SEMANTIC TECHNOLOGIES AND ONTOLOGY MATCHING FOR INTEROPERABILITY INSIDE AND ACROSS BUILDINGS

Jérôme Euzenat, PhD / Senior research scientist, Jerome.Euzenat@inria.fr
INRIA & LIG, Grenoble, France

ABSTRACT

There are many experiments with buildings that communicate information to and react to instructions from inhabiting systems. Fortunately, the life of people does not stop at the door of those buildings. It is thus very important that from one building to another, from a building to its outside, and from a building considered as a whole to specific rooms, continuity in the perceived information and potential actions be ensured. One way to achieve this would be by standardising representation vocabularies that any initiative should follow. But, at such an early stage, this would be an obstacle to innovation, because experimenters do not know yet what is needed in their context. We advocate that semantic technologies, in addition to be already recognised as a key component in communicating building platforms, are adequate tools for ensuring interoperability between building settings. For that purpose, we first present how these technologies (RDF, OWL, SPARQL, Alignment) can be used within ambient intelligent applications. Then, we review several solutions for ensuring interoperability between heterogeneous building settings, in particular through online embedded matching, alignment servers or collaborative matching. We describe the state of the art in ontology matching and how it can be used for providing interoperability between semantic descriptions.

Keywords: Ontology matching, Ontology alignment, Alignment server, Context-based matching, Content-based matching, Context representation, Query mediation.

1 INTRODUCTION

Building information systems may be useful to architects, building owners and managers. However, ultimately, they have to be useful to inhabitants. A building cannot become energy-efficient or a data hub against those who live in it. Hence, a key element of such systems is adaptability to people. This involves understanding at some level the context in which people evolve and taking appropriate actions to support them.

This applies naturally to energy-efficient buildings and thus efficiency must be supported by the whole building information infrastructure. Adapting to seasonal change, weather or energy prices is only a small part of the task. The most challenging part is adaptation to building inhabitants in the context of particular weather or season.

This entails anticipating: raising the heat before people arrive and lowering it before they leave; supporting them while they are here and cleaning up after they leave. An application for dealing with energy efficiency in buildings has to rely on many different sources:

- a local building information system about the building characteristics;
- the web for information such as weather forecast or energy prices;
- the electrical networks for consumption information;
- sensors (light, noise, movement, energy consumption) for monitoring activities;
- user personal information (agenda through phone or web) for planning future move.

This means that this future is not the goal of a single device: PC, TV, set-top box, smartphone, car or building. It has to be taken into account by all of them. Hence these devices need to interoperate. Nowadays, all these devices are particularly heterogeneous. They concentrate on a particular function and do not bother to communicate with other devices. It is typical that an experimental setting, provided for a particular building does not work with the next building. However, if we mostly live in a few buildings, we visit many of them during our daily life and it is not acceptable to be equipped differently for each of those.

So we would like to address here two types of heterogeneity: the heterogeneity between devices available in the building environment and the heterogeneity across buildings. Dealing with this heterogeneity is of utmost importance to ensure the continuity of users experience.

One common way to solve such interoperability problems is to define in advance required information and the way to exchange it. However, the task is already daunting given the quantity of heterogeneous devices. There is a multiplication of devices and efforts with divergent goals: reducing construction costs, long term energy efficiency, constant connectivity, radio-emission reduction, Gemütlichkeit, etc. Moreover, standardising at that stage of research is premature and would hamper the development of technology.

Hence, the solution is in the use of standard but general purpose technologies, i.e., technologies which have not been developed for a particular purpose. Web technologies have these characteristics and semantic web technologies are particularly adapted to the exchange of knowledge and data across a variety of platforms and protocols. Moreover, the benefit of using semantic technologies, and, in particular, ontologies, are already acknowledged in ambient computing and smart buildings, so this does not necessarily introduce new constraints.

However, these technology alone do not solve the ultimate interoperability problem: different devices and information system will use different ontologies. Again, this is perfectly natural and should be handled with the adequate tools. Semantic technologies, and, in particular, ontology matching, allow for overcoming heterogeneity.

