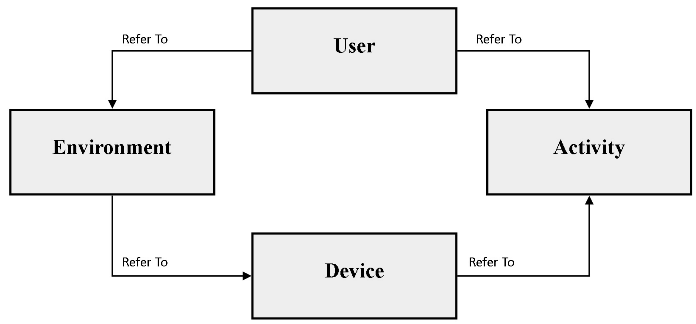

Figure 3. Context ontology overview.

a) Ontology Structure: In our proposed framework, the main element is represented by the ontology that clearly defines the semantics of the considered domain and is used as a shared knowledge base for all related components.
Moreover, it has the ability to reuse knowledge and integrate several knowledge domains. On the other hand, the AAL system is very open and it is able to change. For these reasons, a specific domain ontology, called OntoAALISABETH, has been developed. It shows a particular structure as illustrated in Figure 3, in order to model the whole system. The core is represented by the user that performs activities in own home, monitored by a sensor network embedded in the environment. In accord with this observation, the ontology is composed by four main domain components connected each other. More specifically, User, Environment, Activity and Device are the main parts of AAL system. In fact, User describes the concepts related to user's profile, e.g., age, weight. Another important information can be represented by the medicines that user needs to take, number of children or the presence of a pet. In summary, the knowledge contained in the subdomain is related to user's profile and subsequently it is connected with the performed activities. Each activity is identified by a set of atomic actions and locations. Moreover, it is performed in a time period and it is characterized by a duration. So, in the ontology, it is described in terms of locations, atomic actions, time period and duration.

These two parts of the ontology play the central role. Consequently, the appliances within the AAL environment should adapt to the user, and not vice versa. Then, Environment and Device describe user's house and the sensor network installed. Furthermore, this ontology shows different abstraction layers that composed together form a pyramid-like structure, where each lower level specialises the one on the next upper layer.

The architecture, as reported in Figure 4, is realized by the following main components:

- A static layer (domain and domain-specific ontology);
- A dynamic layer (data and view ontology).

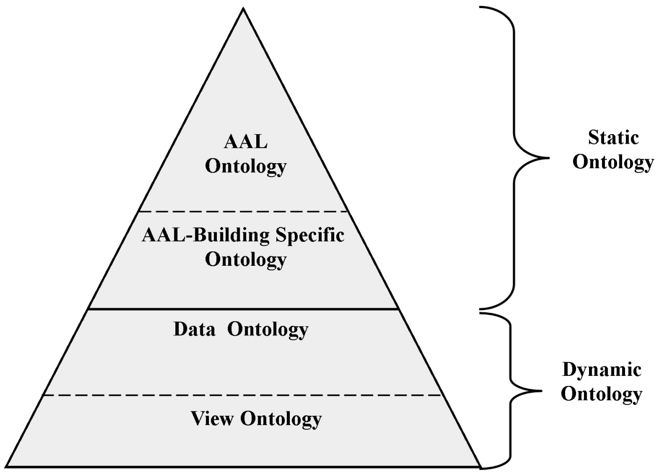

Figure 4. Pyramid-like structure of the ontology.

Each part of our ontology plays a specific role in order to respond to different requirements of the project, as described below.

b) Domain ontology: Initially, an upper domain ontology is built. One should note that this higher level of abstraction can be considered as a ready-to-use ontology for any other analogue domain. In other words, it consists of an ontology, which generally formalizes concepts present in some context, and is thought to be commonly valid. In fact, concepts are described as much generally as possible, carrying static information. Since our instance is an AAL context, as the literature suggests, we implemented a domain ontology extending and reusing an existing one. In our case, the starting ontology to model environment has been chosen to be DogOnt [15]. It has been built in a smart home context, but does not take into account several elements of an AAL environment. Therefore, we have formalized classes and relationships about the SH, its architecture and furniture, the presence and activities of one or more users, the introduction of smart objects with a communication network, sensors and clinical devices, and so on. Specifically, one imported from DogOnt ontology and it is composed of:

- **Building Thing**, it describes all the elements of a Building Environment, divided into Controllable and Uncontrollable elements;
- **Building Environment**, it models rooms and architectural spaces that compose an house;
- **Functionality**, it shapes the ability of a device to be controlled and it defines the possible commands and their range;
- **State**, it classifies continuous or discrete states, according to the kind of values they can assume;
- **Notification**, it models the ability of a device to issue a notification about state/configuration changes and it defines the corresponding notification.

