

Semantic Web Standards

José Luis Medina Burgos
SNET
Computer Engineering
Technische Universität Berlin
Berlin
Email: josmedbur@gmail.com

Abstract—Semantic Web is set to become the future because it makes the understanding between humans and machines easy. In this paper Semantic Web Standards, such as RDF and OWL, the current semantic developing languages, are going to be examined. Before talking about those two languages, Semantic Web will be introduced, along with its related vocabularies, which enable the construction of the Semantic Web, and how the Linked Data is developed and deployed has to be explained. Deeper into the paper Inference property of Semantic technologies is going to be introduced as well as a framework where ontologies can be created and also reasoned. Everything in this paper has been joined with some examples so that a better understanding is possible. Finally, what has been done so far is explained through a semantic-technology-based application as well as what the future outlooks of this technology are within Smart Home application.

I. INTRODUCTION

The traditional and more often outdated "Web of documents", rests on the idea of fulfilling the users requests by returning representations: such representations are no more than documents that are identified by a URI. As matter of fact, this system has been successful so far, and is still being nowadays, but these still remain the necessity of a better understanding between the machine and the person. As the "web of documents" does not take into account semantics, the concept of "Semantic Web" (Web of data) has been introduced as an addition to the "web of documents" and is gaining more importance day by day, since this idea is based on supplying semantics to the World Wide Web allowing this good understanding and more intuitive and an easier use of this medium.

Several technologies have been introduced for the implementation of Semantic Web that make possible the Web of data by linking data, building vocabularies, creating data and making the web capable of storing data, and inventing rules which allow to handle this data. There are many well known standardized technologies which realize the linked data process: RDF (Resource Definition Framework)[1]; OWL (Web Ontology Language)[2]; SKOS (Simple Knowledge Organization System)[3]; and RIF (Rule Interchange Format)[4]. Of course not all of them are useful for the same applications: each of them will require different specifications since, for instance, they have different complexity levels. In this paper only RDF and OWL are going to be introduced as currently they are the most used.

OWL will be considered as the standard semantic technol-

ogy for building ontologies. An ontology (in the semantic web context) is a classification which consists of a set of conceptualizations of knowledge from a certain domain and the relationships between them. A piece of software called Protégé[5] is going to be shown in this text since it is a good framework that performs construction of OWL ontologies.

The next step in the paper after Protégé is Inference[6]. This property (used by semantic technologies such as RDF and OWL) consists of automatically creating new relationships within an ontology which are derived from existing facts or information in the ontology.

So far, an introduction has been given to what is going to be shown through this paper but the query process must not be forgotten, because it is a very important part what the semantic web is made for. A query[7] (comes from the Latin word "quaere" which means to ask) is a set of directives that the user enters in a search engine or a database in order to find certain piece of information. The greatest exponent of the query languages is SPARQL[8] (SPARQL Protocol and RDF Query Language).

In this paper it is impossible to write in depth about the many applications the Semantic Web has so only two of them are going to be shown in the final sections. The first is called PIPS (Personalized Information Platform for Health and Life Services)[44] about a health and knowledge service developed by the European Commission in order to supply Healthcare employees and European citizens. The second one is about adapting Smart Homes to the Semantic Web so that Elderly and handicapped people will be able to live day by day without other Human help (the Smart Home supplies all the services that the person needs).

II. SEMANTIC WEB

The Semantic Web[6] which is considered as an extension of the web of documents (current web) claims to become the future technology. But what is the Semantic Web? As it has been briefly mentioned in the introduction, the Semantic Web provides a better understanding and cooperation between people and machines so that a more intuitive and dynamic web is possible. All the information in the Semantic Web is well defined in a machine-readable format which enables upload data that can be understood by machines and people.

Since nowadays a huge amount of documents exist in the web, it is difficult to find what the user really wants. Thanks to the

meaning given to the data on the web by the Semantic web, it is possible to find the right data. This good classification of data provides a good base for developing ontologies which help create the semantic web.

Regarding what has already been done with ontologies we notice how many different scenarios it can be applied to as well as how helpful and useful those applications may be. Many examples (which have been extracted from Protégé repository) are going to be shown in order to give some interesting information so that the knowledge about this topic can be increased. Protégé provides enough wide and varied ontology implementations to overview the matter we are talking about. For example there are biology-oriented ontologies such as: BioPAX[9] which is used to share data between biological pathways, BIRNLex[10] acronyms for Biomedical Informatics Research Network; health-care-oriented ontologies such as: Dietas[11] as an ontology that acts as if it were a dietician, BreastCancerOntology[12] used for describing characteristics of this illness; to technology-oriented ontologies such as: AIM@SHAPE[13] ontologies for Semantic Web development; economy ontologies such as: FEA-RMO[14] Federal Enterprise Architecture in OWL, Monetary[15] ontology; etc.