In this paper, we show how semantic technologies, developed in the context of the semantic web, are particularly suited for representing information that is embedded, produced and consumed by building information systems. The next section (§2) thus presents semantic web technologies in the context of building information systems and more generally ambient computing. Then, we propose a framework, largely inspired from (Euzenat et. al. 2008), for dealing with heterogeneous ontologies (§4). Finally, we review ontology matching as a key component of this interoperability (§5).

2 SEMANTIC TECHNOLOGIES FOR BUILDING INFORMATION SYSTEMS

In (Euzenat et. al. 2008), we have considered context information management frameworks should be:

- Open, so that new devices and applications can be involved in the environment. It must thus rely on well accepted standards for expressing information which guarantees that components will be able to interoperate.
- Dynamic, so that these devices and applications can be taken into account dynamically. This requires that it can represent new types of information and that it can match these representations so that old parties take advantage of new ones and vice versa.
- Minimal, so that the framework does not put a non realistic burden on application and device developers. This requires to keep minimal the computing resources and specific interfaces needed for using this framework.

To some extent, the same constraints are considered for building information systems in systems such as CSTBox (Zarli et. al. 2010). However, we take here into account the need to achieve interoperability down to the level of data representation.

Because we want to focus on buildings and their inhabitants as they are, most of the information to store in building information systems is indeed context information. Hence, we present below the framework that was proposed for representing context information in pervasive computing (Euzenat et. al. 2008).

We provide example of the use of semantic web technologies for developing an application which needs to assess the temperature in a room and if there is light. This may be for deducing that there is no activity and for cutting heating or indicating that the room is free and suitable for some activities. The important point is that the application may be defined relatively independently from the building model and yet works.

So we first present how information can be modelled in OWL (§2.1) and expressed in RDF (§2.2). Then we explain how OWL ontologies may be extended for expressing more precise information and how SPARQL queries may be used for obtaining information expressed in these extended ontologies without knowing them (§2.3). Finally, we present the use of ontology alignments in order to work with heterogeneous ontologies (§2.4). We only provide an informal presentation of semantic technologies; further details may be found in (Hitzler et. al. 2009).

2.1 OWL ontologies for characterising objects

OWL ontologies are used in order to characterise the objects that can be found within the environment. There are general purpose ontologies such as SUM[1], Cyc[2] or DOLCE[3] that can be used. The essential point is to have ontologies sufficiently generic to cover the various concepts involved in applications: resources, actors, places, dates, activities, permissions, etc. There are several ontologies of this type that have been designed for pervasive computing purposes (Chen et. al. 2004a, 2004b, Wang et. al. 2004, Flury et. al. 2004).

These ontologies are in general not very sophisticated because they were designed to make the machinery of pervasive computing applications work. For each domain, it is necessary to develop a more precise description of exploitable information. (Flury et. al. 2004) proposed a semantic description of four different models (semantical, geometrical, graph theory based, and set theory based models) to represent indoor location information. To cover outdoor location, (Fu et. al. 2005) proposed a geographical ontology which gathers several geographical datasets. (Gandon and Sadeh 2004) presents a semantic description of rules to selectively control who can access to contextual information and under which conditions. A spatio-temporal approach is developed in (Ngoc et. al. 2005) to describe, with a dedicated ontology, user preferences in ubiquitous computing environments and their behavior routine. To describe users and their social relations, the FOAF[4] ontology is appropriate, as well as the GUMO[5] ontology described in (Heckmann et. al. 2007).

More specific representations for buildings are necessary, for instance to take into account building information models (plans, material, lifecycle), equipment (location, provider, characteristics) or energy (cost, capacity, efficiency) (Bourdeau and Laresgoiti 2011).

