Enhancing Alignment Results in Ontology Matching for Smart Cities

Lorena Otero-Cerdeira, Francisco J. Rodríguez-Martínez and Alma Gómez-Rodríguez

LIA2 Research Group. Computer Science Department. University of Vigo. Spain
{locerdeira, franjrm, alma}@uvigo.es

Abstract. In this paper we propose the use of an *ontology matching* algorithm to guarantee the interoperability of the different agents that integrate an smart city. In this sort of environment the different parties need to cooperate and to integrate their information in order to provide enhanced services to the users of the smart city. As the information of these parties may be described by means of different and heterogeneous ontologies, we find the solution in the use of ontology matching techniques. The algorithm presented was designed to be able to exploit the knowledge of previous matched agents to enhance the results and provide the most accurate results possible.

Keywords: internet of things, smart cities, ontology, ontology matching, alignment reuse

1 Introduction

In the last years there has been a remarkable increase in the amount of projects and initiatives related to *Internet of Things* [1] and *Smart Cities* [2]. The Internet of Things is the evolution of the information and communication technologies (ICT), that is taking us from having connectivity at anytime and anyplace to also having it with anything. This situation is reflected by the growing amount of different devices with connecting capabilities, such as RFID tags, NFC devices, sensors, actuators, etc. Such devices are the building blocks of the smart cities.

The idea behind integrating these devices in a city is to turn it into a smart one, so citizen's lives can be improved with new types of services and comfort. These services can be related to almost every aspect of city life and infrastructure, water and energy supply, transportation, healthcare, education, etc. [23], and precisely the cities are looking at ICT to offer services to citizens while reducing costs and improving efficiencies.

To turn a city into a smart one, the first task to address is to develop a rich environment of networks that support digital applications [2]. This task involves, firstly deploying the proper infrastructure which includes different types of sensors, smart devices and actuators, together with the actual networks that allow the communication of these. However, the devices by themselves are not enough and it is necessary to develop applications that exploit these networks of devices.

Hence, in a smart city, smart devices, Sensor Networks (SNs) [23] and applications to exploit them, assume a crucial role. These urban sensors and sensor networks are generally spread over a wide area and continuously measuring different variables. The data collected is processed by the different applications which may trigger an action in some actuator or the response to a user's request.

It is highly likely that the deployment of a smart city is not done all at once but in a series of steps, so it is equally likely that different parts of the smart city are developed by different parties, resulting in the coexistence of different public and private deployments, each one of which possibly using different smart devices and also their own hardware and software architectures. It is necessary to guarantee the success of a smart city to put a special interest in allowing that these different deployments will be able to interact and seamlessly communicate with each other and, that the information gathered by the different devices will be properly integrated and shared among the different systems [26].

This problem is not new to the research community and several alternatives have been already proposed [26][25][11]. These approaches propose using a wrap for the different sensors, or compel the use of some standard or protocol to allow the communication between parties with different knowledge representations. Other efforts include the use of ontologies to semantically describe services and devices available [3]. The work that we have developed is in line with the latest but what we propose exploits ontologies differently.

Our proposal includes the use of *ontology matching* techniques [4] to guarantee the connectivity among the different parties in a smart city.

The remaining of the paper is organized as follows. In section 2, we delve into the use of ontology matching in smart cities and provide the foundations that supported the development of our system. In section 3, a description of our solution is provided and discussed. Finally in section 4, the main conclusions and future lines of work are summarized.

2 Ontology Matching in Smart Cities

A smart city may be seen as a distributed system where several agents on behalf of their users collect data from the environment by using different sensors. The concept of user here should be globally understood, as the user of an agent may be a citizen, a smart device, an application, another agent, etc. The use of ontologies in smart cities is not new as there is for instance the *SCRIBE* ontology [24] [5] designed out of the information gathered from different cities or the $SOFIA^2$ platform [6]. Ontologies help in providing a vocabulary to describe a certain domain and the specification of the meaning of the terms in that vocabulary [4], in our concrete case, ontologies help in defining the different events, entities and services in a smart city. Besides, they are particularly suitable for describing the meaning of the concepts in a communication process between the agents in a Multi-agent System (MAS) [7] and hence they are used as a way of reducing the semantic gap among the different interacting parties.