The other category that has been introduced to fully describe an AAL context is defined by the following classes:

- **Activity**, it models all main daily human activities: sleeping, preparing and having a meal, walking, etc;
- **Consumable Thing**, it describes the main categories of foods, drinks and medicines;
- **Environment Profile**, it is divided in two classes: Person and Natural. The class Person depicts the users' main characteristics (as weight, age, build, etc) instead the class Natural is divided in two subclasses: Season and Weather, with the aim to take into account external environmental conditions;
- **Meal**, it introduces the different repast during a day, as breakfast, lunch, etc.

c) Domain-specific ontology: This first middle layer places below the previous upper ontology, extends several static properties and focuses on the structure of the considered domain. In our domain-specific ontology, we formalize the various components belonging to the home environment: the real structure of the ambient and disposition of rooms, the personal information about who lives in the house, which sensors are installed in the network and how they communicate. Also, the complete knowledge of the domain allows the developer to add new elements and relationships in the ontology, which cannot be described in the technology of data storing.

d) Data ontology: The data ontology extends the previous domain-specific layer introducing the concept that each device generates fine-grained data. In this level, the described classes are instantiated with individuals that present a one-to one correspondence with each record stored in the DB. This procedure is obtained via the use of d2rq language. It consists of a mapping that associates data from data sources with concepts in the ontology. Hence, the whole data ontology is implemented taking into account the sensor network, formalized in the previous layer, and is continuously updated. In this step, the semantic information about the fine-grained data is partially recovered, but the following layer permits to have custom specific views of the system.

e) View ontology: In our system, data are generated by the pervasive network, which is installed to monitor user lifestyle. In particular, such records may assume different meanings depending on the specific context. For instance, if a presence in the bedroom is followed by one in the kitchen, it has a different meaning from the same followed by one in the bathroom. Since a particular record deserves different semantic treatments, the view ontology takes into account such various circumstances. More frequently, one must evaluate the presence in the bedroom from different points of view. In terms of an ontology, this necessity converts to the implementation of new view classes where individuals are inferred. So, alternative views provided by this lower layer are needed in order to reorganize instances of data ontology. These views are defined by the expression of several equivalent classes. They are driven by the main scope to classify instances having well-determined properties and relationships; that is, these classes are populated by the desired individuals and carry the same knowledge replicated several times. The whole process of reorganization is allowed by the use of the reasoning tools, which represents the formal basis for the expressive strength of OWL. In fact, through this instrument, it is possible to obtain additional statements that are inferred from the facts and axioms previously asserted. This reviewing step is the grounding of the preprocessing procedure. Thereafter, the reasoning tool allows to perform semantic queries on the ontology and extract the desired information for the following effective analysis, as reported in Figure 1. One should note that querying the ontology in this final step of the proposed methodology corresponds to select an amount of data generated by virtual sensors, i.e., a group of data following the user interpretation of the system. Moreover, this approach developed by means of inference classes has the important advantage to be extensible and additive. In order to better explain the advantages deriving from the classification of the view ontology, let us consider the following cases. One of the most relevant aspects of our project is the capability of monitoring if the user gets up during the night for eating or toileting. In order to recognize these activities, we proceed creating two views, i.e., macro ontology classes. Each class contains all inferred individuals that allow the eventual recognition of the considered activity. In this particular case, the information about getting up and exiting from the bedroom are common. Instead, presence and utilization of the toilet is found in the first case, while presence in the kitchen and opening a sideboard or refrigerator belong to the second view. Furthermore, in both cases we require that the person comes back to the bedroom after some time and continues to sleep. Hence, these sets of individuals populating the view classes are selected as input for the following step of analysis. It is worth noting that processing data with the described technique allows to preserve relationships and constraints introduced by the previous domain-specific layers of the ontology. Contents of each layer of the pyramid-like structure are shown in Figure 5.

