



UNIVERSITY  
OF TRENTO

---

**DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE**

---

38050 Povo – Trento (Italy), Via Sommarive 14  
<http://www.disi.unitn.it>

ONTOLOGY MATCHING EVALUATION USING GIS WEB  
SERVICES

Lorenzino Vaccari and Juan Pane

December 2008

Technical Report # DISI-08-084



# Ontology matching evaluation using GIS web services

Lorenzino Vaccari<sup>1</sup>, Juan Pane<sup>1</sup>

DISI, University of Trento, Italy  
{vaccari|pane@disi.unitn.it}

**Abstract.** Matching between concepts describing the meaning of representing for heterogeneous information sources is a key operation in many application domains, including web service coordination, data integration, peer-to-peer information sharing, query answering, and so on. In this paper we present an evaluation of an ontology matching approach, specifically of Structure-Preserving Semantic Matching (SPSM) solution which we applied within an emergency response scenario for GIS web service coordination. We evaluate the SPSM solution both on a set of synthesized benchmarks, as well as on real world GIS ESRI ArcWeb services. The results demonstrate the robustness and the performance of the SPSM approach on a large (ca. 700.000) number of matching tasks.

## 1 Introduction

With the advent of the (Semantic) Web an increasing and different number of web services, providing heterogeneous data and functionalities, became available from different sources. Nowadays, the challenge is to discover and chain these services in order to obtain a user defined goal. Of course, each service provider publishes its services based on its background knowledge, therefore, discovering and chaining services requires a semantic interoperability level between different providers. If we consider signatures (name, inputs and outputs) of web services to be tree-like structures, we can use some of-the-shelf ontology matching systems to facilitate the resolution of the semantic heterogeneity and interoperability problem.

In fact, ontology matching is often proposed as a solution to this heterogeneity problem in many applications, including integration of web services. Ontology matching takes two graph-like structures such as, for instance, lightweight ontologies [6] and produces an alignment (set of correspondences) between the nodes of those graphs that correspond semantically to one another. Many solutions to the ontology matching problem have been proposed so far (see [5,17,22] for recent surveys). In this paper we focus on a particular type of ontology matching, namely *Structure-Preserving Semantic Matching* (SPSM) and evaluate a concrete solution proposed in [7], on a set of ESRI ArcWeb services SOAP

methods<sup>1</sup>. We base the evaluation, both on a set of synthesized benchmarks, as well as between services from real world GIS ESRI ArcWeb services.

This paper is an updated version of an earlier conference paper [12], where the most important extension of this paper over the previous work in [12] includes an extensive evaluation of the SPSM matching approach on a set of real-world ESRI ArcWeb services.

The structure of the rest of the paper is as follows. We briefly discuss the related work in Section 2. Section 3 presents a motivating scenario. In Section 4 we outline a Structure-Preserving Semantic Matching approach used to reduce the semantic heterogeneity problem. We provide the evaluation of the discussed solution discussed in Sections 5. Finally, we present the major findings of the paper as well as future work in Section 6.

## 2 Related Work

A substantial amount of work that tackles the problem of semantic heterogeneity has been done in the semantic web, artificial intelligence and database domains, where ontology matching is viewed as a plausible solution, see, e.g., [16], [22] for recent surveys, while examples of individual approaches addressing the matching problem can be found on <http://www.OntologyMatching.org>. These solutions take advantage of the various properties of ontologies<sup>2</sup> (e.g., labels, structures) and use techniques from different fields (e.g., statistics and data analysis, machine learning, linguistics). These solutions share some techniques and attack similar problems, but differ in the way they combine and exploit their results. A detailed analysis of the different techniques in ontology matching has been given in [5].

The most similar approaches to the solution that we used in our scenario are the work presented in [1,19,23], where the services are assumed to be annotated with the concepts taken from various ontologies. The matching algorithms of those works combine the results of atomic matchers that roughly correspond to the element level matchers<sup>3</sup> exploited as part of the work in [10] and [7] which we applied in our scenario.