III. LINKED DATA

As an addition to the classic document-based web, the Linked Data[16][17] concept adds the idea of supplying structured data to the web, which means that data must be linked so that "the web of data" can be explored by people or machines thus making the web more useful. In brief, Linked Data consists of connecting data which is in the web and is not related to one another. In order to link and structure this data, RDF (Resource Description Framework) and HTTP (Hypertext Transfer Protocol) protocol are used. Hence the new linked-data-web is not cutting off from the traditional document-based web, so document-based web has to be considered and, therefore it is necessary to have some knowledge about it. As we know, the "web of documents" is constituted by some Standard Web Technologies, such as: URIs which are used as standard identifiers; those URIs[19] are a set of characters used to identify resources on the internet e.g. URIs[20] and are used to identify hypertext documents called HTML[21], in which the creation of the web content is delegated; and the HTTP[22] as a protocol to transfer hypertext. So the point is that the "web of data" uses these URIs as resources identification, the HTTP for retrieving resources of its descriptions; and RDF (the chosen standard web technology for linked data, which is a XML-based metadata data model) for describing resources. These characteristics make the process of linking data reliable, and consequently the possibility of making the data queryable. Tim Berners-Lee coined four principles of Linked Data (Design Issues: Linked Data)[18]:

- 1) Use URIs to identify things.
- 2) Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents.

- 3) Provide useful information about the thing when its URI is dereferenced, using standard formats such as RDF/XML.
- 4) Include links to other, related URIs in the exposed data to improve discovery of other related information on the web.

This theory might sound pretty much consistent and accurate, thus this may make the concept of linked data difficult to understand. Now it is the time where the applications to real life situations and examples come into action, so that the concept of Linked data finally will be cleared up. In figure 1 there is an example: Ana is a student in Freie Universität that studies Informatik. She likes watching football every Sunday because her friend Paco plays in the Humboldt FC. Paco studies History. He could decide between Humboldt Universität and Freie Universität but decided Humboldt Universität. Actually he is attending some courses at Freie Universität which has an agreement with Humboldt Universität in order to improve its History degree.

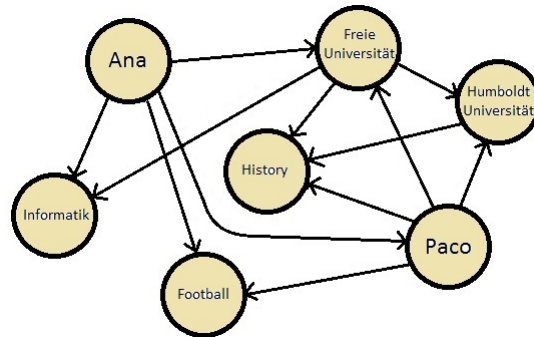


Figure 1. Set of linked entities representing two students

According to the big amount of Linked Data applications, we can classify them in four groups which are clearly demarcated: applications that reuse data from the Linked Open Data (LOD) cloud so that less time is wasted in the process; applications which rate and tag resources unambiguously; applications which consist of answering users questions; and applications that manage event-based data to make it queryable and organized so that people can set or look for events. In addition to this previous information, it is mandatory to say that an application may belong to more than one group. Despite that, this quadruple division allows a better understanding.

In order to clear up those four groups I would like to introduce to you some real examples which are being currently used. One of them is Music Beta[23] a website which is an application deployed by BBC which reuses content obtained from the LOD cloud. It is an application based on Musicbrainz[24] which supplies metadata from Wikipedia (if this metadata is there) via DBpedia interlinking.

Another application, this time a Question-Answering-oriented system, is the Semantic CrunchBase Twitter Bot[26] which is an Auto reply bot from which someone can get information by sending a message to this bot. This application uses the CrunchBase API[25], which is a free directory of technology

companies, investors and people.

Finally, one of the most important and useful examples where the linked data concept is used is DBpedia[27], which is an RDF model database that consists of a Linked Data space whose inner structure comprises HTML-based data browser pages and SPARQL endpoints, and in which the information is obtained from Wikipedia and then structured to make this information free to be accessed on the World Wide Web.

IV. VOCABULARIES

So far an introduction to what Semantic Web is and how data is connected in the core of this concept called the Web of data has been made. However the explanation of this topic cannot continue without looking at semantic web vocabularies in depth, thus they are the tools and the bases of the semantic web development process.

To be more specific, vocabularies allow us to describe certain domains, as well as their resources and the relationships between them. Moreover, another main utility of vocabularies is to classify those resources in order to provide, for instance: a variable level of complexity (from one up to thousands of concepts); constrains for a certain domain; or establish relationships between terms.

Hopefully the concept of vocabularies in this certain context is already explained, but: do we know what the vocabularies are used for? Well, as it has already been experienced, plenty of ambiguous ideas, concepts, etc may exist in this world. In this try to avoid ambiguity, vocabularies help data integration. To fulfill those tasks in a successful way, many languages have been developed and standardized, but in this paper two XML-based techniques released by W3C are going to be studied: RDF (Resource Description Framework) and OWL (Web Ontology Language).