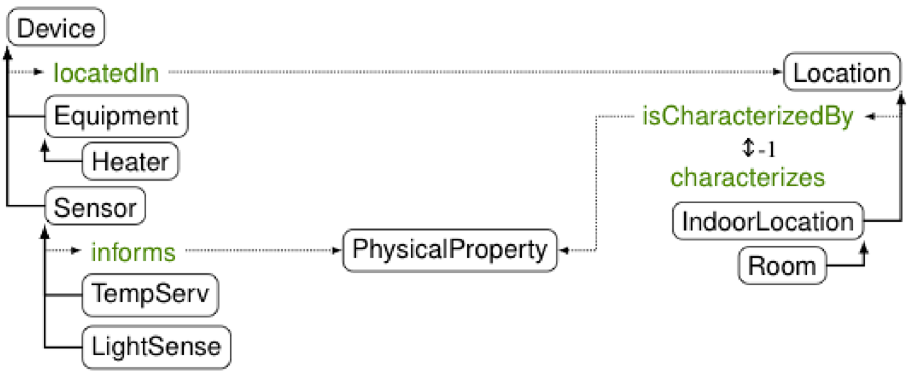

Figure 1: Sample of general purpose ontology concepts (classes are in rounded corner rectangles, properties are related by dotted arrows and plain arrows between classes denote sub-class relationships). The ontology provides classes for devices and locations as well as properties such as "locatedIn".

Ontologies, such as those of Figure 1, can be used by applications to characterize information which is necessary for them. Usually, devices will use the most precise refinements of these models to be exploited by applications.

[1]http://reliant.teknowledge.com/DAML/SUMO.owl
[2]http://www.cyc.com/2003/04/01/cyc
[3]http://www.loa-cnr.it/ontologies/DLP_397.owl
[4]http://xmlns.com/foaf/spec/
[5]http://www.ubisworld.org/

2.2 RDF graphs for modelling information

Information can be represented as RDF graphs (Klyne and Carroll 2004). An RDF graph is simply made of a set of triples relating entities (classes, instances, literal values) through properties (see Figure 2). The benefit of using a general purpose language like RDF is that a common general interface can be defined for components that exchange RDF triples. Interoperability is then trivially guaranteed by considering that they are consumers and producers of RDF. This attitude has been adopted worldwide from linked data to mobile phones.

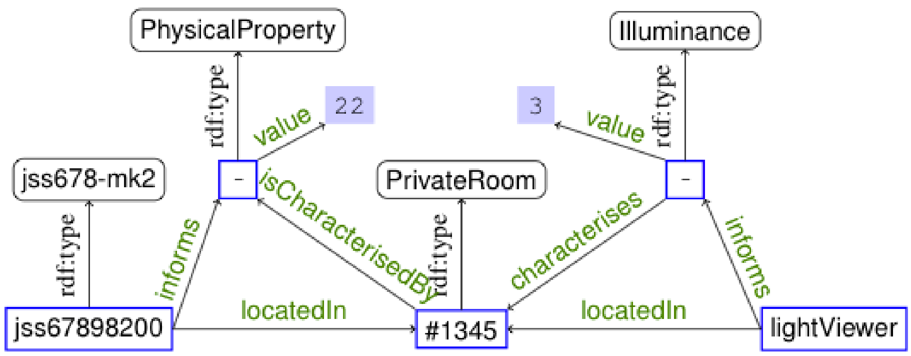

Figure 2: Part of the information concerning a private room in a building can be represented by the following set of triples (classes are still in rounded corner rectangles, instances are in rectangular boxes, data is in blue rectangles; instances are related to their classes through rdf:type properties). It presents two sensors providing information about physical properties of the room (one of these properties is illuminance).

Figure 2 illustrates that information may not be expressed with regard to the general purpose ontologies as presented in Figure 1.

In order for applications to know which devices to query, devices must publish the query types to which they can answer. This can be achieved by publishing the classes of objects and properties on which the component can answer. Ontologies are the natural way to achieve this and OWL is particularly suited for designing shared ontologies.