The ontology matching evaluation theme has been given a chapter account in [5]. There are several individual approaches to the evaluation of matching approaches in general, see, e.g., [9], as well as with web services in particular, see, e.g., [23,14,19]. OAEI [3,4] is a coordinated international initiative that organizes the evaluation of the increasing number of ontology matching systems. Unfortunately, at present, matching of web services has not been addressed by

<sup>1</sup> <http://www.arcwebservices.com/v2006/help/index.htm>

<sup>2</sup> The term ontology is used here in a wide sense, and, hence, encompasses sets of terms, classifications, database schemas, thesauri, etc.

<sup>3</sup> Element level matching techniques compute correspondences by analyzing concepts in isolation, ignoring their relationships with other concepts. In turn, structure level matching techniques compute correspondences by exploiting the results of element level matchers and by analyzing relationships between concepts.

OAEI. In turn, the Semantic Web Service (SWS) challenge initiative aims at evaluating various web service mediation approaches [20]. However, as also noticed in [21], the key problem with the current evaluations of web service matching approaches is the lack of real world web service data sets as well as their size, for example as from [20], the participants of SWS were operating with 20 web services.

The goal of this paper is the evaluation of the SPSM approach on real world GIS web services. Specifically, we conduct two kinds of experiments: *(i)* an evolution experiment with 1.600 matching tasks, where we deal with meaning and syntactic robustness tests, and *(ii)* a classification robustness experiment with 50 matching tasks, where we compare the results of an automatic and a manual classification of a selected set of GIS ESRI ArcWeb service operations.

### 3 Motivating scenario

Emergency management activities, that in the following we will reference as emergency response (eResponse) activities, are developed and implemented by means of the essential analysis of information and the coordination of the involved actors, including emergency personnel, army, volunteers, etc. The choice of the eResponse domain is driven by its potential to deal with a distributed-knowledge and dynamic environment. Disaster data and events can be acquired, modeled, fused and displayed in state-of-the-art Spatial Data Infrastructures (SDIs) [2,11,15,13,18] using distributed data sources, the majority of which are spatial.

Within the general SDI management scenario, we focus on the most commonly used specific use case, i.e., Map Request Service. The service requestor, both in an emergency or normal situation, needs to visualize a map of a region with geo-referenced information. Usually, the searched map is a composition of different geographic layers offered by many GIS service providers. Each layer can be obtained by many different service providers, and the failure of one service provider should not compromise the eResponse framework ability of analyzing data in order to predict and react to possible disasters. This scenario clearly states the necessity of finding and integrating several map service providers for redundancy.

We developed an initial prototype in which data from different web services in the network can be collected, visualized and analyzed. In this prototype, we simulate an emergency situation (the flooding of the city of Trento, Italy) in which we gather GIS data from different providers showing an integrated view of the situation as shown in Figure 1.

The GUI in Figure 1 represents a control panel used by the emergency coordinators. Through the emergency GUI the users can visualize the map and the events of the emergency situation. The map shows static geographic datasets (topographic map, probable flooding areas, escape roads, meeting points, refuge centers, sensor network) as well as dynamic datasets (e.g. Figure 1 shows the po-

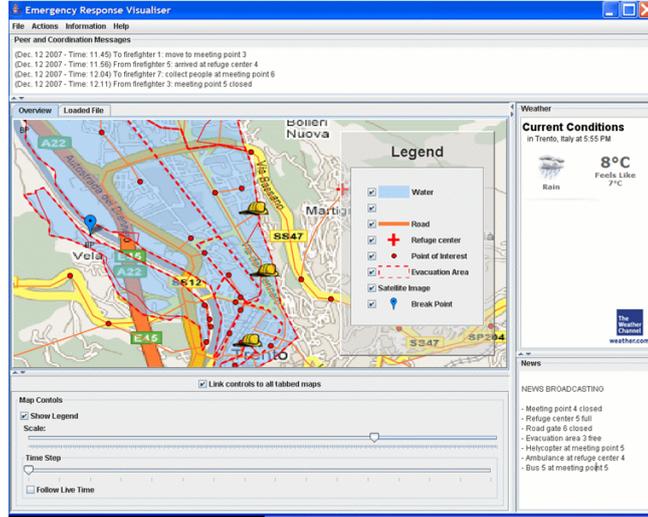


Fig. 1. Emergency response visualiser.