A. RDF

The well known Resource Description Framework (more usually named after RDF)[28] is a XML-based language used for describing resources in a certain domain and creating relationships between them basically used for representing information on the web. FOAF is maybe one of the best examples which uses this RDF technology and will provide a good, understandable way to make the explanation more clear. FOAF[29] is the acronym of Friend of a Friend which is an ontology built for describing people, the activities they do and create, and the relationships between them. FOAF is a machine-readable ontology which can be easily written, manipulated and processed, thus it is RDF-based ontology. The atomic unit used in RDF is the so called triple. Each one of these triples consists of the combination of three parts: the subject, the object, and the predicate. For instance, we can create a triple in FOAF like: Lidia knows Jessica (Figure 2).



Figure 2. RDF triple

Such subjects and objects are known as the nodes of an RDF graph[30] which is a set of triples. Each of these nodes is a graphical representation of a URI reference, a literal or a blank; and the relationships that link them called properties-are URI references as well (Figure 3).

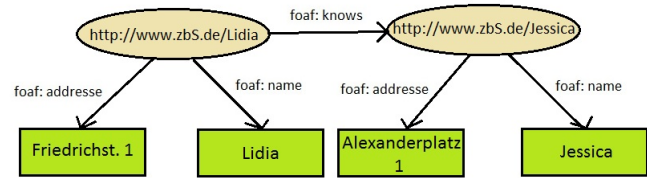


Figure 3. RDF graph

In the next example we can see what a fragment of RDF code (based on figure 5) looks like:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:Person rdf:about="http://www.zbS.de/Lidia">
    <foaf:adresse>Friedrichst. 1</foaf:adresse>
    <foaf:name>Lidia</foaf:name>
    <foaf:knows rdf:about="http://www.zbS.de/Jessica"/>
  </foaf:Person>
  <foaf:Person rdf:about="http://www.zbS.de/Jessica">
    <foaf:adresse>Alexanderplatz 1</foaf:adresse>
    <foaf:name>Jessica</foaf:name>
  </foaf:Person>
</rdf:RDF>
```

As it can be figured out in the picture two persons are being described (Lidia and Jessica) whose names and addresses have been depicted, and there is a relationship between them: Lidia knows Jessica.

Besides the fact that RDF is a simple data model, another important goal of this technology is the inference property which is supported by the formal semantics that RDF contains which allows an unambiguous workspace. In brief, the concept of inference means new relationships can be established automatically thanks to the information obtained from existing facts in the ontology.

Another point to consider concerning RDF is the extensible vocabulary that it uses. Such vocabulary has URIs as the greatest exponent which are used to name all the resources in the set of data. Not to forget that, as it is said at the beginning of this chapter RDF is a XML-based language which means that RDF is able to exchange information among other XML-based applications. Those specifications make this system ready for future growth (extensibility)[31]. Hence, this performance makes it possible to represent any resource and its state, although the result may not be compatible with the real world, thus for example wrong constrains might be established. This problem has to be managed by the programmers and designers.

B. OWL

The second language that is going to be mentioned is the Web Ontology Language (OWL)[32] which is a markup language for creating ontologies. OWL is XML-based language -since OWL is an extension of RDF- and allows us to publish or share ontologies on the web. It may sound somewhat unclear, and the point is that the concept of Ontology must be explained in order to keep on explaining OWL. Ontology is a word that comes from Greek and is used by philosophers to study different aspects of human beings and their relationships. According to this definition, ontologies in Semantic Web are conceptualization of knowledge from a certain field and the relationships between the entities this ontology is made up of. For instance OWL is very useful for creating academic, industry, medical, etc ontologies in order to ease the access to data referred to those areas of knowledge.

Furthermore, OWL supplies users with three sublanguages which fulfill different requirements depending on what the designer wants. For simple classifications and constrains we can use OWL Lite thanks to its simplicity for example it is easy to translate from taxonomy to thesaurus. Taxonomy and thesaurus are two kind of ontologies which are really extended. Moreover, when the maximum expressiveness is required OWL DL (Description Logics) may be used. It insures the computation of all conclusions in a finite period of time. Finally OWL full can be used by anyone who needs its main characteristic: freedom of using the whole OWL language spectrum, but on the other hand computational completeness is not insured.

As in RDF, an example will be used to go through the explanation of this chapter: it is a model that consists of kind of Pizzas[33] and theirs toppings and as the result of adding these topics some subclasses as Vegetarian Pizza, Margherita Pizza, and so on can be created. As an example some fragments of OWL XML/RDF syntax code concerning to the Pizza ontology have been pasted:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">

  <owl:Ontology>

    <owl:Class rdf:about="#SNETPizza">
      <rdfs:label>Spanish</rdfs:label>
      <rdfs:subClassOf rdf:resource="#NamedPizza"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#hasTopping"/>
          <owl:someValuesFrom rdf:resource="#CPU"/>
        </owl:Restriction>
      </rdfs:subClassOf>

    <owl:ObjectProperty rdf:about="#isToppingOf">
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
        FunctionalProperty"/>
      <rdfs:comment>Any given instance of topping should only
        be add ed to a single pizza (no cheap half-measures on our
        pizzas)</rdfs:comment>
      <rdfs:range rdf:resource="#Pizza"/>
      <rdfs:domain rdf:resource="#PizzaTopping"/>
      <rdfs:subPropertyOf rdf:resource="#isIngredientOf"/>
```

```
</owl:ObjectProperty>
...
</owl:Ontology>
</rdf:RDF>
```