2.3 Matching information needs to actual information

Devices can be added at any time (which occurs when new people enter a room for instance). There is no reason, a priori, that added devices as well as new applications are really compatible. Indeed, each newly introduced sensor will provide more precision or information which has not been considered at application design time. In the same way, the applications cannot know all kinds of available sensors. Ontology description languages can help solving this problem. Fortunately, knowledge representation techniques in the OWL language always permit to specify a concept or a property without questioning those which existed originally.

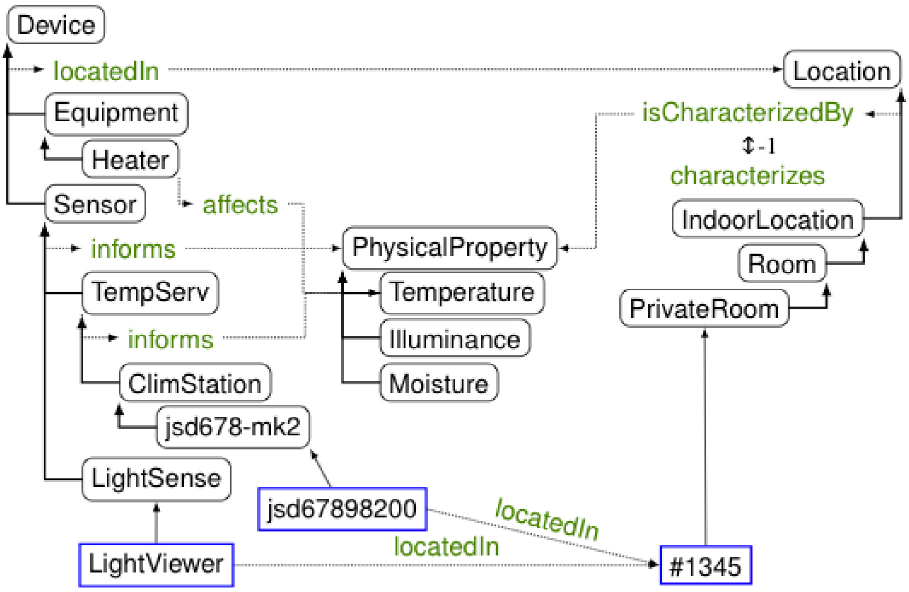

Figure 3: In a particular environment, general purpose ontologies (Figure 1) are refined into more specific ontologies. They introduce new concepts (Heater, TempServ) and new constraints on existing concepts (the "informs" value of a TempServ is a Temperature). Here, instances of the building model are presented (clim station, LightViewer and private room: instances are denoted by grey rectangles and linked to their classes by thin arrows).

Let assume that an application wishes to know the temperature in the room and if there is light. A high level ontology enables to characterize its needs: the temperature and the illuminance are physical properties of the room. For that purpose, a query language like SPARQL (Prud'hommeaux and Seaborne 2006) may be useful for querying or subscribing to sources. The requested information (query) may be expressed by the graph pattern depicted in Figure 4 which corresponds to a set of RDF triples.

Graph patterns, such as presented in Figure 4, are the main components of SPARQL queries. However, it is not necessary to require devices to answer SPARQL queries. The ability to traverse RDF graphs is most of the time sufficient. Another benefit of using RDF and SPARQL is that services do not necessarily need to tell in advance to which query they are able to answer: applications can ask query that were not anticipated.

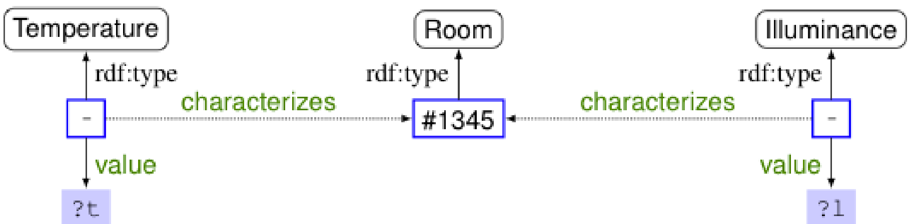

Figure 4: A query graph pattern (question marks instead of instance names introduce free variables). It can correspond to the information that an application requires from the environment (here the temperature and illuminance of a room). This information is expressed in function of the general ontologies of Figure 1.