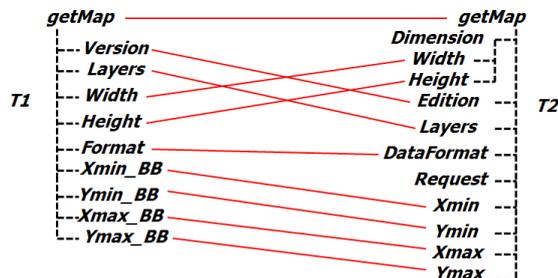
sition of the firefighters involved in the simulation). Moreover, through the GUI, emergency coordinators can perform actions (enact the emergency plan, recall digital services, change the map legend, search for other GIS datasets, send statements to the emergency peers, etc) as well as ask information about the emergency situation (i.e., evacuated people, list of the emergency peers, blocked roads, situation of the meeting points and of the refuge centers, etc).

## 4 Structure-Preserving Semantic Matching

In our scenario service providers are selected at run time and they can dynamically change, according to their availability. Let us suppose that we want to find a service implementation with the following signature:  $getMap(Version, Layers, Width, Height, Format, XMin\_BB, YMin\_BB, XMax\_BB, YMax\_BB)$  ( $T1$  in Figure 2). Therefore we match  $T1$  with the capabilities of a provider offering the service, such as:  $getMap(Dimension(Width, Height), Edition, Layers, DataFormat, Request, Xmin, Ymin, Xmax, Ymax)$  ( $T2$  in Figure 2).  $T1$  and  $T2$  can also be viewed as web service operations, which in turn, can be represented as tree-like structures.

As shown in Figure 2, the first description requires the second argument of  $getMap$  function ( $Layers$ ) to be matched to the fourth one ( $Layers$ ) of  $getMap$  function in the second description. The value of  $Version$  in the first description must be passed to the second web service operation as the  $Edition$  argument. Moreover,  $Request$  in  $T2$  has no corresponding term in  $T1$ .

The purpose of Structure-Preserving Semantic Matching (SPSM) is to reduce semantic heterogeneity in service descriptions. Specifically, a semantic similarity



**Fig. 2.** Two web service descriptions (trees) and correspondences (lines) between them.

measure (*TreeSim*) is used to estimate similarity between services under consideration. This scenario poses additional constraints on conventional ontology matching. In particular, we need to compute the correspondences holding among the full graph structures and preserve certain structural properties. SPSM operation takes two graph-like structures and produces a set of correspondences between those nodes of the graphs that correspond semantically to one another, (i) still preserving a set of structural properties of the graphs being matched, namely that functions are matched to functions and variables to variables; and (ii) only in the case if the graphs are globally similar to one another, e.g.,  $T1$  is 0.75 similar to  $T2$  according to some measure (see [7] for the in-depth discussion). The implementation of SPSM is based on (i) a formal theory of abstraction [8] and (ii) a tree edit-distance algorithm [24].

The rationale behind the definition of the experiment is to simulate how good SPSM would be in comparing two given web services, in order to assess their similarity, provided that there is a mechanism that can discover such services. This would mean that we have means of automatically replacing the actual service provides as needed (e.g. if one service becomes inactive).

## 5 The GIS web service evaluation

The SPSM approach assumes that the web services described in WSDL will be annotated with some kind of meaningful descriptions of: (i) what the operations are (e.g. *getMap*); (ii) what the inputs and outputs are: i.e. that arguments are labeled descriptively and not merely as “*input1*”, “*var1*” and so on. Any additional mark-up that is used to provide semantics for web services outside of the WSDL files can also be amenable to our techniques, provided, as is usually the case, that descriptions of inputs and outputs can be captured in a tree structure.

In our scenario, we base our test cases on ESRI ArcWeb WSDL operations and we compare labeled trees that correspond to their signatures. ArcWeb services is a rich and well documented set of Web services application program-

ming interfaces (APIs) for integrating mapping functionality and GIS content into browser, desktop, mobile, and server applications.

We conducted two different kinds of tests. The first one has been inspired by the work on systematic benchmarks of the Ontology Alignment Evaluation Initiative (OAEI) [4]<sup>4</sup>. In this experiment we matched original labeled trees to synthetically altered trees. Moreover, we compared the performance of the SPSM algorithm against the performance of a baseline algorithm, such as edit-distance<sup>5</sup>. In the second experiment we compared a manual classification of our GIS ArcWeb services dataset, the so-called “reference alignment”, to the unsupervised one discovered by SPSM. Finally, we evaluated efficiency and quality of the results of SPSM matching solution on these test cases.