First thing to take into consideration from this fragment is the Header of the code which contains the "prefixes". Each of those prefixes represents different sets of attributes which can be used later in the ontology. The aim of doing this is to provide short forms (for instance "rdf") instead of the whole direction (<http://www.w3.org/1999/02/22-rdf-syntax-ns>). After the header it comes the collection of assertions which form the ontology, inscribed between the owl:Ontology tag element. Inside the "body" of this ontology there are different structures that can be differentiated such as: classes, subclasses, properties or restrictions. For instance we can see how the class SNETPizza has been created and CPU is one among all the topics that SNETPizza may have. Continuing with the code it is easy to notice that the class SNETPizza is a subclass of NamePizza which means SNETPizza is one kind of pizza which contains certain ingredients such as CPU, OS, etc. In order to delimit the ingredients SNETPizza may contain restrictions. In those restrictions there are two indicators: the one that sets cardinality restrictions; and the one that defines the kind of value allowed to be used. In OWL two kind of properties are available: the object properties (owl:ObjectProperty); and the data type property (owl:DatatypeProperty). In the example can be seen how the property isToppingOf is created: first of all it is defined that it is a functional property; then it is necessary to set the range of values permitted and the domain in which the property will act. Finally isToppingOf is a subproperty of isIngredientOf.

This is just a small example of the power of OWL language which contains many more attributes such as special properties (inverse, equivalent, symmetric, transitive, etc), cardinality restrictions and Boolean combinations (complementOf, unionOf, intersectionOf).

V. FRAMEWORK FOR CREATING ONTOLOGIES

As a well known java-based open source ontology editor, Protégé[34] has been chosen. Such a good framework, in which knowledge-based applications can be created as well, provides the possibility to create, manipulate and visualize ontologies in several formats: RDF or RDFS, OWL and XML Schema are the available formats.

When concerning ontologies modeling, there are essentially two ways to do it: the Protégé-Frames editor; and the Protégé-OWL editor. The first one follows the concept of knowledge representation systems (KRSs)[35] which helps us create ontologies by building conceptualization of concepts and organizing them in a hierarchy; creating a set of relationships between concepts and creating instances from this concepts. Hence this modeling way is very useful, for instance for the Biological Processes Ontology which classified every term related to Biology such as cellular location, illnesses, composition of elements, documentations related to biological stuff, and so on. Thanks to the connection between elements on the

ontology, relations between components can be discovered by querying the system. In addition to what has been explained so far there are some important characteristics of Protégé-Frames that must be highlighted: it includes a plug-in architecture which can contain some elements such as graphics, sounds, different storage formats as well as additional support tools. Besides Protégé-Frames has a java-based API that makes it possible for a great compatibility.

The second way for ontologies modeling is the Protégé-OWL editor and lies on the Web Ontology Language (OWL) described in a previous chapter. As we should know so far OWL[36] is one of the current standard for Semantic Web creation capable of creating a whole ontology: conceptualization of classes; properties between them and instances created from these classes. Once the ontology is built, the property of inference comes out: the formal semantics that OWL contains determines the logical consequences not presented in the ontology. Moreover this editor can support OWL and RDF ontologies as well as the edition and visualization of its classes, properties and rules. Equally important is the possibility of integration and execution of reasoners which are pieces of software that enable the inference property to spread its power across the ontology. The concept of inference and reasoner will be explained in depth in the next chapter.