In the sample application, the important issue is that the class of the sensors are temperature and light sensors. That these properties can be obtained by a ThermServer sensor, like a thermometer, located in the same room is not the relevant to the application. On the other hand, if the goal of the application is to reduce the clim when the outside temperature is low, it is important to know that this is not only a sensor. Using ontologies to express information permits a new equipment whose capabilities have not been known at application design time to enter and new applications to benefit from these possibilities.

Similarly, the application does not need to know that a room is a private room for artist to dress up or a public room that is left to the children to practice music instruments if the application only needs to know if someone is active in this room and what is its temperature. But nothing prevents the building model from declaring it like this. This will characterize the room precisely. Figure 3 shows how the ontologies of Figure 1 can be extended for the purposes of using sensors in room #1345.

The information required to answer the query can be found in the RDF graph of Figure 2, but it needs the content of the given ontology to do so. Indeed, the query pattern asks for a “Temperature”, but nothing is qualified as such in the RDF graph. However, the information provided by the temperature server have to be a “Temperature” according to the ontology of Figure 3, hence there is a temperature available. On the other side, the output of the LightSensor is not related to the room by “characterizes”. However, the ontology tells us that this is the inverse of “isCharacterizedBy” so we can use it as a valid answer for the query. Hence, the temperature of 22°C and the illuminance of $3lm/m^2$ can be extracted from the graph by the query with the help of the ontologies.

This example shows the benefit of using semantic web technologies for dealing with this information: ontologies disseminated on the web provide the background knowledge necessary to interpret raw information.

The information is described by the device more precisely than in the query: the query does only wants to identify the temperature and illuminance for the room. The applications must be as general as possible when describing their needed information (the room temperature, the activity) whereas the information management systems must be as precise as possible on what they produce. That will permit the most specialized applications to take advantage of them.

2.4 Alignments

The proposed information management system makes it possible to introduce new devices in the environment by extending the ontology in such a way that existing applications can make the best use of them. However, this view holds if all parties share the same ontology. This is not always the case.

Indeed, two sensor manufacturers will probably use two different ontologies to describe their products. Moreover, applications will probably rely on domain ontologies related to their scope. For instance, Figure 6 shows on the left-hand side an ontology in which areas are classified with respect to their function while on the right-hand side, access is considered instead.

Reconciling these heterogeneous ontologies may be achieved through the use of ontology alignments. An alignment is as set of correspondences between ontology entities. Figure 6 presents such an alignment which expresses relations between the function ontology and the access ontology. It is visible that the latter is more general than the former: Backstage, Dressing and GreenRoom are more specific than Private Room and Parking is more specific than Outdoor location.

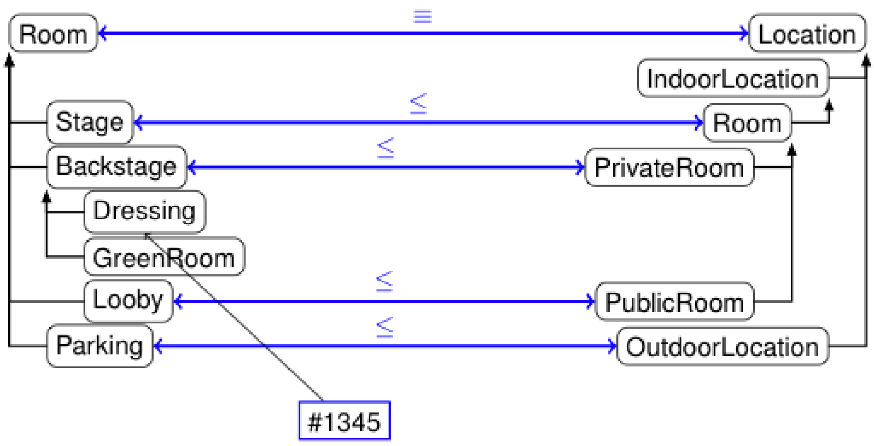

Figure 6: Two ontologies (left: functional, right: access) and an alignment. Each correspondence, i.e., blue arrow, relates entities of each ontology with a relation (≡ for equivalence and ≤ for less general). Room #1345 being a Dressing, it is necessarily a Room in the access ontology because Backstage is more specific than PrivateRoom.