For both the experiments, we used standard measures as precision, recall and F-measure to evaluate quality of the SPSM matching results[5].

### 5.1 Test case 1: evolution scenario.

Ontology and web service engineering practices suggest that often the underlying trees to be matched are derived or inspired from one another using different kind of operations to change the meaning and the syntax of the original tree [7]. Therefore, it is reasonable to compare a tree with another one (derived from the original tree). We evaluated SPSM following this technique, performing syntactical and semantical alterations to each node in  $T1$ , to produce  $T2$ , with a probability ranging in  $[0.1 \dots 0.9]$ .

The evaluation dataset was composed of trees that are alterations of several original trees. Initially, 80 “original” trees were extracted from the ESRI ArcWeb services collection. Then, 20 altered tree were automatically generated for each original tree. Pairs composed of the original tree and one varied tree, thereby resulting in 1600 matching tasks, were then used as input to the SPSM solution. The alteration operations were applied to node names (node names being composed of labels), and correspond to the following four alteration categories:

1. **Replace a node name with an unrelated node name** randomly selected from a generic dictionary (Brown corpus<sup>6</sup> for this implementation).
2. **Add or remove a label in a node name.** A label was dropped only if the node name contained more than one label. Label addition in node names was performed by using words extracted from the Brown corpus.
3. **Alter syntactically a label:** this test aimed at mimicking potential misspellings of the node labels. We randomly selected a set of labels to be modified and, for each word, we randomly decided how to modify it by using three types of alterations: character dropped, added, or changed.

<sup>4</sup> <http://oaei.ontologymatching.org/2006>

<sup>5</sup> The edit-distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character

<sup>6</sup> <http://icame.uib.no/brown/bcm.html>

4. **Replace a label in a node name with a related one** (e.g. synonyms, hyponyms, hypernyms): this test aimed at simulating the selection of an operation which meaning was similar (equivalent, more general or less general) to the original one. In the implementation of these type of alterations we used a number of generic sources like WordNet 3.0 and Moby<sup>7</sup>.

We implemented evaluation tests to explore the robustness of the SPSM approach towards both typical syntactic alterations (i.e. replacements of node names, modification of node names and misspellings) and typical meaning alterations (i.e. usage of related synonyms, hyponyms, hypernyms) of node names.

Since the generated tree alterations were known, these provided the ground for computing the expected similarity score (*ExpScore*), and hence, the reference results were available by construction (see also [4]). This allowed for the computation of the matching quality measures.

Initially we set the value of  $Score_i$  for each node in  $T2$  to 1. Then each alteration operation, applied sequentially to each node, reduced  $Score_i$ . When all the alteration operations were applied, the *ExpScore* between two trees  $T1$  (the original one) and  $T2$  (the modified one) ranged between  $[0 \dots 1]$  and was computed as follows:

$$ExpScore(T1, T2) = \frac{\sum_{i \in N} Score_i}{N} \quad (1)$$

where  $N$  is the node number of  $T1$ ,  $T2$  and  $Score_i$  is the expected similarity score assigned to each node of  $T2$ .

In order to compute quality measures, we compared the set of correct correspondences  $R$  produced by SPSM with the set of correspondences  $S$  considered to be correct.  $R$  is the set of results from SPSM whose *TreeSim* value was greater than a cut-off threshold, and  $S$  is the set of altered trees  $T2$  whose *ExpScore* was higher than a fixed threshold (*corrThresh*). *corrThresh* separates the trees  $T2$  that a human user would, on average, consider as still similar to the original  $T1$  from those that are too different.

For each alteration operation, we computed precision, recall and F-measure using an empirically established threshold *corrThresh* = 0.6. In addition, we compared SPSM recall, precision and F-measure values with the ones obtained from a baseline matcher (edit-distance for our experiment).