D. Framework Implementation

Let explain more deeply the framework by using Figure 6, which clearly shows the dataflow of the system. The starting point is represented by a data table full of useful but also noisy data. At the real beginning, there is a simple extraction of data, every instance of the ontology represents every row of the database. The second phase is represented by filtering and selecting data in which we model data entries in a different way. In fact, the concept of "entry" is substituted by the concept of "event". By the deployment of OWL Java library, data entries are aggregated and picked out in order to generate more complex events, called "macro-event". During this step, macro-events are further selected and filtered. This could be done thanks to the deployment of ontology rules and reasoner. In fact, as described before, the importance of the ontology is its ability in preprocessing task and semantic filtering. It is possible to define rules to select, clear or divide by semantic meaning. The next task is performed by the interaction with the user, which could not be an IT expert. The user can choose two important parameters: which pattern he wants to analyze and the query string. So the system sends to the CEP engine, Esper, the events as Java POJO objects. Then Esper according to the input query string analyses the stream of events looking for patterns that match with the query.

1) Reasoning & Rules: In order to reorganize and preprocessing data, a set of rules are introduced into ontology. In particular, such rules allow to group data depending on time interval. In order to reach the aim, a set of built-in rules, an alternative paradigm for knowledge modeling, are introduced to acquire new knowledge by establishing new object property connections between unrelated entities. These rules, which are able to extend the expressivity of OWL, are evaluated periodically during runtime and new facts are added into the ontology. The built-in rules can be easily extended by defining custom rules. The definition of equivalent classes is driven by the main scope to classify instances carrying determined properties and relationships; that is, these classes are populated by the desired individuals. Instead, the preprocessing phase is based on the ability of the reasoning tool to query the ontology and extract the required information for the following effective analysis. One should note that querying the ontology in this final step of the proposed methodology corresponds to select an amount of data generated by virtual sensors, i.e., a group of data following the user interpretation of the system.

2) Complex Event Process (CEP) Analysis: The ontology structure is extremely powerful but it has serious expressiveness limitations: the lack of support of temporal reasoning. Considering the nature of our analysis and the dynamic updating of dataset, traditional methods do not allow to perform reasoning over time and space, so we introduce a CEP engine in order to perform the temporal analysis procedure. This engine permits to combine data from multiple sources to infer events or patterns that suggest more complicated circumstances. In fact, the main objective is to recognize significant events. These identifications could be eventually reused to discover further more complex events, through additional uses of CEP engine.
In our framework, we deploy Esper [28], a CEP engine library, which is able to process large volumes of incoming messages or events, regardless of whether incoming messages are historical or real-time in nature. It is successfully used into finance field (e.g., trading, risk management).

3) Java Component: The whole framework is developed using Java Language. We chose Java because using that it is possible to read and manage OWL Ontology thanks to Jena API. Furthermore, we are able to dynamically create Java classes at runtime. In fact, the initial idea is to develop a "context-free" framework, in other words the framework has not to be related to structure of the ontology, sensor type and home environment. Moreover, Esper does not inherently support a specific access to OWL ontology, so it is necessary to map the events of the ontology into simple Plain Old Java Objects (POJOs). For this reason, we implemented a component that by the deployment of Javassist library [29], is able to create Java classes corresponding to ontology classes. These Java classes are also called "POJO classes". In that way, the system results extremely dynamic regarding the variables of the environment.

4) Graphical User Interface (GUI): In Figure 8 is shown a first Graphical User Interface (GUI). It is composed of different buttons and text boxes. The execution is divided essentially in three phases:

1) by clicking the "Load Data" button, records contained into the DB are exported into the Ontology file thanks to mapping procedure;
2) on the right the user can choose which type of pattern wants to analyze;
3) then the user can define a query string on his own and wait for results after clicking the "Execute query" button.