The application is able to use the alignment for translating queries before sending them to devices or applications, and eventually translating the answer back. Indeed, if the room #1345 is a dressing in the functional ontology, it will still be an answer to the query of Figure 4 with the help of the alignment (the query asks for a Room which is more general than PrivateRoom, which is more general than Backstage, which is more general than Dressing).

Alignments may be more precise than the one of Figure 6 by providing confidence about the given correspondences and relating other entities such as properties or compound expressions (David et. al. 2011).

3 ONTOLOGY MATCHING

As we have seen, interoperability can be achieved through alignments between the ontologies used by different parties. Ontology matching consists of generating an alignment from two ontologies, that can be used for various purposes such as merging ontologies, transforming data or querying. Ontology matching is actively researched and many algorithms have been provided for finding correspondences (Euzenat and Shvaiko 2007). Different features of ontologies are usually used for performing matching. Beside the classification provided in (Euzenat and Shvaiko 2007), we now consider that there are two broad categories of matchers: content-based matchers (§3.1) and context-based matchers (§3.2).

3.1 Content-based matchers

Content-based matchers are those matchers using the content of the ontologies in order to match them.

- terminological techniques are based on the text found within ontologies for identifying ontology entities (labels), documenting them (comments) or other surrounding textual sources (related element labels). These techniques come from natural language processing and information retrieval. They can use the string structure themselves, e.g., string distances, or the ontology as corpus, e.g., statistical measures based on the frequency of occurrence of a term.
- structural techniques are based on the relations between ontology entities. These can be relations between entities and their attributes, including constraints on their values, or relations with other entities. These techniques take advantage of type comparison techniques or more elaborate graph techniques, e.g., tree distances, path matching, graph matching.
- extensional techniques compare the extension of entities. These extensions can be made of other entities, e.g., instances.
- semantic techniques are based on the semantic definition of ontologies. They use extra formalised knowledge and theorem provers for finding consequences of a particular alignment. This can be used for expanding the alignment or, on the contrary, for detecting conflicting correspondences.

Of course, most of the systems combine several techniques in order to improve their results. The techniques can be combined by aggregating distance results (van Hage et al. 2005), by using selection functions for choosing which one to use in the present case (Jian et al. 2005, Tang et al. 2006), or by deeply involving them all in global distance computation (Euzenat and Valtchev 2004, Melnik et al. 2002).

In the present case, all these techniques may be used for online matching but extensional techniques. Indeed, it is unlikely that devices will have much instances to offer at the moment of matching.

3.2 Context-based matchers

Context-based matchers take advantage of the connections ontologies have with a broader context in order to assess the correspondences between their components. These connection may come from different sources:

- Existing alignments with other ontologies, hence, having two ontologies already matched to a third one may be easier to match. There may be different approaches depending on the type of

ontology selected: this may be a top-level ontology, a reference ontology for a particular domain or even all the ontologies of the web.
- Specific resources: These may be dictionaries defining words used in the ontology labels, multilingual lexicon providing the translation between several languages or encyclopedia such as wikipedia or dbpedia[6], its semantic web counterpart.
- Annotated resources: two ontologies used to annotate the same type of resources, e.g., web pages, pictures, products in a catalogue, offer the same possibility as extensional techniques. The larger the annotated resources, the easier it is to use statistical or data analysis techniques. The approaches differ depending on whether the two ontologies share resources, e.g., they index the same set of documents, or not (in which case a similarity between the extensions may be established).

Usually, context-based matchers are slower than content-based matchers for two reasons: they usually rely on large datasets and they involve combinatoric searches (to potential resources, potential matches and reconciliation).