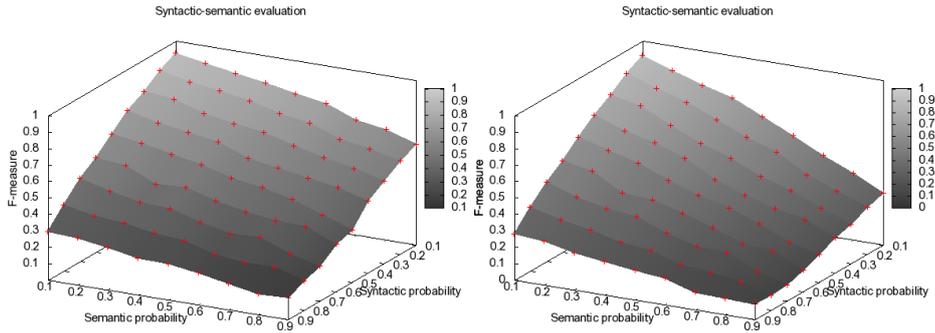
We repeated all experiments described above 10 times and the presented results correspond to the average values. The maximum value of standard deviation for the plotted average values was 0.013.

Comparison with a baseline matcher (Figure 3) shows how the SPSM approach is always comparable with the baseline when only syntactic alteration are considered, whereas SPSM results were always better (in average more than 20%) when “meaning” alterations were introduced. This is exactly what one would expect, since the SPSM approach includes a number of state-of-the-art

<sup>7</sup> <http://www.mobysaurus.com>

syntactic matchers (that are first used in the internal matching algorithm) plus a number of semantic matchers that enter into play for the alterations in the meaning of nodes labels [7].

Figure 3 shows the comparison between SPSM and the baseline when “*Replace a label in a node name with a related one*” and “*Alter syntactically a label*” alterations were combined together. The diagrams show the scores of F-measure (we selected a cut-off threshold of 0.7) for both SPSM and edit-distance (baseline) matchers.



**Fig. 3.** F-measure values for SPSM (on the left) and edit-distance (baseline, on the right) matchers.

## 5.2 Test case 2: classification scenario.

In this test case, we aimed at investigating the capability of the SPSM algorithm in the unsupervised clustering of a set of meaningfully related web service operations. The evaluation setup corresponds to a manual classification (reference alignment) of a selected set (50) of ArcWeb service operations. The construction of the set for this test case was independent from the set used in *Test case 1*. This construction of the reference alignment included the following steps:

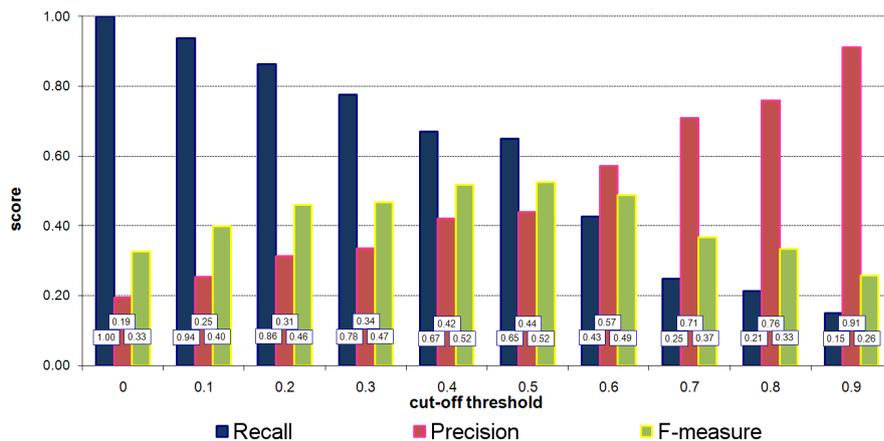
1. Classify manually the set of operation conforming to the WSDL file description of the operations.
2. Eliminate some general (valid for all the group) operations, e.g., *get\_Info(data\_Sources, token)*; these operations did not contribute to operation-specific information to the classification process.
3. Refine the classification by logically regrouping some operations, e.g., *find\_Place(place\_Name, place\_Finder\_Options, token)* was grouped with the “*address finder*” set of operations.

We thus compared the constructed manual classification of the selected Web Service operations with the automatically obtained by SPSM approach. Specifically, we:

- Compared each operation signature with all other one signatures.
- Computed a SPSM similarity measure between each signature and all the other signatures.
- Classified the operations depending on a given similarity threshold.