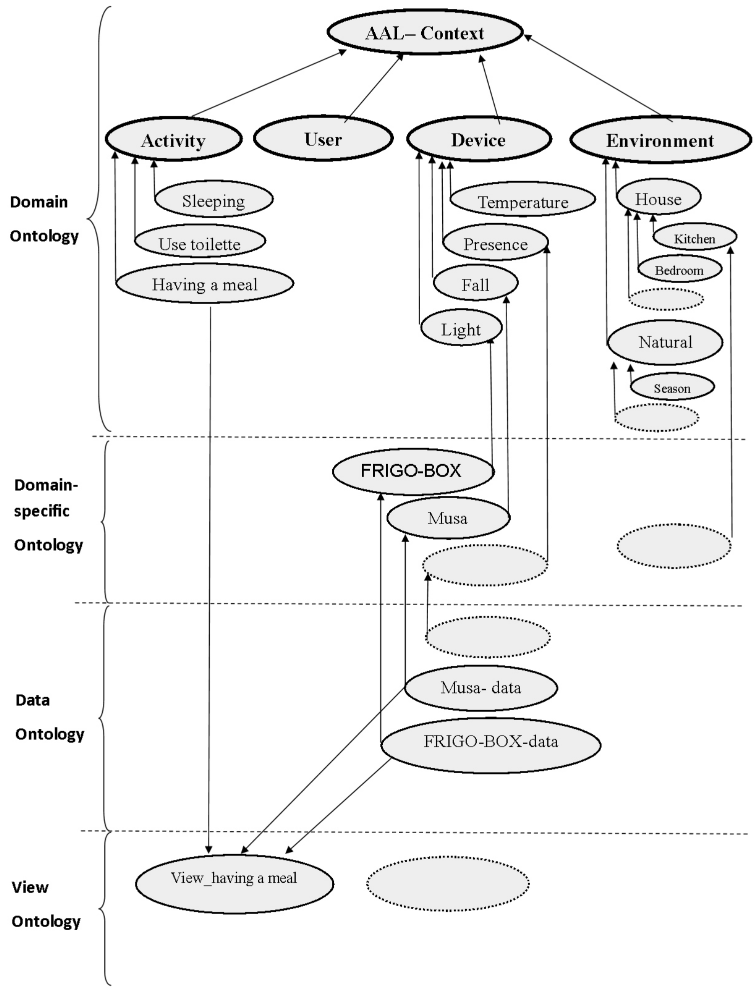

Figure 5. Class hierarchy diagram of OntoAALISABETH.

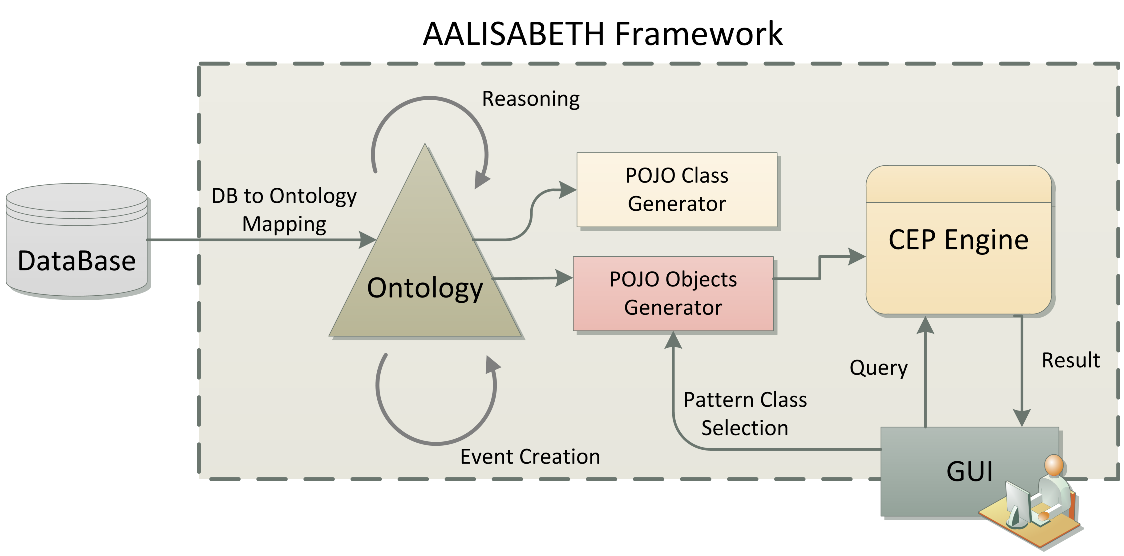

Figure 6. A detailed Architecture Model.

This is a simple GUI that allows us to deploy the main features of the framework.

