DATA POINT MAPPING APPROACH TO AIRPORT ONTOLOGY MODELLING AND POPULATION

Nikola Tomašević, Marko Batić, Vuk Mijović, Sanja Vraneš
University of Belgrade, Institute Mihajlo Pupin

Abstract – *For development of the intelligent airport energy management system, a comprehensive airport data model is required to describe all static knowledge relevant for the airport energy management, enabling the integration and interoperability of different technical systems. One way of providing such comprehensive airport data model is based on the ontology modelling paradigm. Having in mind that airports are rather complex infrastructures with numerous, heterogeneous devices coming from different vendors, and therefore characterized with the large quantities of the static data, it is important to choose a suitable approach for ontology modelling. This paper presents one of the possible approaches to the airport ontology modelling and population which combines three different, but yet complementary methods: LODRefine tool, SPIN mapping and SPARQL Update queries. The flexibility of the proposed approach was seen in possibility to instantiate any airport infrastructure of the given complexity. The input for population of the airport ontology model was extracted from the various data sources such as data point lists, technical sheets, audits, interviews, questionnaire etc. Finally, as a test-bed platform for the ontology population, two specific airport infrastructures were chosen, Malpensa airport in Milan and Fiumicino airport in Rome.*

1. INTRODUCTION

Present-day airport energy management systems (EMS), are leveraged upon the legacy supervisory control and data acquisition (SCADA) systems and building management systems (BMS) which are faced with the problem of complex heterogeneous infrastructure comprising high number of the field devices produced by different vendors and using different protocols. For providing more advanced airport EMS, classification and description of the target airport facility within a comprehensive airport data model is a prerequisite. This implies description of the belonging devices, systems, their technical characteristics and relations, communication protocols, semantics of the low-level data etc.

One way of providing such comprehensive airport data model is based on the ontology modelling paradigm which was proposed in [1]. Ontology is one of Semantic Web pillars and can be defined as a formal way of representing the knowledge as a set of concepts and their relations in domain of interest. In other words, the ontology is a formal, explicit specification of a shared conceptualization [3]. Therefore, it can be utilized to provide the description of the domain of interest by defining the related entities and their interdependencies, but also to reason upon the modeled entities. The advantage of the ontology concept reflects in reducing the field level heterogeneity and in easier adoption of the future technical equipment. So far, a number of ontology-based facility management frameworks were proposed in the literature such as [4], [5] supporting the integration of multi-vendor devices/sensors. Furthermore, the ontologies were used to increase the energy efficiency and to provide adequate energy saving strategies of so-called smart homes such as in [6], [7]. However, none of them deals with the airport energy management domain and provides a holistic approach to the airport facility modelling.

On the other hand, an ontology-based Airport Data Model (ADAM) layer serving as a common metadata layer of the airport EMS was proposed in [1] and [2]. ADAM layer served as a central data repository which provided the needed semantics to the involved EMS components. In that way, the integration and interoperability of the overall system was supported. The airport ontology developed as part of the ADAM layer, provided a semantic enrichment of low-level signals in order to provide a high-level messages to the airport operator or airport manager. First, a generic model was developed, i.e. the core airport ontology, which had to be extended and instantiated further to model a specific airport infrastructure.

This paper provides a detailed explanation of the ontology modelling procedure and presents one of the possible approaches which could be undertaken for ontology population task. At the start of the development procedure, it was necessary to identify a suitable modelling approach and to choose general concepts behind the modelling. This influenced the decision about the ontology model characteristics such as granularity, abstraction and classification of related entities within the ontology class hierarchy.

As previously mentioned, the airport ontology served as a central data repository of the overall airport EMS solution, and therefore it had to be populated with the static data about the airport facility and target systems/equipment (such as significant energy consumers). First, it was needed to acquire the relevant semantics and to populate it within the airport ontology model. This implied the definition of new concepts, as well as instances and their properties. As proposed in [1], the core airport ontology was first defined providing a generic model of the airport facility. To model a specific target airport infrastructure, the core airport ontology had to be extended and populated based on the static data gathered from the field. For both modelling and populating the airport ontology, two specific airport infrastructures were used as a test-bed platform, Malpensa (MXP) airport in Milan and Fiumicino (FCO) airport in Rome due to their rather complex infrastructure and variety of the field-level devices installed at the site.