We built a  $n \times n$  matrix (where  $n$  was the number of the WSDL operations) that contained our “ground truth”, i.e, the manual classification of each pair of operations, which we considered to be correct. Then, we used the SPSM algorithm to independently classify the same operations in an automatic way. For each pair of operations, the SPSM algorithm returned a similarity measure that was compared with a cut-off threshold in the range  $[0 \dots 0.9]$ .

We computed recall, precision and F-measure comparing the set of the relevant (manual) classifications and the set of the retrieved (automatic) correspondences. Figure 4 shows, for each cut-off threshold, recall, precision and F-measure scores. Classification quality measures depend on the cut-off threshold values and the SPSM solution demonstrates good overall matching quality (i.e. F-measure) on the wide range of these values. In particular, the best F-measure values exceed 50% for the given GIS operations set. For example, for a *TreeSim* threshold of 0.5, precision is 0.44, recall is 0.65, and F-measure is 0.52.



**Fig. 4.** Recall, precision and F-measure values for classification robustness test.

In this experiment, we investigated how the proposed SPSM approach could be used in determining (in an unsupervised manner) the “class” of a specific GIS operation directly from the information present in its WSDL operation signature. Although, the results are encouraging, still 50% of GIS operation were incorrectly classified, due to the limited knowledge presented in the signatures only. In this case, the presence of more informative and semantically structured annotation would improve significantly the automatic classification at the expense obviously of a greater effort from the designer/programmer.

### 5.3 Performance evaluation.

The evaluation was performed on a standard laptop Intel Centrino Core Duo CPU-2Ghz, 2GB RAM, with the Windows Vista (32bit, SP1) operating system, and with no applications running but a single matching system. The efficiency of the SPSM solution is such that the average execution time per matching task in the evaluation under consideration was 43ms (the average number of the parameters of the WSDL operations was 4). The quantity of main memory used by SPSM during matching did not rise more than 2.3Mb higher than the standby level.

## 6 Conclusions and future work

The focus of this paper was the presentation and the extensive evaluation of a proposed *Structure-Preserving Semantic Matching* (SPSM) approach as a practical solution to the semantic heterogeneity problem between different implementations of required geo-services. This approach has been compared with other state-of-the-art solutions and discussed as a possible solution to the semantic heterogeneity problem in the field of coordination of Web services. We conducted an extensive set of empirically tests to evaluate quality and efficiency indicators of the SPSM approach. We based our extensive tests on a set of 80 real ArcWeb WSDL operations.

To provide an empirical evaluation of SPSM two different kind of tests were conducted. The first one was inspired by the work on systematic benchmarks of the Ontology Alignment Evaluation Initiative (OAEI) [4]. In this experiment we compared the results obtained with the SPSM approach to the results obtained with a baseline matcher; where F-measure for SPSM resulted 20% higher. In the second experiment we compared a manual classification (“reference alignment”) of our GIS ArcWeb services dataset, to the unsupervised classification computed by SPSM, obtaining encouraging results. Finally, we have evaluated performance and efficiency of SPSM matching approach on these test cases, obtaining small memory footprint and fast matching speed for the given tasks. This suggest that SPSM could be employed to find similar web services implementations at runtime.

Future work on the approximate SPSM proceeds at least in the following directions: *(i)* conducting extensive and comparative evaluation, including other kind of GIS web services like the ones available from OGC specifications and GRASS package, *(ii)* extending the matching approach for dealing with fully-fledged GIS ontologies like the ones provided by INSPIRE directive, *(iii)* incorporating domain specific preferences in order to drive approximation, thus allowing/prohibiting certain kinds of approximation (e.g not approximating vector maps with raster maps, although these are both maps), and *(iv)* use different kind of thesaurus like the multilingual GEMET or AGROVOC thesauri or to support multilingual matching.

**Acknowledgments.** This work has been supported by the FP6 OpenKnowledge European STREP (FP6-027253)<sup>8</sup>. This work has been developed in collaboration with various people and in particular with Maurizio Marchese and Pavel Shvaiko. We thank Paolo Besana for helping with the evaluation setup and David Dupplaw for the development of the emergency GUI. Moreover, we thank Fausto Giunchiglia, Fiona McNeill, and Mikalai Yatskevich for many fruitful discussions on the Structure-Preserving Semantic Matching.