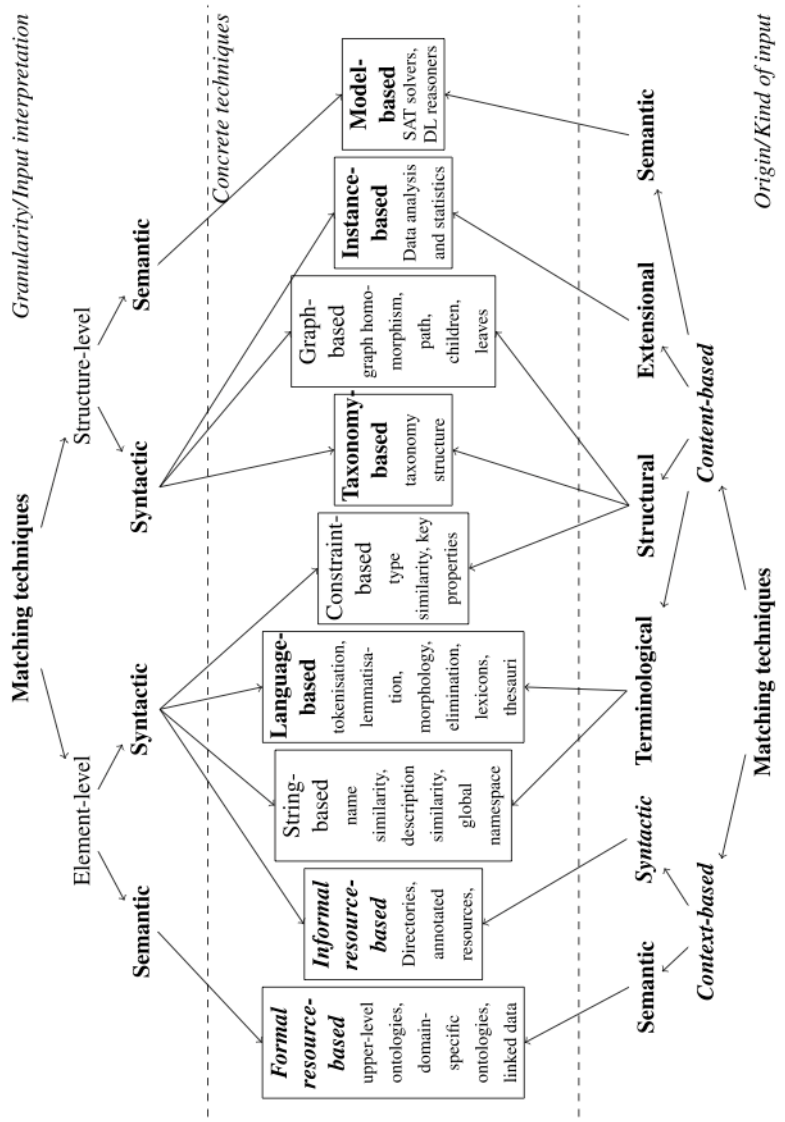

Fig. 1. **Fig. 2.** Classification of matching techniques

However, there are several reasons why ontologies by themselves are not enough to guarantee the interoperability of the different agents. For instance, the agents may use different ontologies to represent the information gathered from the sensors, the software applications in the smart city may be developed by different providers that represent their internal knowledge using different ontologies, there may be agents or applications included in the smart city in a later stage or even itinerant agents that only need a concrete service at a certain time. In order to actually reduce the heterogeneity in the definitions and allow a seamless communication of the parties, we relied on *ontology matching techniques* [8].

These techniques allow the identification of *alignments* for pairs of ontologies where an alignment identifies the set of correspondences holding between the entities belonging to the ontologies [9]. Apart from the manual identification of correspondences fulfilled by human experts which has been practically dismissed due to its cost, there are automatic and semi-automatic methods to compute the alignment between the ontologies which exploit different features of the ontologies or use external resources to identify the possible correspondences between the concepts.

Different classifications have been made for the matching techniques although for the scope of this paper, we followed the one that Euzenat and Shvaiko propose in [4]. This classification, as shown in figure 1[19], can be read both top-down, then stressing the interpretation that the different techniques provide for the input information, and also bottom-up, focussing on the type of input that the matching techniques use. Regardless of the direction of the reading, they both meet at the *concrete techniques* layer.

In the following section, while describing our solution to the ontology matching problem in smart cities, we briefly describe the different techniques that we have used linking them to this classification.

3 Solution Description

In this section we briefly describe our algorithm for ontology matching in smart cities and how we have enhanced its results by following an *alignment reuse* [12] approach.