5) Tools & Technologies: We are now trying to summarize tools and technologies that allowed us to develop this framework. First of all, the OWL ontology is developed and tested in Protégé 4.3 [30], together with the Pellet Reasoner Plug-in [31], which permits the creation and population of equivalent classes. Through the definition of a mapping configuration we deployed a OBDA system, in order to write down the statements that map the Database to the ontology. To implement the framework, we use Java as a coding language to combine several techniques. Then, the ontology is managed by means of the OWL API. Thereafter, the Pellet reasoner is invoked through Jena [32] to perform reasoning over the ontology together with the individuals. The SPARQL query is also executed through Jena.

Basically, using Jena we load the ontology file created with Protégé into an ontology model (a Java object implementing the OntModel interface). We then choose to utilize Esper as CEP tool for several reasons: its open-source Java library for complex event processing, it can be used in different data streams and CEP applications, it has adapters that allow the user to provide different input formats for the representation of events.

The whole Java framework is developed using Eclipse IDE [33].

E. Test & Validation

The test configuration is composed of a standard flat in which an elderly person lives. The inhabitant has a regular behaviour although he knows that a sensor network was installed and where sensors are positioned. This sensor network captures a great number of actions performed by the user. Data generated are initially store into a classical DB, then are imported into our framework.

In this section, we are going to explain, using a simple example case, capabilities and features of our approach. First, we have to describe the characteristic of the test configuration more in detail.

1) Use Case Description: Our test house is a ground floor flat, where a man lives. In Figure 7, it is possible to see the plan of the flat. It is composed of a living room / kitchen, a bedroom, a bathroom and an office.

The sensors installed are (the number in the list correspond to the number on Figure 7):

1) *Presence Couch* sensor: a sensor that measures the weight of the couch;
2) *Passage Office* sensor: observes the passage of a person coming in/out office;
3) *Passage Living (Hall)* sensor: as for sensor 2, observes the passage of a person coming in/out living room;
4) *Opening Fridge* sensor: observes when the fridge is opened;
5) *Opening Pantry* sensor: observes when the pantry is opened;
6) *Scale*: observes and captures what the user is weighing;
7) *WC flush* sensor: captures when the flushing device is used.

In this set of sensors, probably the most interesting is the Scale sensor. In fact by the use of particular plates provided by Radio-Frequency IDentification (RFID) tag, the sensor observes the weight and the type of dish. That allow us to calculate the calories and the amount of carbohydrates.

This sensor network will be our heterogeneous data source in the provided example.

2) Pattern Recognition Example: Considering the described above type of sensor available we define this query string:

```
select * from  pattern [ every (A =
    DataElementEvent(event_type=1)
    -> (B = DataElementEvent(event_type=4)
    and not DataElementEvent(event_type=1))
    -> (C = DataElementEvent(event_type=3)
    and not DataElementEvent(event_type=4))
    -> D = DataElementEvent(event_type=7
    and timestamp_start <
    A.timestamp_start + 3600000))]
```

This pattern represents a sequence of actions, starting from the bathroom, then the passage into the living room, then opening the fridge and finally being present on the couch. All of these actions must be performed within 10 minutes ($A.timestamp\_start < A.timestamp\_start + 3600000$, timestamp values are processed as milliseconds long type). In order to fully understand Esper querying syntax we remand to references [28].

In Figure 8, we want to illustrate the output of the described Esper query. Once the event set and user have been selected, POJO objects representing events are created and passed as input to Esper engine. Now, we define our query pattern and execute it. On the bottom, are displayed the results that are all patterns that match the query.

In our example, the framework was able to find two matching patterns for the input dataset, represented by five days of monitoring. As it is possible to note in Figure 8, patterns found are related to the dates "*2015-02-04*" and "*2015-02-05*". The example exposes the main potentialities of the framework, starting from management and selection of a huge amount of data, and passing through the definition of an ad-hoc query string for every type of recognizable human behaviour. In fact, this kind of example must be considered as an initial point for a more complex and elaborated use case according to a precise and meaningful social analysis.

IV. Conclusion

In this paper, firstly we have presented an ontology-based framework to retrieve semantic information from a data repository. Later, we have illustrated a case of use in which our methodology is applied and the obtained results.