In order to explain how ontologies can be created in Protégé many modifications in the famous Pizza ontology have been done as an example. A little schema of this ontology example has been developed with some classes already seen (Figure 4):

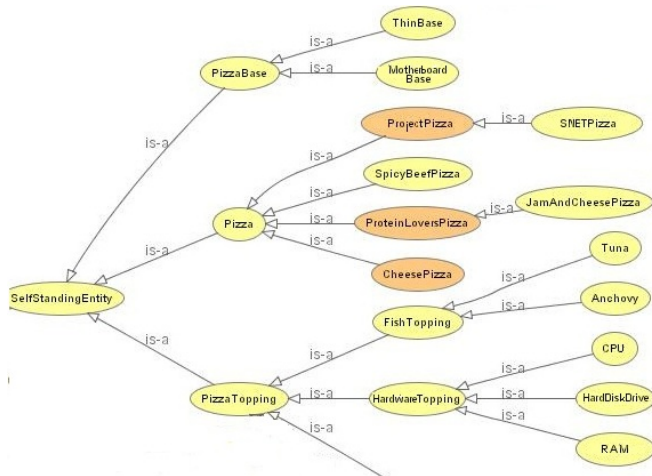


Figure 4. Mindmap for SNETPizza.owl

Specifically a new class called SNETPizza and all the necessary information to build it up such as subclasses have been implemented in this framework. First of all several classes, which are subclasses of PizzaTopping and PizzaBase, have been created: HardwareTopping, SoftwareTopping, OSTopping and their associated subclasses such as: CPU, HardDiskDrive, RAM and OpticalDiskDrive belong to HardwareTopping; CounterStrike, Eclipse and Protg belong to SoftwareTopping; and different versions of Android belong to

OSTopping. Other subclasses are MotherboardBase as a subclass of PizzaBase; ProjectPizza which is a subclass of pizza; and finally SNETPizza which contains all the features for describing a SNET pizza. As it can be seen, groups of classes can be created what it means that, for instance: HardwareTopping, SoftwareTopping, OSTopping are subclasses of PizzaTopping but they are disjoint between them. Other important entities of the ontology are properties. In this case hasTopping and hasBase have been used: related to hasTopping it is defined that a SNETPizza always will have one among OSs, one CPU, one RAM and one HardDiskDrive; related to hasBase it is defined that it always will have a MotherboardBase. The two properties have been defined with a certain domain (which is Pizza in hasTopping and hasBase) and a range (which is PizzaBase for hasBase and PizzaTopping for hasTopping). Another task to take into consideration is setting restrictions: restrictions are important in order not to for example let a SNETPizza have more than one OS.

VI. INFERENCE

The concept of inference has been lightly introduced in the previous chapters but it has to be explained in more detail in order to totally understand what it is exactly and which important frameworks are being currently used. As it is said, inference[37] means new relationships can be established automatically thanks to the information obtained from existing facts in the ontology such as existing relationships, rules, etc. Inference is a really important principle which has made Semantic web possible to exist. For instance, whether a data set has a triple that says "Lidia isIn Berlin" and another one says "Berlin isIn Germany" then the system may add the triple "Lidia isIn Germany". But why is inference that important?

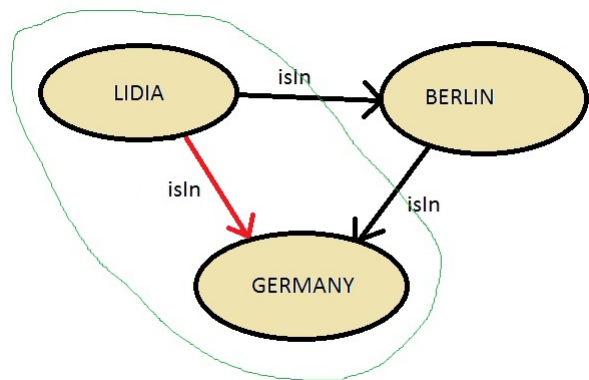


Figure 4. The new triple is sorounded by the green line

It is known that information can be introduced in the semantic web by using the regular vocabularies such as OWL, RDF(S), and so on. Somehow it is possible to use the inference power by means of "rules". Those rules are just functions which return a certain result or conclusion after some given premises. To be more specific, rules are divided in two main groups: Derivation Rules and Reaction Rules[38]. In the first group fits the rules that can derive new information from existing

information. That is the reason why they are also named Deduction Rules.

The second group of rules is called Reaction Rules. Those kind of rules are established when a certain event takes place. Such an important group of rules has many important members such as: Condition-action rules (Production Rules), whose action appears when some condition is fulfilled; Event-Condition-Action (ECA) which consists of event part (what is the element that trigger the rule), condition part (the system logical test is done in order to figure out if the event was the required one), and the action part (execute the action).

In addition to this interesting topic it is not less important to introduce a real scenario where the inference idea can be used: a reasoner. As we already know, since it was lightly introduced in a previous chapter, a reasoner is a piece of software capable of deducting or deriving new information from the existing information in the ontology. Semantic Reasoners[39] are generalizations of Inference engines[40] (piece of software that derives results from existing knowledge) since reasoners provide techniques in order to enhance the power of themselves. For instance, some reasoners use first-order predicate logic[41] (which is a set of rules use to derive and a formal language) as a core so that providing reasoning. The one mentioned here is going to be the so called Pellet[42] - which can be inserted as a plug-in in Protégé-. It is a very spread software since it is an open source written in Java and its license is free as well as being able to perform its task satisfactorily. Pellet is the right choice for OWL DL ontologies and supports reasoning with nominal (enumerated classes of object value restrictions); conjunctive query answering (belongs to first order queries); as well as ontology debugging. The procedure is easy: the ontology consistency reasoner receives an input and then gives back the result as consistent, inconsistent or unknown. More important features -besides which have already been mentioned- are: supports ontology analysis and repair; Pellet supports entailment (the key is inference whereas for Description logic the keys are satisfiability and subsumption divide complex behavior into many simple behavior parts-); new datatypes can be defined and be tested with this reasoner.