Input for the ontology population was extracted from the raw data points (i.e. low-level signals) monitored by the BMS which were identified as relevant from the energy management perspective. This data were used first to extend the core airport ontology model and then to populate it. In this way, it was possible to instantiate any airport infrastructure of interest which is the main advantage of the proposed approach. This paper also elaborates in detail possible data point mapping approaches which were used for the airport ontology population. This task was mainly performed in fully automated manner. Three different, but yet complementary approaches have been taken into account for translation of the data point lists into the ontology: LODRefine tool [8], SPIN mapping [9] and SPARQL Update queries [10]. To provide an overview of their functionalities and capabilities, each of the mentioned approaches were described in detail.

The reminder of this paper starts with Section 2 in which the main aspects and objectives of the airport ontology in context of airport EMS are described. Section 3 elaborates in detail the overall airport ontology modelling procedure and some modeling issues which influenced the structure of the airport ontology hierarchy. Section 4 presents possible data point mapping approaches which were utilized to perform the population of the ontology model. Finally, Section 5 concludes the paper.

2. AIRPORT ONTOLOGY SPECIFIC ASPECTS

Before detailed elaboration regarding the airport ontology development and population approach, a brief overview of the general context in which the airport ontology is used as a meta-model and platform supporting the integration and interoperability between different subsystems is given. As previously mentioned, a need for a common metadata layer resulted in the development of a unique Airport Data Model (ADAM) (shown in Figure 1) for which an ontology-based approach was adopted [1], [2]. The aim of the ADAM layer was to serve as a common knowledge base repository used by the EMS components and to contribute to the integration and interoperability of these components. It provides a semantic enrichment of various signals coming either from the legacy BMS, SCADA, applied fault detection and diagnosis (FDD) algorithms or directly from the sensors/data-loggers installed within the airport facilities, thus enabling the high-level information for the end-user (airport operator or energy manager) as described in [1].

The main objectives for the development of ADAM, tailored to suit the needs of the comprehensive EMS solution, would be the following:

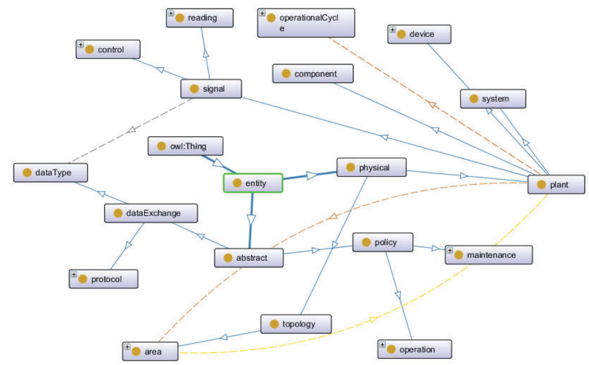

Figure 1. Excerpt of airport ontology class relations and dependencies.

- **Modelling the domain of interest**, i.e. defining the infrastructure of the concrete airport building by classifying installed systems relevant to the energy consumption aspect with belonging sensor/actuator devices;
- Providing means for **technical characterization and semantic interpretation** of signals going to/from the installed equipment (which incoming/outgoing signal belongs to which device/system, what its characteristics are, relations to other devices/signals);
- Providing the **topological profile of the airport facility** and information about geographical location of every installed device/signal (useful for analyzing spatial correlation of data).

Initially, it was important to identify the modelling approach which will be undertaken and the general concepts behind the modelling. The decision regarding the granularity, abstraction and classification of different entities at different levels of the ontology hierarchy is highly influenced by these issues. Since it was meant to be used as a central data repository of the airport EMS, the airport ontology had to be populated in order to store all the static data regarding the targeted systems (information related to the Significant Energy Users (SEU), e.g. air handling units (AHU), such as nominal mass flow, fan drive power etc.) which will be later used to calculate the energy waste due to faults and corresponding saving potential. In other words, for development of the ontology-based ADAM layer, first it was necessary to acquire all relevant information about the airport infrastructure and then to transfer it into the airport ontology which affected both the domain model (ontology hierarchy) and creation of new instances of the ontology entities. For transferring the gathered information into the airport ontology, i.e. for the task of the ontology population, it was important to select yet a suitable approach which will be elaborated in detail in Section 4.