## References

1. R. Aggarwal, K. Verma, J. Miller, and W. Milnor. Constraint driven web service composition in METEOR-S. In *Proceedings of the 1st IEEE International Conference of Services Computing (SCC)*, pages 23–30, 2004.
2. L. Bernard, M. Craglia, M. Gould, and W. Kuhn. Towards an SDI research agenda. In *Proceedings of the 11th European Commission-Geographic Information (EC-GI) and GIS Workshop*, pages 147–151, 2005.
3. C. Caracciolo, J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaisé, C. Meilicke, J. Pane, P. Shvaiko, H. Stuckenschmidt, O. Šváb Zamazal, and V. Svátek. Results of the ontology alignment evaluation initiative 2008. In *Proceedings of the International Workshop on Ontology Matching (OM) at the 7th International Semantic Web Conference (ISWC)*, 2008.
4. J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Šváb, V. Svátek, W. R. van Hage, and M. Yatskevich. Results of the ontology alignment evaluation initiative OAEI. In *Proceedings of the Workshop on Ontology Matching (OM) at the 6th International Semantic Web Conference (ISWC) + the 2nd Asian Semantic Web Conference (ASWC)*, pages 96–132, 2007.
5. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.
6. F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Encoding classifications into lightweight ontologies. *Journal of Data Semantics*, VIII:57–81, 2007.
7. F. Giunchiglia, F. McNeill, M. Yatskevich, J. Pane, P. Besana, and P. Shvaiko. Approximate structure-preserving semantic matching. In *Proceedings of the 7th Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pages 1234–1237, 2008.
8. F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial Intelligence*, 57(2-3):323–389, 1992.
9. F. Giunchiglia, M. Yatskevich, P. Avesani, and P. Shvaiko. A large scale dataset for the evaluation of ontology matching systems. *The Knowledge Engineering Review Journal*, 24(2), 2009, to appear.
10. F. Giunchiglia, M. Yatskevich, and P. Shvaiko. Semantic matching: Algorithms and implementation. *Journal on Data Semantics*, IX:1–38, 2007.
11. R. Groot and J. McLaughlin. *Geospatial Data Infrastructure: Concepts, Cases and Good Practice*. Oxford University Press, 2000.
12. M. Marchese, L. Vaccari, P. Shvaiko, and J. Pane. An application of approximate ontology matching in eresponse. In *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, pages 294–304, 2008.

---

<sup>8</sup> [www.openk.org](http://www.openk.org)

13. I. Masser. *Creating Spatial Data Infrastructures*. ESRI Press - RedLands - California, 2005.
14. R. Mikhaiel and E. Stroulia. Examining usage protocols for service discovery. In *Proceedings of the 4th International Conference on Service Oriented Computing (ICSOC)*, pages 496–502, 2006.
15. D. Nebert. *Developing Spatial Data Infrastructures. The SDI CookBook*. Global Spatial Data Infrastructure (GSDI), 2004.
16. N. F. Noy. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33(4):65–70, 2004.
17. N. F. Noy, A. Doan, and A. Y. Halevy. Semantic integration. *AI Magazine*, 26(1):7–9, 2007.
18. H. Onsrud. *Research and Theory in Advancing Creating Spatial Data Infrastructure Concepts*. ESRI Press - RedLands - California, 2007.
19. S. Oundhakar, K. Verma, K. Sivashanmugam, A. Sheth, and J. Miller. Discovery of web services in a multi-ontology and federated registry environment. *International Journal of Web Services Research*, 2(3):1–32, 2005.
20. C. Petrie, T. Margaria, U. Kuster, H. Lausen, and M. Zaremba. Sws challenge: Status, perspectives, and lessons learned so far. In *Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS)*, pages 447–452, 2007.
21. S. Schulte, J. Eckert, N. Repp, and R. Steinmetz. An approach to evaluate and enhance the retrieval of semantic web services. In *Proceedings of the 5th International Conference on Service Systems and Service Management (ICSSSM)*, pages 237–243, 2008.
22. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146–171, 2005.
23. E. Stroulia and Y. Wang. Structural and semantic matching for assessing web-service similarity. *International Journal of Cooperative Information System*, 14(4):407–438, 2005.
24. G. Valiente. *Algorithms on Trees and Graphs*. Springer, 2002.