Our work represents a different approach to the pattern (or activity) recognition problem. The novelty of our architecture is represented by the combination of two concepts: semantic reasoning with temporal analysis. Due to the ability of ontology to reuse knowledge, it represents the central element of the presented methodology. The developed ontology, named OntoAALISABETH, is characterized by four layers: a top-level ontology followed by a domain-specific one, and data layer which establishes over a final basis-view layer. The top-ontology has a particular structure, in particular it is composed of four domain ontology system - User, Environment, Activity, Device - that represent the whole knowledge base in AAL domain. The last part is thought as a data preprocessing step. It plays the role to organize data according to the desired context

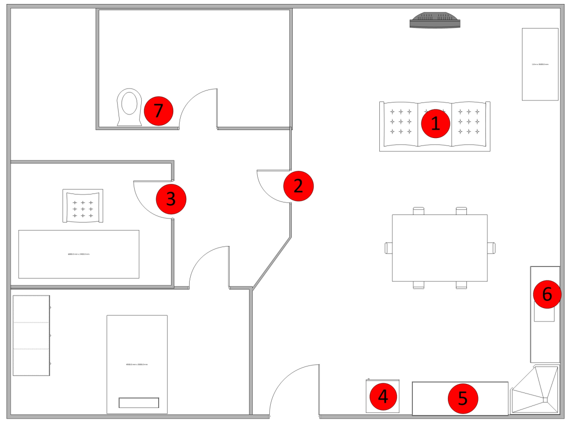

Figure 7. A plan of the flat.

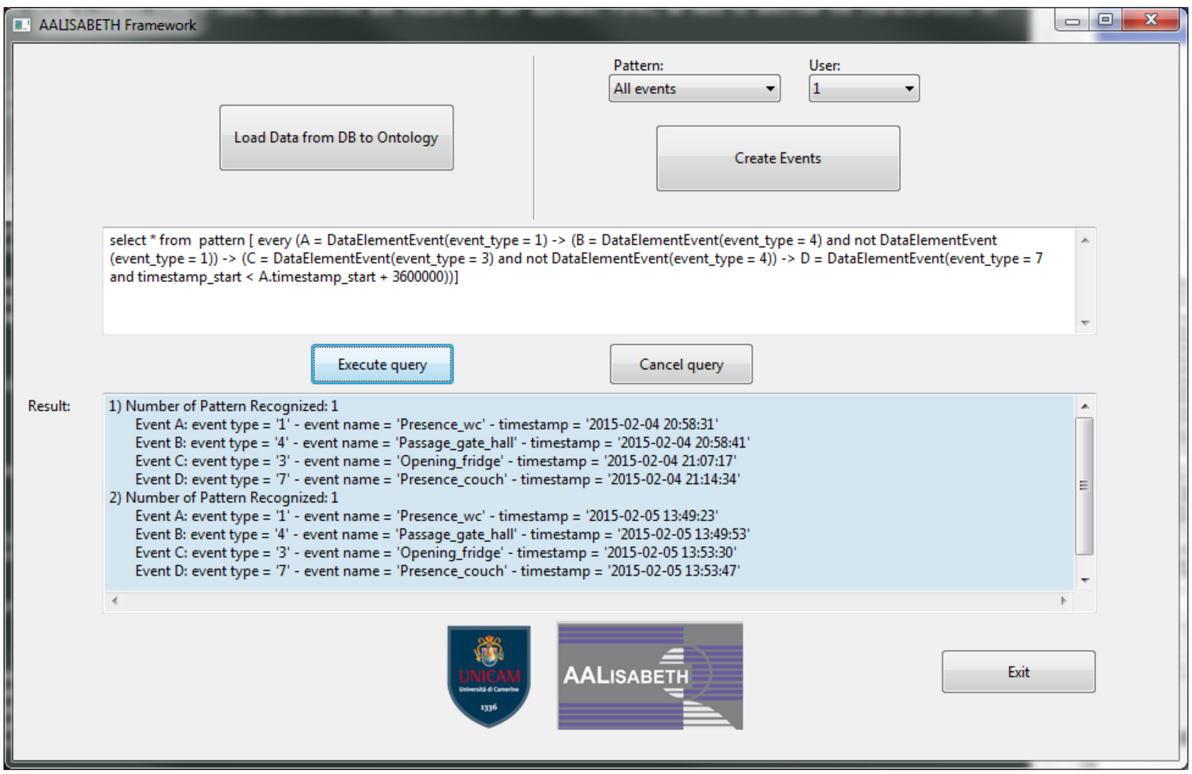

Figure 8. The AALISABETH GUI with patterns recognized.

views, in order to allow a proper analysis. Then, due to the expressiveness limitation of ontology, the Cep engine has been introduced. In fact, the CEP engine works as an analysis tool to support the timed reasoning.