3. AIRPORT ONTOLOGY MODELLING

At the beginning of the airport ontology development process (as shown in Figure 2), the core airport ontology was defined providing a generic model of the airport facility [1]. Core airport ontology was comprised of the common concepts identified as relevant from the perspective of the energy management usually present in the airport infrastructures. It provided modelling guidelines to describe the technical characteristics and topological profile (precise location) of the systems and devices installed at the site.

As a test-bed platform for the ontology modelling, two specific airport infrastructures were chosen, Malpensa (MXP) airport in Milan and Fiumicino (FCO) airport in Rome. Serving as a two major European air-traffic hubs, MXP and FCO airports were taken as suitable also owing to their rather complex infrastructure, technical characteristics and different aspects of the existing devices and modules. Based on the data gathered for the technical characterization of the ICT systems and the analysis of the different aspects of the involved devices and equipment, the core airport ontology had to be extended and further instantiated to model specific target airport facility. In other words, by extending and populating the core airport ontology these two separate ontology instances emerged representing two full-blown airport ontology models tailored according to the current state of the target pilots.

Development of the airport ontology (Figure 2) consisted of the definition of the main ontology entities with the corresponding properties and relations, as well as the harmonization with the contemporary ontology modelling regulations and standards. For this purpose, the leading ontology modelling standards (SUMO ontology [11], IFC (Industry Foundation Classes) data model [12] and CIM as part of IEC 61970 series of standards [13]) were taken into account. For ontology modeling, OWL as one of the most utilized modeling languages was used [14]. Protégé^{TM} tool [15] was selected as ontology development framework used not only for modelling but also for ontology population tasks.

An excerpt from Protégé user interface depicting the water pump entity as part of the extended airport ontology model is presented in Figure 3. It could be seen from Figure 3 that each instance of the class "waterPump" is modelled with belonging list of properties such as "device_id" representing the unique identifier (ID) of the device, "partOf_system" indicating the system to which this specific device belongs, "belonging_signal" representing the list of belonging "signal" instances etc. Furthermore, some of the nominal technical characteristics of device are represented by properties such as "outputPower_kW", "hydraulicPressure_kPa" and "waterFlow_l/s". The topological perspective of the corresponding device instance is modelled with the properties representing the area/position where that specific device is installed (properties "locatedAt_area" and "position").

The input for population of the airport ontology model was extracted from the various data sources such as BMS

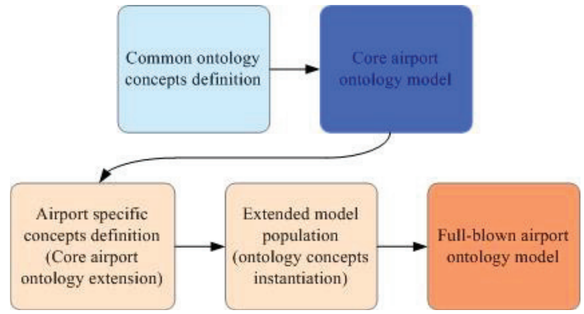

Figure 2. Airport ontology development process.

data point lists, technical sheets, equipment manuals audits, interviews, questionnaire etc. These sources contained the information that enabled classification of energy management related entities (such as HVAC system, power supply, water supply system etc.) and definition of the corresponding relations between them. The extracted information had to be transferred into the airport ontology by instantiating the corresponding instances of the ontology entities with belonging property values and relations. Once the airport ontology was populated, semantic queries were used for extracting the knowledge from the airport ontology.