VII. QUERY LANGUAGES: SPARQL

A query language is a set of primitives and commands that can perform directives in order to retrieve information from the Semantic Web. As we know so far, this information in the shape of data may be inscribed on documents, databases or RDF files. Since the topic is semantic web we are going to talk about RDF. RDF enables us to insert information on the web and link it but: what has to be done in order to obtain this information? SPARQL[43] (SPARQL Protocol and RDF Query Language) is a well known query language intended to query RDF systems and return a result from it. The structure of this query language consists of triples (called basic graph patterns), as RDF does, but there is a difference between them: each of the part that belongs to the triple (subject, predicate, and object) can be a variable. When some information is wanted to be searched a query has to be written. This query will look for subgraphs in the data that are equivalent to the query subgraph by checking whether the

RDF terms from the subgraph of the RDF data may fit in the variables.

As an example for this theme, the one used in RDF chapter can be used , referring to FOAF:

Data:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>
_:a foaf:name "Lidia Garcia Munoz".
_:a foaf:address "Friedichstr. 1".
_:b foaf:name "Jessica Brown".
_:b foaf:address "Alexanderplatz 1".
```

Query:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?address
WHERE
{
  ?x foaf:name ?name.
  ?x foaf:address ?address }
}
```

Query result:

Name	Address
"Lidia García Muñoz"	"Friedichstr. 1"
"Jessica Brown"	"Alexanderplatz 1"

VIII. APPLICATION EXAMPLES

A. PIPS

The first of them is called PIPS[44] (Personalized Information Platform for Health and Life Services) which is a health and knowledge service developed by the European Commission in order to supply Healthcare employees with an outstanding and up to date source of knowledge and European citizens with healthy lifestyles in the shape of food industry, healthcare suppliers and so on. PIPS manages to gather citizens, researchers, health related policy makers, public organizations as well as healthy suppliers, food and drug industry and services creating a continuously updating knowledge setting so that quality of life in Europe would be improved.

In brief PIPS has been created by joining some technologies in order to allow users enter an interactive world to get the right and custom information, just by using their computer, laptops, mobile devices, etc. Such technologies, that PIPS requires, as software agents, reasoners, knowledge management or natural language generation need a certain environment where they can work properly and so that PIPS can perform their tasks. This environment needs ontological approaches in order to make it possible.

Lets imagine that one ill man is carried to the hospital. Doctor Joe sees him and determines several symptoms that can help him to figure out which illness he has to fight. Doctor Joe is an excellent professional but in order to be certain of what he is about to diagnose he uses PIPS. He introduces the symptoms: chills, fever, sore throat, muscle pains, severe headache, coughing, weakness/fatigue and general discomfort. Then PIPS retrieves the result: the patient has influenza (commonly known as flu). After the hard work performed by doctor Joe to diagnose flu and the help of PIPS he can finally make up his mind and tell to the patient which kind of medicaments he has to take. PIPS can automatically retrieve

the medicaments to be taken as well.

One of those ontologies PIPS uses is a Food Ontology which provides users with plenty of information about different kinds of food. This information consists of the quantity of food and liquid a user must take a day; nutritional characteristics about them as well as what the ingredients are of some elaborated products. Unfortunately there are not that many food ontologies, but several food ontologies have been, and are still being, developed in order to ease food matters. For example, What am I eating[45] is a ontology-based online dictionary which provides food information. Also AIMS[46] (Agricultural Information Management Standards) which belongs to Food and Agriculture Organization of the United Nations has developed AGROVOC, an ontology that contains information about agriculture, fishing and forestry in 20 different languages. Finally, LanguaL the International Framework for Food Description[47], is a thesaurus-based system capable of capturing and describing data about food as well as retrieving it.

In order to build the ontology first of all the environment in which the ontology is going to rule has to be described so that a certain implementation framework will be delimited. Once the ontology has been roughly designed it is not unwise trying to find existing ontologies that may help the process. How important are the terms that has to be taken into consideration in order to define classes and how are those classes classified? For this last point Eurocode2 coding system provides a good base for class hierarchy formation. So the next steps are creating the properties of classes and slots (which have to be defined) and at the end instances will be able to be created. Concerning the development process of the ontology, Protégé has been used since it is an editor where ontologies can be developed graphically. Hence this is an intuitive and a simple way to edit ontologies which, once the ontology has been created, can be translated into OWL. Despite the fact that Protégé can translate into OWL, it is necessary to check the OWL code created in order to make the ontology consistent. For that purpose reasoners are used: Pellet can be used as a plug-in in Protégé.