It takes as input two OWL [18] ontologies and relies on the exploitation of some initial correspondences which we named *binding points* and which are similar to the *anchors* initially used by systems such as *LogMap* [13], *Anchor-Flood* [14], *Anchor-Prompt* [15] or *ASCO* [16], although the procedure followed to compute the binding points is remarkably different to the one used to obtain the anchors in each one of these systems.

These initial correspondences are obtained by using some *language-based* and *terminological techniques*. The language-based techniques consider names as words in a natural language and exploit their morphological features. Some of the methods used, as part of the pre-processing of the strings, are *tokenisation*, that consists of splitting words into shorter sequences by means of a separator (blanks, punctuation marks, camel-case changes, etc) and *stopword elimination*, that consists of removing words such as articles, prepositions, etc.

On the other hand, the terminological techniques consider their inputs just as strings and apply string-distance measures to asses the similarity between two entities. In our case we have used *Jaro-Winkler distance* [10] and *Levenshtein distance* [10] on the

[19] Extracted from the book Ontology Matching [4]

pre-processed strings. The results of these distances are weighted in order to obtain an only lexical value for each pair of entities in the ontologies to match. To weight the results of these measures, another similarity distance is used, in this case, it is based on the exploitation of *WordNet* [17] as a external resource. This is also a language-based technique that takes advantage of the definitions provided by this lexical database to evaluate the distance between two terms.

Once the similarity between the terms in the ontologies has been determined, only those pairs with the highest value are selected to become the *initial binding points*.

These initial correspondences sequentially undergo several procedures that take advantage of some *structural* features of the ontologies and that allow the discovery of new binding points. These binding points can identify both pairs of classes or properties. Each one of the newly discovered binding points is assigned a tag that identifies the procedure and branch within it that led to its discovery. If a binding point is reached by several procedures, all the tags are recorded.

1. *Properties Inverse Procedure:* this procedure retrieves new correspondences between properties by exploiting the existence of inverse properties defined with the construct *owl:inverseOf.*
2. *Properties Domain Range Procedure:* this procedure obtains new correspondences between classes by comparing the domains and ranges of the initial properties. Not only the first-level domain and range classes are evaluated but the procedure continues until reaching the higher levels of the hierarchy.
3. *Classes Properties Procedure:* this procedure allows the retrieval of both new correspondences between classes and properties. This procedure recursively identifies the similar properties existing among the class correspondences, and then assesses the existence of other classes belonging to the domain or range of this properties that could be a new correspondence.
4. *Classes Family Procedure:* this procedure retrieves new correspondences between classes. It exploits the familiar relations of the classes. For each pair of them, its superclasses, subclasses, and sibling classes are evaluated to determine the existence of new possible matches.

These procedures are iteratively applied until no new correspondences are discovered. Once these procedures have finished all the correspondences that have been discovered are filtered to produce the final output of the algorithm. To do so, the tagging is very important as it allows the identification of the different procedures and sub-procedures. It is based on the idea that the different procedures exploit different structural features of the ontologies and hence the likelihood that the obtained results are good is not the same for all of them.

To evaluate the performance of the algorithm we intended to use ontologies from the smart city domain. However, the amount of ontologies in this area proved not to be enough to allow an accurate evaluation. Hence we have used the testbed provided by the *Ontology Alignment Evaluation Initiative 2013* [19] (OAEI-13) which provides different series of tests to evaluate the performance of a matching algorithm. This is usually done by using the standard information retrieval metrics of *precision, recall* and *f-measure* [4].

— *Precision:* measures the ratio of correct correspondences over the total number of returned ones. It reflects the degree of *correctness* of an algorithm.

$$precision = \frac{\#true\_positives}{\#correspondences\_found} \quad (1)$$

— *Recall:* measures the ratio of correct correspondences over the number of expected ones. It reflects, the degree of *completeness* of an algorithm.

$$recall = \frac{\#true\_positives}{\#existing\_correspondences} \quad (2)$$

— *F-measure:* is a measure introduced to compare the systems with just one value since it is highly likely that the system with a higher recall may have a lower precision and vice versa.

$$f-measure = \frac{precision*recall}{(1-\alpha)*precision+\alpha*recall} \quad (3)$$

These measures were used to evaluate the performance of our algorithm. Among the range of tests at the OAEI-13 we have tested our algorithm with several of them, although for the scope of this paper we will be focussing on the *conference* track which aims at finding alignments within a collection of ontologies from the domain of conference organization. The results obtained by our algorithm for each pair of input ontologies are compared with a reference alignment also obtained from the OAEI-13 website. In table 1 we include the average results obtained for this task.