Both extension and population of the core airport ontology had to be performed based on the semantics of the raw data coming from the field devices, i.e. low-level signals. The data semantics were extracted from the BMS data point lists carrying the information about every low-level signal that the EMS might encounter. Since at both target airports, the semantics of the low-level data/signals were aligned with the Unified data point naming (UDPN) convention [16], harmonization of the airport ontology instances was carried out correspondingly. The harmonization considered mapping of the data point semantics, such as signal identifier, signal source, data type, signal characteristics etc. into the airport ontology as corresponding entity instance or property value. Thus, the flexibility of such ontology modelling and population approach could be seen in possibility to instantiate any airport infrastructure starting from the core model. The following section will elaborate in detail possible data point mapping approaches which were used to populate the airport ontology.

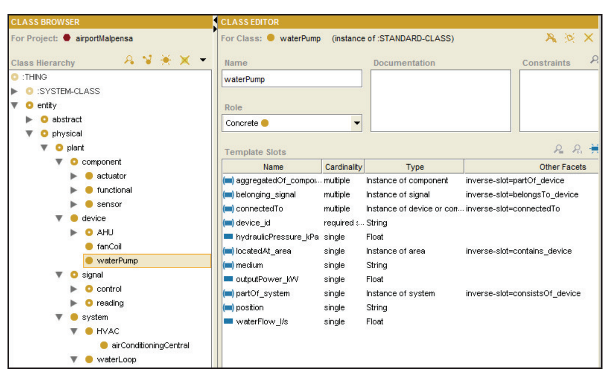

Figure 3. Device entity (water pump) properties.

4. DATA POINT MAPPING APPROACH

For providing a way for common understanding among EMS components, as already mentioned, population of the airport ontology had to be performed in correspondence with the UDPN convention [16] utilized for raw data/message exchange. More precisely, ontology population was carried out based on the BMS data point lists holding raw data semantics compiled according to the UDPN convention. In order to transfer the raw data semantics into the airport ontology, each entity instance was defined with the property values representing the corresponding UDPN attributes. For example, class "signal" instance was defined with properties representing the location of the source device (UDPN attributes "building" and "zone"), (sub)systems to which it belongs (UDPN attributes "system" and "subsystem1/2"), signal type (UDPN attributes "medium", "position" and "kind") and unified data point name representing unique signal ID. In that way, both extension and instantiation of the core airport ontology into two full-blown airport ontology instances (for MXP and FCO airports) was carried out in line with UDPN convention.

In order to extend and populate the airport ontology model according to the UDPN convention, BMS data point lists (i.e. Excel sheets carrying the information about every low-level signal/device that EMS might have to deal with), shown in Figure 4, were mapped into the airport ontology entities. More precisely, the data point lists were used first to extend the core airport ontology model by developing more fine-grained model. This implied identification of sub-concepts of already existing ones, such as definition of different types of sensors/actuators, signals etc. After the core airport ontology was extended, population of model had to be performed. This considered mapping of the raw data semantics, i.e. the corresponding data cells (representing previously mentioned UDPN attributes) from the data point lists into the airport ontology as corresponding entity instance or as a property value of specific instance (such as properties of class "signal" instance).

Population of the information into the airport ontology was mainly performed in fully automated manner, while the "fine tuning" (for instance, establishment of additional relations among newly created and/or mapped instances/properties) was performed subsequently in a semi-automatic manner. These alignment tasks had to be carried out for both airport ontology instances representing the full-blown facility models of MXP and FCO airport.

Having the previously mentioned in mind, the following three different, but yet complementary approaches have been considered for mapping of the information into the airport ontology:

- LODRefine tool [8],
- SPIN mapping [9], and
- SPARQL Update queries [10].

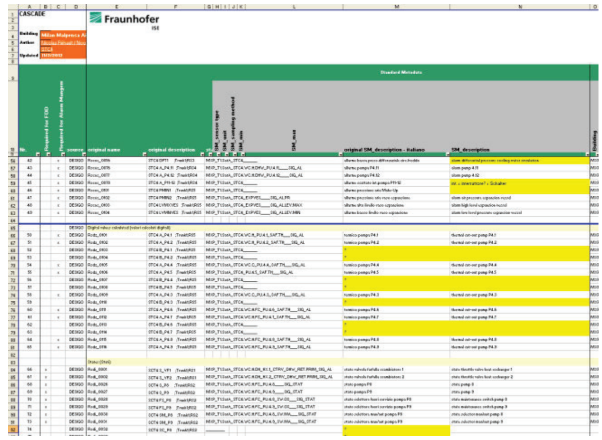

Figure 4. Excerpt from data point lists.