B. Semantic Data Management for Smart Homes

By way of further information about the topic as well as showing you in which direction Semantic web is going to grow, the example of Smart Homes will be introduced in this paper[48]. Smart Homes (SH) are houses which have been automatized so that the house itself can perform the housework and the daily routine activities. Elderly and people with some illness such as mentally or physically handicapped can or will be able to choose this Smart Home alternative in order not to depend on human care. It sounds like this is a good solution but despite of all that has been done, more research is necessary so that Smart Homes will be able to react when the handicapped person needs something, for example when the disabled person wants to go from one room to another the system must be aware in order to open the door at the certain time.

Here is where the semantic technologies will take part since e.g. the system must not be only able to water the plants every certain interval of time, or control the central heating; but the system has to observe and recognize what is happening in the place and what their characteristics and behavior are; reason and interpret the data obtained in the previous observation step; and act according to the data gathered by the sensors as well as predict future actions. This data is obtained in a machine readable format which enables the total compatibility with semantic technologies and successful management of this data by the software agent which operates in this particular environment. Moreover using ontologies in this environment allows us to create and represent relationships between data as well as automate comprehension of data by the system since the data is stored in machine readable well defined data.

According to the development of this application, in this paper the solution proposed -when concerning data modeling- is to create seven ontologies in order to define the Smart Home environment. Those ontologies are: one ontology for actions and daily routine actions; another for different areas at home; an ontology for humans; another ontology for health-care; one ontology for software; and finally one for time. For the creation of these ontologies Protégé OWL editor is going to be used again since it provides the creation of ontologies as well as knowledge bases.

The developers of this system suggest dividing the creation of semantic part of Smart Homes into two phases: the first one is about describing all the detection equipment (e.g. sensors) by manually semantic implementation (e.g. in Protégé) as this equipment is limited in a Smart Home and will not take so long. The second phase is no less important: the information collected from those sensors must be translated to terms so that the semantic system or humans can directly understand what the sensor is capturing in real time. For instance, instead of receiving "0" or "1" when someone opens the door it receives "close" or "open".

When concerning data storage, the semantic system guarantees all the data collected to be saved in knowledge bases as RDF triples. The implementation recommended by the authors of the paper in referencing consists in dividing it into two related components and place them in a centralized repository. The first of those components will contain the semantic descriptions for the entities playing a role in the Smart Home such as the humans, sensors, and so on. This information is considered as static data in the environment since it only needs to be stored once. The second component is the one that deals with the so called dynamic data which is the information collected from the daily routine actions that depends on time also i.e. always need to be stored when the action takes place.

So far, how to collect data and how to store it have been roughly explained on these previous chapters. From now on how to give a use to this data will be explained so that a good perspective of this semantic-based application will be brought to mind. As far we know a lot of machine readable data have been stored and in order to use it and reason it the figure of Assistive Agents take part in this environment. The mission of

an Assistive Agent is to interpret the obtained data which is already machine readable data- and retrieve an action in real time to assist daily routine actions.

IX. CONCLUSION

In this paper we have talked about what the Semantic Web is and what the pieces that form it are. For clarity, many examples have been included throughout the text such as the RDF application FOAF or the Pizza ontology developed in OWL. Going from why the data must be linked (Linked Data) to what the concept of inference is and why so important is and why several characteristics that Semantic Web needs in order to exist, and to provide such amazing results, are so important..

Also what has been done so far has been reported in this text through PIPS application. Thanks to this ontology-based application we can see what the advantages of semantic technologies are and what they can be useful for. Finally Semantic technologies in Smart Homes has been introduced. This is a future outlook for using semantic technologies which will be very useful as it will be able to help people day to day. It is these application examples, and a thousand more that already exist, or that are being developed, that give us hope to have a better future.