Table 3. Average values obtained in the conference track

Precision	Recall	F-measure
0.86	0.57	0.67

In the smart cities domain, there is a series of ontologies that describe the resources and services that are available for the agents. If an agent needs a certain resource or service, it will need to match its ontology to the appropriate one in the smart city. Depending on where the service or resource is deployed, the agent may need to match its ontology to a part of the ontology that describes the smart city itself, usually when the agent needs access to a resource, or to another agent's ontology, usually when the agent needs a service that is offered by the other one. This situation is depicted in figure 2. In any case, this process will output an alignment between both ontologies. If several agents need to access the same resource or service, the process will be repeated several times.

Our intuition is that if a new agent arrives in the smart city and is willing to use a service or resource, the alignments previously obtained from other agents may help in tuning the alignment process for this new agent and therefore they may be used to enhance the results produced by the algorithm.

This led us to delving into *alignment reuse techniques* [20] which in spite of not being a particularly used matching technique [4], it was precisely the one that better met the our requirements. This technique is grounded on the idea that when describing an application domain the ontologies to be matched are similar to already matched ones and hence this knowledge may be reused. This idea was implemented in the *COMA* [21] and *COMA++*[22] systems which are two of the most well-known ones and that have been continuously evolving since 2002 to include new matchers and features.

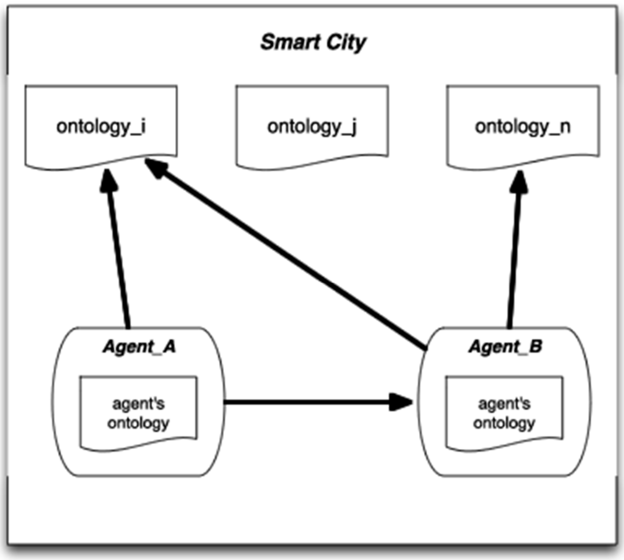

Fig. 2. **Fig. 3.** Smart City

To asses the viability of integrating alignment reuse as part of the ontology matching proposal for smart cities, we have used the ontologies of the conference track. The procedure followed to do so is to feed the algorithm with some intermediate alignments that are then used to identify binding points between the ontologies to match.

Consider the following example, let us suppose that there are three different ontologies, A, B and C, and that we need to match ontology A to ontology C. If we also have available the alignments between A and B ($Align\_A\_B$) and, B and C ($Align\_B\_C$), then it is possible to identify a path that, using these intermediate alignments, may link entities in A to entities in C. We refer to this as a *ring* between A and C through B, and it is graphically represented in figure 3.

Tables from 2 to 6 show the results of testing this approach with the ontologies of the conference track. From the ontologies available at this track in the following examples we have used the following ones: *cmt*, *conference*, *confOf*, *edas*, *ekaw* and *sigkdd*.

Table 2 shows the results obtained by directly matching the *conference ontology* to the *confOf ontology*. These values are included to provide a baseline to compare the results obtained when using rings.