Having in mind their individual functionalities and advantages, these approaches could be implemented separately or in combination. However, for population of the airport ontology to model both MXP and FCO airport facilities, they were applied in combination in order to exploit their full capabilities. To provide the qualitative overview of their particularities, each approach was elaborated in the following subsections.

A. LODRefine transformation

LODRefine [8] is a LOD-enabled version of OpenRefine tool for cleaning, linking and transformation of data from one format into another. It is a part of the LOD2 Stack, which is the output of the FP7 project LOD2 (Grant Agreement No. 257943) and it is comprised of tools for managing the life-cycle of Linked Data. For the purpose of this paper, LODRefine tool was utilized for automatic translation of information stored within BMS data point lists (in form of Excel sheets for both MXP and FCO airports) directly into the RDF triplets simply by defining the translation rules (using the corresponding LODRefine user interface functionalities) as it can be seen in Figure 5.

For definition of data extraction/translation rules, i.e. for establishment of the target RDF triplets, entities and corresponding properties/relations of the core airport ontology were utilized as a base. The result of data translation/mapping is shown in Figure 6 in a form of RDF triplets carrying the extracted semantics (such as signal ID represented in compliance with the UDPN convention, indicated by a red rectangle). Finally, the RDF triplets carrying the extracted information regarding the newly created instances and their belonging property values were subsequently imported (as OWL ontology model carrying the extracted semantics) into the core airport ontology model using the Protégé editor.

In this way, the LODRefine tool was first utilized to perform extension of the core airport ontology and then for the population of the extended ontology model. At the same time, this approach provided the possibility to perform the "raw" mappings and alignment with the

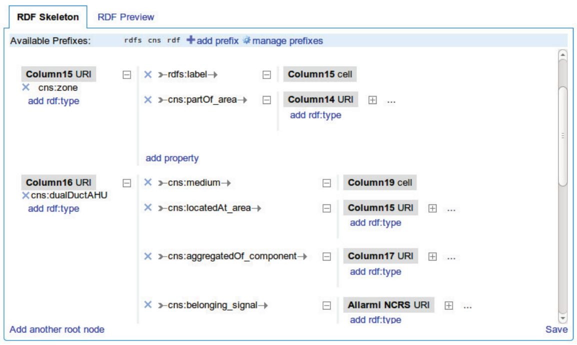

Figure 5. LODRefine translation rules.

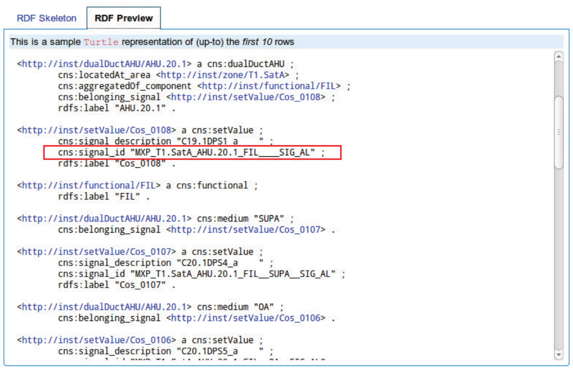

Figure 6. Extracted semantics as RDF triplets (LODRefine)

UDPN convention in completely automatic manner, while the following two approaches were considered subsequently as more convenient for "fine tuning" of the airport ontology in semi-automatic manner as it will be described further.