REFERENCES

- [1] *Resource Description Framework (RDF)*. <http://www.w3.org/RDF/>.
- [2] *Web Ontology Language (OWL)*. <http://www.w3.org/2004/OWL/>.
- [3] *SKOS Core Guide*. <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/>.
- [4] *RIF Overview*. <http://www.w3.org/TR/rif-overview/>.
- [5] *Protégé*. <http://protege.stanford.edu/>.
- [6] Sure, Y., 2003 *Knowledge Technology Fact Sheet - Semantic Web*. Institute AIFB, University of Karlsruhe.
- [7] *Query*. <http://searchsqlserver.techtarget.com/definition/query>.
- [8] *SPARQL Query Language for RDF*. <http://www.w3.org/TR/rdf-sparql-query/>.
- [9] *BioPAX*. <http://www.biopax.org/>.
- [10] *BIRN Lex*. <http://www.bioontology.org/BIRN Lex-NIF>.
- [11] *Dietas Ontology*. <http://serdis.dis.ulpgc.es/a013715/aic/ontologias/>.
- [12] *Breast Cancer Ontology*. <http://acl.icnet.uk/mw>.
- [13] *AIM Ontology*. <http://www.aimatshape.net/resources/aas-ontologies>.
- [14] *Federal Enterprise Architecture Reference Model Ontology*. <http://www.osera.gov/owl/2004/11/fea/FEA.owl>.
- [15] *Monetary Ontology*. http://protegewiki.stanford.edu/images/d/de/Monetary_ontology_0.1d.zip.
- [16] Bizer, C., Heath, T., Berners-Lee, T., 2009 *Linked Data - The Story So Far*. International Journal on Semantic Web and Information Systems.
- [17] Zeng, M., Needleman, M., Oh, Sam., Phipps, J., Summers, E., DeRidder, J., Hodge, G., 2010 *Linked Data – Enabling Standards and Other Approaches*. ASIST 2010, October 2227, 2010, Pittsburgh, PA, USA.
- [18] Berners-Lee, T., 2006 *Linked Data*. <http://www.w3.org/DesignIssues/LinkedData.html>.
- [19] *URI Wikipedia*. <http://en.wikipedia.org/wiki/Http>.
- [20] *What do HTTP URIs Identify?*. <http://www.w3.org/DesignIssues/HTTP-URI.html>.
- [21] *HTML Wikipedia*. <http://en.wikipedia.org/wiki/Http>.
- [22] *HTTP Wikipedia*. <http://en.wikipedia.org/wiki/Http>.
- [23] *Music Beta*. <http://www.bbc.co.uk/music/beta>.
- [24] *Musicbrainz*. <http://musicbrainz.org>.
- [25] *Crunchbase*. <http://www.crunchbase.com>.
- [26] Hausenblas, M., 2009 *Linked Data Applications*.
- [27] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S., 2009 *DBpedia - A Crystallization Point for the Web of Data*. Freie Universitt Berlin, Universitt Leipzig, Digital Enterprise Research Institute.
- [28] Klyne, G., Carroll, J., McBride, B., 2004 *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C.
- [29] Brickley, D., Miller, L., 2010, *FOAF Vocabulary Specification 0.98*. Marco Polo Edition.
- [30] *RDF Graph and Syntax*. <http://www.obitko.com/tutorials/ontologies-semantic-web/rdf-graph-and-syntax.html>.
- [31] *Extensibility*. <http://en.wikipedia.org/wiki/Extensibility>.
- [32] Smith, M., Welty, C., McGuinness, D., 2004, *OWL Web Ontology Language*. W3C.
- [33] Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C., 2004, *OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns*. Department of Computer Science, University of Manchester and Stanford Medical Informatics, Stanford University.
- [34] Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C., 2004, *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0*. University of Manchester.
- [35] *Open Knowledge Base Connectivity Home Page*. <http://www.ai.sri.com/okbc/>.
- [36] *OWL WEB ONTOLOGY LANGUAGE CURRENT STATUS*. http://www.w3.org/standards/techs/owl#w3c_all.
- [37] Kääriäinen, P., 2009, *Inferences in the Web*. Helsinki University of Technology.
- [38] *Condition Rules*. http://en.wikipedia.org/wiki/Reactive_planning#Condition-action_rules_.28productions.29.
- [39] *Semantic Reasoner*. http://en.wikipedia.org/wiki/Semantic_reasoner.
- [40] *Inference Engine*. http://en.wikipedia.org/wiki/Inference_engine.
- [41] *First-order Logic*. http://en.wikipedia.org/wiki/First-order_predicate_logic.
- [42] *Pellet: OWL 2 Reasoner for Java*. <http://clarkparsia.com/pellet>.
- [43] *SPARQL*. <http://en.wikipedia.org/wiki/Sparql>.
- [44] Cantais, J., Dominguez, D., Gigante, V., Laera, L., Tamma, V., *An example of food ontology for diabetes control*. Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK ITACA, Universidad Politecnica de Valencia, 46520 Valencia, Spain Istituto Scientifico Universitario San Raffaele, Via Olgettina, 60, 20132 Milan, Italy.
- [45] *What am i eating?*. <http://www.whatamieating.com/>.
- [46] *AGROVOC - Agricultural Information Management Standards*. <http://aims.fao.org/pages/592/sub>.
- [47] *LanguaL*. <http://www.langual.org/>.
- [48] Chen, L., Nugent, C., Al-Bashrawi, A., 2009, *Semantic Data Management for Situation-aware Assistance in Ambient Assisted Living*. School of Computing and Mathematics, Shore Road, Newtownabbey Co. Antrim, NI, UK, BT37 0QB 0044 28 90368837.