4 ENSURING INTEROPERABILITY

In the context of building information systems, agreeing on standard universal and self-contained ontologies is not a reasonable assumption. Furthermore, such an approach will probably hamper the development of ontologies and technology. Hence, we have to rely on alignments. Defining a priori all alignments between all the possibly encountered ontologies suffers from the same problems as standardising ontologies. Not all the work on ontology matching is relevant to building information systems. In order to choose matching techniques for a particular applications, it is necessary to consider its characteristics. Indeed, the characteristics of matching in such systems requires that it be:

- automatic: it is not possible to rely on the user to directly help matching, and indirect help cannot postulated because there will not always be users;
- fast: when a process needs an alignment it cannot wait for hours to have the results, hence, either matching should be processed online or precomputed alignments must have been stored;
- correct: it is rather important the provided alignments be correct even if some level of fault-tolerance is possible; it is less important that it be complete.

We identify three possible approaches to obtain such alignments that we consider below: online embedded matching (§4.1), ontology alignment service (§4.2) or collaborative alignment (§4.3).

4.1 Online embedded matching

Online embedded matching consists, for each application willing to communicate with the environment, and this applies to building information systems as well, to be able to match ontologies on the fly. This does not seem to be a reasonable option due to the important resource consumption that may be involved in this matching task. Usually matchers have to compromise speed for correctness: some matchers are very fast and provide good results, but not fully accurate results. Given the requirements here, online embedding matching does not seem reasonable.

In addition, this may lead the application to rely only on matchers it embeds instead of taking advantage of the many new matchers available each year. An alternative solution is still to perform online matching but to rely on an external Alignment server able to perform matching on the fly. In this case, the server may provide many matchers and update them.

4.2 Ontology alignment service

Alignment servers (Euzenat 2005) help agents (information managers and applications in this case) to find an alignment between different ontologies they face. They provide mechanisms for:

- Archiving (and retrieving) past alignments;
- Dynamically matching two ontologies;
- Translating queries and answers to queries between information managers that use different ontologies;

[6]http://dbpedia.org

- Finding out an ontology close to a specific ontology (this can be useful for finding intermediate ontologies which will facilitate matching).

An Alignment server uses a functional interface that allows the explicit handling of ontologies alignments that have been developed in the framework of the semantic web[7]. It could be invoked as a web service or through specific communication interfaces. Such a server could even directly embedded in centralised mechanisms such as the CSTBox (Zarli et. al. 2010) to be made available to various applications and devices.

One important feature of alignment servers in this context is their ability to store well identified and certified alignments that may be shared across applications. Indeed, the necessary alignments will often be the same, across different applications and different device providers, hence it will be convenient to share them across matchers.

4.3 Collaborative matching

Finally, Alignment servers may also be used for supporting collaborative matching in which application and devices only use parts of alignments, e.g., the correspondences needed to transform queries, and report when these correspondences are useful or when they lead to errors. This helps the server to rank correspondences and alignments and to improve its answers over time.

These three approaches are not incompatible and might even be used concurrently. For example, parties could agree on sharing common high level ontologies and leaving more specific ontology evolve freely and independently. This is a strategy enabling a close account for a fast evolving domain.

5 CONCLUSION

We have considered the problem of deploying building information systems that can fruitfully interoperate with their environment (inhabiting devices, surrounding buildings, or larger infrastructure) and yet deal with expressive information. We showed how semantic technologies, as developed for the semantic web, may be used for that purpose: RDF for expressing data, OWL for defining the vocabularies, SPARQL for asking queries and alignments for bridging ontologies. We provided a quick survey of ontology matching and described how alignments may be provided to applications and devices.

ACKNOWLEDGMENTS

The work on the Alignment server has been partially supported the European Integrated Project NeOn (IST-2005-027595) and ANR project WebContent.

REFERENCES

Bourdeau, M. Laresgoiti, I. (2011) “Energy and Behavioural modelling for managing energy resources in residential buildings.” this volume.