B. SPIN based mapping

SPARQL Inferencing Notation (SPIN) [9] is a SPARQL-based language used for representation of the mappings between RDF/OWL ontologies. It is used to transform instances of source classes into instances of target classes. Furthermore, it is used for definition of rules that map not only the entire instances but also their specific properties/values. For definition of mapping rules, SPARQL CONSTRUCT keyword was used to map classes from one graph pattern to another while binding the source and target classes was performed in the WHERE clause of the SPIN mapping rule shown below.

```
CONSTRUCT {
    ?target ?targetPredicate1 ?newValue .
}
WHERE {
    ?this ?sourcePredicate1 ?oldValue .
    BIND (spin:eval(?expression, sp:arg1, ?oldValue) AS ?newValue) .
    BIND (spinmap:targetResource(?this, ?context) AS ?target) .
}
```

Following this approach, instances and corresponding relations/properties had to be initially extracted (such as from the BMS data point lists) and imported as RDF

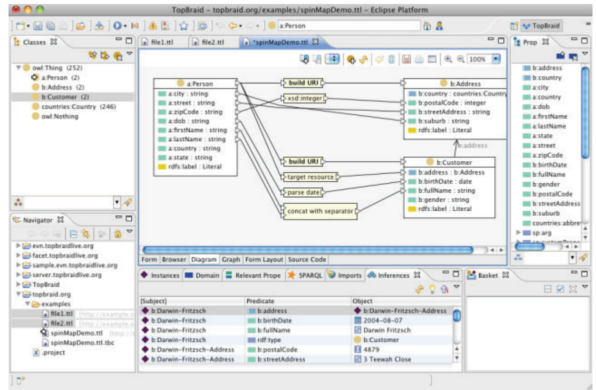

Figure 7. SPIN based mapping.

triplets (simply by importing the corresponding CSV file carrying the source information) that could be further mapped (taken as a source objects with corresponding property values) into the target model, i.e. the entities of the previously extended airport ontology model (by using the SPIN mapping rules). Based on the corresponding pre-defined mapping rules, desired relations were established, not only among already existing instances (generated by using the first approach), but among the newly imported ones as well.

Additionally, SPIN mapping language is a high-level language that is suitable to be edited graphically (simply by drag and drop of source/target classes). This convenient feature was provided by TopBraid Composer starting from release 3.5. Furthermore, it provides a visual editor (shown in Figure 7), i.e. graphical user interface, that significantly facilitated alignment tasks of the airport ontology entities and their interdependencies.

C. SPARQL Update based mapping

Final "fine tuning" of the airport ontology model was performed through the SPARQL Update [10] queries performed from the Java script directly upon the ontology model. SPARQL Update is an extension of the SPARQL query language that provides the ability to add, update and delete RDF data. It enables the generation of specific SPARQL Update queries including commands such as MODIFY (for modifying the existing instances), INSERT (for inserting the newly created instances/property values) and DELETE (for deleting the existing instances/property values) based on which target entity property can be updated according to the handled data. New instances and property values which should be inserted/modified in the ontology model could be manually defined either within the Java script itself (through corresponding query arguments) or extracted from the source Excel sheet handled by JAVA Excel API.

This approach allowed "semi-automatic" handling of the already created ontology instances, i.e. it provided the possibility for an update of the target instances and corresponding properties/relations of airport ontology. For performing alignment and mapping tasks, patterns of SPARQL update queries were defined based on which corresponding modifications were carried out (within the JAVA script). Example of SPARQL Update query is shown below. From the presented example, it could be seen that property "waterFlow_l/s" of the target device "waterPump", having the specific device ID (taken as the input argument "id") , was updated (more precisely, first deleted and then updated) with the new nominal water flow data (taken as an input argument "flow").

```
// SPARQL Update - To replace a property first delete current value and then
insert new one
queryStr.append("DELETE { " +
            "?s pref:waterFlow_l_s ?o. "+
            "} " +
            "WHERE { " +
            "?s pref:device_id \"" + id + "\"^^xsd:string. " +
            "?s pref:waterFlow_l_s ?o. " +
            "} " +
            "INSERT { " +
            "?s pref:waterFlow_l_s \"" + flow + "\"^^xsd:float. " +
            "} " +
            "WHERE { " +
            "?s rdf:type pref:waterPump. " +
            "?s pref:device_id \"" + id + "\"^^xsd:string. " +
            "}");

// Update execution
UpdateAction.parseExecute(queryStr.toString(), model);
```

5. CONCLUSIONS

To provide advanced and more intelligent airport EMS, an ontology-based ADAM layer was proposed in [1] and [2]. The airport ontology, developed as part of ADAM layer, served as a central data repository which provided the needed semantics to the EMS components, thus supporting the integration and interoperability of the overall system. First, the core airport ontology providing a generic model of the airport facility was modelled, which had to be extended and instantiated further to model a specific airport infrastructure. This paper proposes one of the possible approaches to perform the airport ontology modelling and population. In other words, it explains the undertaken modelling approach and the general concepts behind the modelling.