In general, knowing the lifestyle of human in home is not an easy task, since each person has the different way to perform the activity. Some activity events are occurred in specific time for each day. Hence, time can be used to distinguish the activity in specific detail. For example, we can know the "Eating & drinking" activity, which is for breakfast, lunch, or dinner based on time. In addition, it is very important for further analysis, i.e., health-care system needs to know the time when patient has a meal in each day. Thus, if we know the lifestyle of human in home environment, we can predict, which activity will occur in specific time. For instance, human take a bath twice a day, after wake up and before goes to sleep. The system can predict the "take a bath" activity, if user wake up in the morning and go to the bathroom.

Unfortunately, we can not provide any comparison in term of performance with different kinds of approaches. However, the model described is very extensible and it is not bound by sensor types or environment variables.

Regarding possible future developments, it could be very interesting if the pattern found is stored into the ontology as a new macro-event with a label. In this way, it will be possible to look for particular pattern into this macro-event set. In this manner, we will create another level of abstraction with a more complex and detailed behavioural analysis. Moreover,

the main disadvantage is probably represented by the Esper query language that could be tough for an non-IT expert, for this reason, it could be extremely helpful to develop a custom pseudo-code notation, closer to human language, which will be interpreted by the framework.
A good improvement could be represented by the implementation of a real-time analysis, in other words, every time the database receives a new record, it alerts the framework that extracts the information and manages it in order to detect a matching pattern at real-time. This kind of approach will be useful also in emergency situations in which the caregiver has to rescue the user (e.g., a detected fall).
In conclusion, we have developed our work in an AAL context, but thanks to extensible nature of the framework it is reasonable to think that our approach can be applied also on smart cities, a city characterized by modern urban production factors in a common framework with the intent to improve the quality of life and a sustainable economic development.

Acknowledgment

The authors would like to thank every partner of AALISABETH project for the great working collaboration done until now. They also acknowledge the financial contribution of the Marche Region administration, under the action Smart Home for Active and Healthy Aging, for supporting the research on the AALISABETH project.

References

[1] R. Culmone, M. Falcioni, and M. Quadrini, "An ontology-based framework for semantic data preprocessing aimed at human activity recognition," in The Eighth International Conference on Advances in Semantic Processing, SEMAPRO 2014. IARIA, 2014, pp. 1–6.

[2] The ambient assisted living (aal) joint programme. Last checked: 2015-05-28. [Online]. Available: http://www.aal-europe.eu/

[3] F. Castiglione, V. Diaz, A. Gaggioli, P. Liò, C. Mazzà, E. Merelli, C. G. Meskers, F. Pappalardo, and R. von Ammon, "Physio-environmental sensing and live modeling," Interactive Journal of Medical Research, vol. 2, no. 1, 2013.

[4] F. Corradini, E. Merelli, D. R. Cacciagrano, R. Culmone, L. Tesei, and L. Vito, "Activage: proactive and self-adaptive social sensor network for ageing people," ERCIM News, vol. 2011, no. 87, 2011.

[5] E. Frontoni, E. Gambi, L. Palma, L. Pernini, P. Pierleoni, D. Potena, L. Raffaeli, S. Spinsante, P. Zingaretti, D. Cacciagrano, F. Corradini, R. Culmone, F. D. Angelis, E. Merelli, B. Re, L. Rossi, A. Belli, A. D. Santis, and C. Diamantini, "Interoperability issues among smart home technological frameworks," 2014.

[6] Aalisabeth - ambient-aware lifestyle tutor, aiming at a better health. Last checked: 2015-05-28. [Online]. Available: http://www.aalisabeth.it/

[7] P. Chahuara, A. Fleury, F. Portet, and M. Vacher, "Using markov logic network for on-line activity recognition from non-visual home automation sensors." in AmI, ser. Lecture Notes in Computer Science. Springer, pp. 177–192.

[8] A. Sharma, Y.-D. Lee, and W.-Y. Chung, "High accuracy human activity monitoring using neural network," International Conference on Convergence Information Technology, 2008.