Chen, H., Finin, T., Joshi, A. (2004a) “An ontology for context-aware pervasive computing environments.” *Knowledge engineering review* 18(3):197–207.

Chen, H., Perich, F., Finin, T., Joshi, A. (2004b) “SOUPA: Standard ontology for ubiquitous and pervasive applications.” In Proc. 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston (MA US), pp258–267.

David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C. (2011) “The Alignment API 4.0.” *Semantic web journal* 2(1):3-10.

Euzenat, J. (2005) “Alignment infrastructure for ontology mediation and other applications.” In M. Hepp, A. Polleres, F. van Harmelen, and M. Genesereth, editors, Proc. 1st ICSOC international workshop on Mediation in semantic web services, Amsterdam (NL), pp81–95.

Euzenat, J., Valtchev, P. (2004) “Similarity- based ontology alignment in OWL-lite.” In Proc. 15th European Conference on Artificial Intelligence (ECAI), Valencia (ES), pp333–337.

Euzenat, J., Shvaiko, P. (2007) “Ontology matching.” Springer, Heidelberg (DE).

[7]http://alignapi.gforge.inria.fr/

Euzenat, J., Pierson, J., Ramparani, F. (2008) "Dynamic context management for pervasive applications." *Knowledge engineering review* 23(1):21-49.
Flury, T., Privat, G., Ramparany, F. (2004) "OWL-based location ontology for context-aware services." In Proc. Artificial Intelligence in Mobile Systems, Nottingham (UK), pp52–58.
Fu, F., Jones, C., Abdelmoty, A. (2005) "Building a geographical ontology for intelligent spatial search on the web." In Proc. IASTED International Conference on Databases and Applications (DBA), Innsbruck (AT), pp167–192.
Gandon, F., Sadeh, N. (2004) "Semantic web technologies to reconcile privacy and context awareness." *Journal of web semantics* 1(3):241–260.
Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., Krner, A. (2007) "The user model and context ontology GUMO revisited for future web 2.0 extensions." In Proc. 3rd Contexts and ontologies workshop Roskilde (DK), pp37–46,.
Hitzler, P., Krötzsch, M., Rudolph, S. (2009) "Foundations of semantic web technologies", Chapman & Hall/CRC.
Jian, N., Hu, W., Cheng, G., Qu, Y. (2005) "Falcon-AO: Aligning ontologies with Falcon." In Proc. K-CAP Workshop on Integrating Ontologies, Banff (CA), pp87–93.
Klyne, G., Carroll, J. (2004) "Resource description framework (RDF): Concepts and abstract syntax." Recommendation, W3C.
Melnik, S., Garcia-Molina, H., Rahm, E. (2002) "Similarity flooding: a versatile graph matching algorithm." In Proc. 18th International Conference on Data Engineering (ICDE), San Jose (CA US), pp117–128.
Ngoc, K. A. P., Lee, Y.-K., Lee, S. (2005) "OWL-based user preference and behavior routine ontology for ubiquitous system." In OTM Conferences (2), Agia Napa (CY), *LNCS* 3761:1615–1622.
Prud'hommeaux, E., Seaborne, A. (2006) "SPARQL query language for RDF. Working draft, W3C.
Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K. (2006) "Using Bayesian decision for ontology mapping." *Journal of web semantics* 4(1):243–262.
van Hage, W., Katrenko, S., Schreiber, G. (2005) "A method to combine linguistic ontology-mapping techniques." In Proc. 4th International Semantic Web Conference (ISWC), Galway (IE), *LNCS* 3729:732–744.
Wang, X. H., Zhang, D. Q. , Gu, T. , Pung, H. K. (2004) "Ontology based context modeling and reasoning using OWL." In Proc. 2nd IEEE Conference on Pervasive Computing and Communications Workshops (PERCOMW), Washington (DC US), pp18–47.
Zarli, A., Pascual, E., Cheung, D. (2010) "Information and communication technology for intelligent and integrated control in buildings: current developments and future research." In Proceedings of the CIB W78 2010: 27th International Conference, Cairo (EG).