The task was to populate the airport ontology with the static data regarding the airport facility and target systems/equipment (such as significant energy consumers). To provide the input for modelling, first the needed data and relevant semantics were acquired. Then, the core airport ontology had to be extended and populated based on the acquired data to model a specific target airport infrastructure. This included the definition of new concepts, instances and their properties. Two European airports were taken as a test-bed platform for modelling tasks, MXP airport in Milan and FCO airport in Rome. For population of the airport ontology, the BMS data point lists carrying the semantics about the low-level signals were taken into account. Three different, but complementary approaches for translation of the data point lists into the ontology were elaborated: LODRefine tool, SPIN mapping and SPARQL Update queries. An overview of the functionalities for each of the mentioned approaches was provided. In this way, it was possible to instantiate any airport infrastructure of interest.

ACKNOWLEDGEMENTS

The research presented in this paper is partly financed by the European Union (FP7 CASCADE project, Pr. No: 284920), and partly by the Ministry of Science and Technological Development of Republic of Serbia (SOFIA project, Pr. No: TR-32010).

REFERENCES

[1] Tomasevic N., Batic M., Vranes S., "*Ontology-enabled airport energy management*," ICIST 2013, 3rd International conference on information society technology, pp. 112-117, Kopaonik, 2013.

[2] Batic M., Tomasevic N., Vranes S., "*Ontology API for web-enabled FDD system*," ICIST 2013, 3rd International conference on information society technology, pp. 142-147, Kopaonik, 2013.

[3] Gruber T.R., "*A translation approach to portable ontology specifications*", Knowledge Acquisition, Vol. 5 (2): pp. 199–220, 1993.

[4] Dibley M., Haijiang Li, Miles, J., Rezgui, Y., "*Towards a synchronized semantic model to support aspects of building management*", 7th IEEE International Conference on Industrial Informatics, Cardiff, Wales, pp. 307-312, 2009.

[5] Praus F., Granzer W., Kastner W., "*Enhanced control application development in Building Automation*", 7th IEEE International Conference on Industrial Informatics, Wales, pp. 390-395, 2009.

[6] Reinisch C., Kofler M.J., Iglesias F., Kastner W., "*ThinkHome Energy Efficiency in Future Smart Homes*", EURASIP Journal on Embedded Systems, 2011.

[7] Rossello-Busquet A., Brewka L.J., Soler J., Dittmann L., "*OWL Ontologies and SWRL Rules Applied to Energy Management*", 13th International Conference on Computer Modelling and Simulation, Cambridge, pp. 446-450, 2011.

[8] LODRefine tool, Available: http://code.zemanta.com/sparkica/

[9] Knublauch H., Hendler H.A., Idehen K. (eds.), "SPIN - Overview and Motivation", W3C Member Submission 22 February 2011, Available: http://www.w3.org/Submission/2011/SUBM-spin-overview-20110222/

[10] Gearon P., Passant A., Polleres A. (eds.), "SPARQL 1.1 Update", W3C Recommendation 21 March 2013, http://www.w3.org/TR/sparql11-update/

[11] Suggested Upper Merged Ontology (SUMO), http://www.ontologyportal.org/

[12] Industry Foundation Classes (IFC) data model, buildingSMART, http://www.buildingsmart.org/standards/ifc/

[13] Common Information Model (CIM) Users Group, http://cimug.ucaiug.org/default.aspx/

[14] Lacy L.W., Owl: Representing Information Using the Web Ontology Language, Trafford Publishing, 2005.

[15] The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/

[16] Keane M., Costa A., Blanes L., Donnely C., Torrens I., Monaghan P., Brogan M., McCaffrey M., CASCADE ICT for Energy Efficient Airports, Project Deliverable D2.1 – "*CASCADE Methodology for Energy Efficient Airports*", 2012.