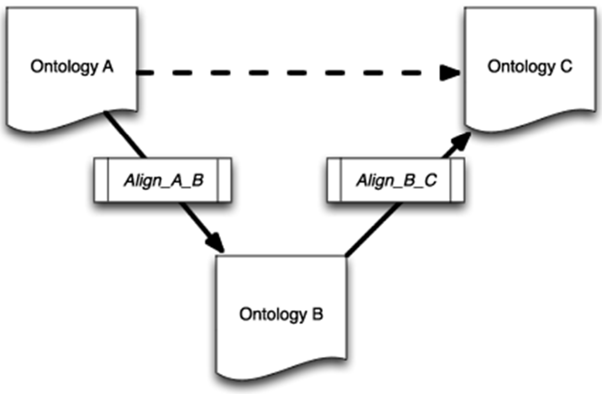

Fig. 3. **Fig. 4.** Ring

Table 4. Results obtained without using any ring

conference - confOf	
Precision	0.9
Recall	0.6
F-measure	0.72

Tables 3 and 4 show two different sets of results obtained when using an additional ontology as ring. Table 3 contains the set of results obtained when matching *conference* to *confOf*, using as additional input the alignments output when matching *conference* to *edas* and *edas* to *confOf*. Table 4 presents the set of results obtained using *ekaw*. As we can observe, in any case, the values obtained are better than those in table 2. However, the improvement using *edas* was more noticeable in the *recall*, while the improvement using *ekaw* was in the *precision*.

Table 5. Results obtained using *edas* for the ring

conference - (edas) - confOf	
Precision	0.86
Recall	0.80
F-measure	0.83

Table 6. Results obtained using *ekaw* for the ring

conference - (ekaw) - confOf	
Precision	1.00
Recall	0.73
F-measure	0.84

When using *edas* for the ring, 10 different paths from *conference* to *confOf* were detected which allowed the identification of 5 new correspondences. Using *ekaw*, just 8 different paths were identified which added 2 new correspondences that were not detected when directly running the matching process. We considered then a combined approach using both *edas* and *ekaw* at a time, seeking to obtain results with the precision enhancement provided by *ekaw* and the recall enhancement provided by *edas*. The results obtained are shown in table 5.

Using a multiple ring the amount of identified paths rises to 13. The results obtained with this approach show that precision is not as high as when just using the single ring with *ekaw* as there is an extra incorrect correspondence that is added to the final output. However, for recall and F-measure, the values obtained are remarkably higher that those obtained with single rings and when compared to the baseline results we observe an improvement of 10.95% for precision, 62.26% for recall and 35.48% for f-measure.

Table 7. Results obtained with *edas* and *ekaw* used at the same time

conference - (edas &ekaw) - confOf	
Precision	0.92
Recall	0.86
F-measure	0.89

In spite of being a positive outcome, the results vary when using the alignments with other ontologies as rings. Table 6 shows the results obtained when considering the alignments with *cmt* and *sigkdd* for the rings. The results in this table also show an improvement compared to the baseline in table 2, although they are not as remarkable as those in table 5.

Table 8. Results obtained with *cmt* and *sigkdd* at the same time

conference - (cmt & sigkdd) - confOf	
Precision	0.90
Recall	0.66
F-measure	0.76

Other tests run using more alignments showed no improvement compared with using just two as in the examples presented previously. However the testbed that we used is not large enough to entirely dismiss that possibility. An issue that we have identified is that even when using the reference alignments provided by the OAEI, which are the golden standard used to compare the results of any algorithm, there were some paths that we identified, that led to selecting as binding points pairs of entities that then were not considered as a valid correspondence in the reference alignment.

4 Conclusions & Future Work

In this paper we have introduced the smart cities domain and underline some communication and interaction problems that are highly likely to show up in this kind of development. We have also described the foundations of the *ontology matching* based approach that we propose to tackle such problems in the smart cities, and the ontology matching algorithm that we have defined to address this problem. We have described the measures of precision, recall and f-measure used to evaluate this type of algorithm, and the results obtained when doing so.

We have then looked at some alternatives to refine the alignments obtained by our algorithm and therefore improve such results. Among the different techniques we have focused on *alignment reuse* as it was particularly indicated for scenarios like ours. We have used previously existing alignments, *rings*, to identify paths of between the ontologies to match and therefore enhance the results obtained. We have tested this approach using the ontologies available from the *conference track* of the Ontology Alignment Evaluation Initiative 2013. And we have verified the viability and validity of the approach.

In spite of these good results that account for the viability of our approach, there are some issues that need to be addressed in order to obtain the best results possible and hence to improve the usability of the smart cities. There is, for instance, the need to test our proposal using ontologies taken from the real domain where it will be deployed, the smart cities. Additionally, as we introduced in section 3, there seems to be a direct relation between the rings chosen and the goodness of the results obtained, so we aim at focusing on determining the features that make a *ring* better than other. It is also necessary to explore techniques that will allow the alignment reuse in the real environment, so we are turning to some techniques such as *alignment storing and sharing* and *alignment annotation*.