[9] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, Jan. 2000, pp. 4–37. [Online]. Available: http://dx.doi.org/10.1109/34.824819

[10] T. van Kasteren and B. Krose, "Bayesian activity recognition in residence for elders," in Intelligent Environments, 2007. IE 07. 3rd IET International Conference. IET, 2007.

[11] D. J. Patterson, D. Fox, H. A. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage." in ISWC. IEEE Computer Society, pp. 44–51.

[12] L. Getoor and B. Taskar, Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press, 2007.

[13] T. Gruber. What is an ontology? [Online]. Available: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html (2009)

[14] M. Gagnon, "Ontology-based integration of data sources," in Information Fusion, 2007 10th International Conference on.

[15] D. Bonino, E. Castellina, and F. Corno, "The dog gateway: enabling ontology-based intelligent domotic environments." IEEE Trans. Consumer Electronics, no. 4, pp. 1656–1664.

[16] J. Mocholí, P. Sala, C. Fernández-Llatas, and J. Naranjo, "Ontology for modeling interaction in ambient assisted living environments," in XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010. Springer, 2010, pp. 655–658.

[17] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An ontology-based context model in intelligent environments," in Proceedings of communication networks and distributed systems modeling and simulation conference, vol. 2004, 2004, pp. 270–275.

[18] S. M. Sedighi and R. Javidan, "Semantic query in a relational database using a local ontology construction," South African Journal of Science, vol. 108, no. 11-12, 2012, pp. 97–107.

[19] J. Barrasa Rodríguez, Ó. Corcho, and A. Gómez-Pérez, "R2o, an extensible and semantically based database-to-ontology mapping language," in In Proceedings of the 2nd Workshop on Semantic Web and Databases. Springer-Verlag, 2004.

[20] OWL 2 Web Ontology Language Document Overview, W3C Recommendation, Std., 10 2009. [Online]. Available: http://www.w3.org/TR/owl2-overview/

[21] N. Cullot, R. Ghawi, and K. Yétongnon, "Db2owl : A tool for automatic database-to-ontology mapping." in SEBD, 2007, pp. 491–494. [Online]. Available: http://dblp.uni-trier.de/db/conf/sebd/sebd2007.html#CullotGY07

[22] M. Rodriguez-muro, L. Lubyte, and D. Calvanese, "Realizing ontology based data access: A plug-in for protégé," in In Proc. of the Workshop on Information Integration Methods, Architectures, and Systems (IIMAS 2008. IEEE Computer Society Press, 2008, pp. 286–289.

[23] T. Tran Thi Kim and H. Werthner, "An ontology based framework for enriching event log data," in SEMAPRO 2011, The Fifth International Conference on Advances in Semantic Processing, 2011, pp. 110–115.

[24] D. R. Ferreira and L. H. Thom, "A semantic approach to the discovery of workflow activity patterns in event logs," International Journal of Business Process Integration and Management, vol. 6, no. 1, 2012, pp. 4–17.

[25] K. Taylor and L. Leidinger, "Ontology-driven complex event processing in heterogeneous sensor networks," in Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part II, ser. ESWC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 285–299. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017936.2017959

[26] The d2rq platform. Last checked: 2015-05-28. [Online]. Available: d2rq.org/

[27] D. Anicic, P. Fodor, S. Rudolph, and N. Stojanovic, "Ep-sparql: a unified language for event processing and stream reasoning," in Proceedings of the 20th international conference on World wide web. ACM, 2011, pp. 635–644.

[28] Esper 5.2 documentation. Last checked: 2015-05-28. [Online]. Available: http://www.espertech.com/esper/documentation.php

[29] S. Chiba, "Javassist - a reflection-based programming wizard for java," in International Business Machines Corp, 1998.

[30] Protégé - a free, open-source ontology editor. Last checked: 2015-05-28. [Online]. Available: http://protege.stanford.edu/

[31] Pellet reasoner plug-in for protégé 4. Last checked: 2015-05-28. [Online]. Available: http://clarkparsia.com/pellet/protege/

[32] Apache jena - a free and open source java framework for building semantic web. Last checked: 2015-05-28. [Online]. Available: http://jena.sourceforge.net/

[33] Eclipse ide. Last checked: 2015-05-28. [Online]. Available: https://www.eclipse.org/