References

[1]Katole, B., Sivapala, M., V, S.: Principle elements and framework of internet of things. International Journal Of Engineering And Science **3**(5) (07 2013)

[2]Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., Oliveira, A.: Smart cities and the future internet: Towards cooperation frameworks for open innovation. In Domingue, J., Galis, A., Gavras, A., Zahariadis, T., Lambert, D., Cleary, F., Daras, P., Krco, S., Müller, H., Li, M.S., Schaffers, H., Lotz, V., Alvarez, F., Stiller, B., Karnouskos, S., Avessta, S., Nilsson, M., eds.: Future Internet Assembly. Volume 6656 of Lecture Notes in Computer Science., Springer (2011) 431–446

[3]Chin, J.S.Y., Callaghan, V., Clarke, G., Hagras, H., Colley, M.: End-User Programming in Pervasive Computing Environments. In Yang, L.T., Ma, J., Takizawa, M., Shih, T.K., eds.: PSC, CSREA Press (2005) 187–192

[4]Euzenat, J., Shvaiko, P.: Ontology matching. 2nd edn. Springer-Verlag, Heidelberg (DE) (2013)

[5]Uceda-Sosa, R., Srivastava, B., Schloss, B.: Using Ontologies to make Smart Cities Smarter. IBM Research. (06 2012)

[6]:SOFIA2. http://scfront.cloudapp.net/ (5 2014)

[7]Wiesman, F., Roos, N., Vogt, P.: Automatic ontology mapping for agent communication. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2, ACM (2002) 563–564

[8]Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Software Engineering (2012)

[9]Gal, A., Shvaiko, P.: Advances in ontology matching. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **4891 LNCS** (2008) 176–198

[10]Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In: IIWeb. (2003) 73–78

[11]Duman, H., Hagras, H., Callaghan, V.: Intelligent association exploration and exploitation of fuzzy agents in ambient intelligent environments. Journal of Uncertain Systems **2**(2) (2008) 133–143

[12]Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. Very Large Data Base (VLDB) Journal **10**(4) (2001)

[13]Jiménez-Ruiz, E., Grau, B.C. In: Logmap: Logic-based and scalable ontology matching. Springer (2011) 273–288

[14]Hanif, M.S., Aono, M.: Anchor-flood: Results for oaei 2009. In Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A., eds.: OM. Volume 551 of CEUR Workshop Proceedings., CEUR-WS.org (2008)

[15]Noy, N.F., Musen, M.A.: Anchor-prompt: Using non-local context for semantic matching. In: Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA (2001)

[16]Le, B.T., Dieng-Kuntz, R., Gandon, F.: On Ontology Matching Problems (for building a corporate Semantic Web in a multi-communities organization). In: Proc. of the Sixth International Conference on Enterprise Information Systems - Porto - Portugal, Kluwer (14-17 April 2004)

[17]Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM **38**(11) (1995) 39–41

[18]World Wide Web Consortium. W3C: OWL: Web Ontology Language (2014)

[19]Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., dos Santos, C.T.: Ontology Alignment Evaluation Initiative: Six Years of Experience. Journal on Data Semantics - Springer (2011) 158 – 192

[20]Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10**(4) (2001) 334–350

[21]Do, H., Rahm, E.: COMA - a system for flexible combination of schema matching approaches. In: Proceedings of the 28th VLDB Conference, Hong Kong, China (2002)

[22]Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Proc. ACM SIGMOD Conference. (2005)

[23]Naphade, M.R., Banavar, G., Harrison, C., Paraszczak, J., Morris, R.: Smarter Cities and Their Innovation Challenges. IEEE Computer **44**(6) (2011) 32–39

[24]Uceda-Sosa, R., Srivastava, B., Schloss, R.J.: Building a Highly Consumable Semantic Model for Smarter Cities. In: Proceedings of the AI for an Intelligent Planet. AIIP '11, New York, NY, USA, ACM (2011) 3:1–3:8

[25]Villanueva, F.J., Santofimia, M.J., Villa, D., Barba, J., López, J.C.: Civitas: The Smart City Middleware, from Sensors to Big Data. In Barolli, L., You, I., Xhafa, F., Leu, F.Y., Chen, H.C., eds.: IMIS, IEEE (2013) 445–450

[26]Fazio, M., Paone, M., Puliafito, A., Villari, M.: Heterogeneous Sensors Become Homogeneous Things in Smart Cities. In You, I., Barolli, L., Gentile, A., Jeong, H.D.J., Ogiela, M.R., Xhafa, F., eds.: IMIS, IEEE (2012) 775–780