VOAR: A Visual and Integrated Ontology Alignment Environment

Bernardo Severo[1], **Cassia Trojahn**[2], **Renata Vieira**[1]

[1]Pontifícia Universidade Católica do Rio Grande do Sul (Brazil)
[2]Université Toulouse 2 & IRIT (France)
bernardo.severo@acad.pucrs.br, cassia.trojahn@irit.fr, renata.vieira@pucrs.br

Abstract

Ontology alignment is a key process for enabling interoperability between ontology-based systems in the Linked Open Data age. From two input ontologies, this process generates an alignment (set of correspondences) between them. In this paper we present VOAR, a new web-based environment for ontology alignment visualization and manipulation. Within this graphical environment, users can manually create/edit correspondences and apply a set of operations on alignments (filtering, merge, difference, etc.). VOAR allows invoking external ontology matching systems that implement a specific alignment interface, so that the generated alignments can be manipulated within the environment. Evaluating multiple alignments together against a reference one can also be carried out, using classical evaluation metrics (precision, recall and f-measure). The status of each correspondence with respect to its presence or absence in reference alignment is visually represented. Overall, the main new aspect of VOAR is the visualization and manipulation of alignments at schema level, in an integrated, visual and web-based environment.

Keywords: Ontology Alignment, Ontology Matching, Ontology Visualization

1. Introduction

The Linked Open Data and Semantic Web efforts have been promoting the publishing of interlinked collections of data, in which the vocabulary is defined with the help of ontologies. In order to allow the interoperability between these different data sources, one has to be able to find semantically related entities in these sources. The task of finding semantically related entities (concepts, properties, instances) between two ontologies is known as ontology matching (Euzenat and Shvaiko, 2007). Ontology matching is a key challenge and there are many academic efforts proposing different ways to perform this task, which are evaluated in the context of evaluation campaigns, like the ones organized by the Ontology Alignment Evaluation Initiative (OAEI)[1].

Even with a variety of matching systems and a strong community working in this problem, the area still lacks web-based, visual and integrated environments in which ontology alignments can be easily manipulated and compared. Different matching systems propose ways for visualizing their generated alignments, like YAM++ (Ngo and Bellahsene, 2012a), HOMER (Udrea et al., 2007) and AgreementMaker (Cruz et al., 2009). Others environments provide an interface for manipulating alignments through a set of operations, like the Alignment Server (David et al., 2011), while few systems make available web-based interfaces for invoking their matching algorithms, like LogMap (Jiménez-Ruiz and Grau, 2011). Another resource largely used in the ontology matching community is the Alignment API (David et al., 2011), that provides the basis for developing matching systems and means for alignment manipulation and evaluation that can be reused by other applications. In this paper we present a comparison of such tools/environments and VOAR.

The main motivation behind VOAR (**V**isual **O**ntology **A**lignment Envi**R**onment)[2] is to assist end users in the series of tasks for alignment manipulation, evaluation and visualization as a single integrated resource. VOAR provides an environment in which alignments can be shown and manipulated in different ways (combining with other alignments, editing correspondences, graphically displaying ontology entities and correspondences between them), as well as evaluated (graphically displaying evaluation outputs) and generated (using external matching systems). We believe that VOAR offers an alternative visual environment to the community interested in ontologies and ontology alignment.

The remainder of this paper is organized as follows. In 2. we provide the basic definitions of ontology alignment. Then, in §3. we detail the architecture of VOAR, describing the tool and its modules. The main related work are discussed in §4. where we present a comparative overview of the different matching tools and environments and VOAR. Finally, we conclude the paper and present future work §5..

2. Background

The value of ontologies for a range of applications has long been recognized. In Computer Science, the most common definition of an ontology comes from Gruber in (Gruber, 1995) “an explicit, formal specification of a shared conceptualization of a domain of interest”. Following this definition, an ontology should be machine-readable, agreed (shared) by a community or group, and restricted to a specific domain of interest. Ontologies provide a model of the concepts of a domain and how these concepts are related to each other. While these different ontologies may be similar, they may differ in granularity or detail, use different representations, or model the concepts, properties and axioms in different ways.

[1]http://oaei.ontologymatching.org/

[2]http://voar.inf.pucrs.br/

The process of finding correspondences between ontology entities is known as ontology matching (Euzenat and Shvaiko, 2007). It takes as input two ontologies o and o' and an (possibly empty) alignment A to be completed, and determines as output an alignment A', i.e., a set of correspondence. A simple correspondence can be defined as follows:

[Correspondence (Euzenat and Shvaiko, 2007)] A correspondence can be defined as $<e,e',r,n>$, such that: e and e' are entities (e.g., elements, properties, classes, instances) of o and o′, respectively; r is a relation holding between two entities e and e', (for instance, *equivalence* ($\equiv$), *more general* ($\sqsupseteq$), *disjointness* ($\perp$), *overlapping* ($\sqcap$)); and n is a confidence measure number in the [0;1] range. The confidence assigns a degree of trust on the correspondence from the matcher.

Different matching approaches have emerged from the literature. A review on these approaches can be found in (Rahm and Bernstein, 2001; Kalfoglou and Schorlemmer, 2003; Euzenat and Shvaiko, 2013). The main distinction between each is due to the type of knowledge encoded within each ontology, and the way it is utilized when identifying correspondences between features or structures within the ontologies. While *terminological* methods lexically compare strings (tokens or n-grams) used in naming entities (or in the labels and comments concerning entities), *semantic* methods utilize model-theoretic semantics to determine whether or not a correspondence exists between two entities. Approaches may consider the *internal* ontological structure, such as the range of their properties (attributes and relations), their cardinality, and the transitivity and/or symmetry of their properties, or alternatively the *external* ontological structure, such as the position of the two entities within the ontological hierarchy. The instances (or extensions) of classes could also be compared using *extension*-based approaches. In addition, ontology matching systems rely not on a single approach.

In order to help users evaluating the suitability of proposed matching approaches to their needs and help developers of matching systems to improve their systems, evaluation of the generated alignments is an important task (Euzenat et al., 2011). With this aim, OAEI organizes ontology matching evaluation campaigns yearly since 2004 (Euzenat et al., 2011). The most common way for evaluating matching systems is to compare their generated alignments (A') with a reference alignment R, usually manually created by a domain expert. Typically, classical measures as precision (P), recall (Re) and f-measure (F) are used to compare alignments:

$$P = \frac{|R \cap A'|}{|A'|} \tag{1}$$

$$Re = \frac{|R \cap A'|}{|R|} \tag{2}$$

$$F = \frac{2 * P * Re}{P + Re} \tag{3}$$

Ontology alignments are important resources for the exploitation of ontologies and visual tools are important for the tasks of alignment manipulation and visualization, therefore we developed VOAR, described in the next section.

3. VOAR: Visual Ontology Alignment Environment

VOAR was built based on the PLATAL tool (Severo et al., 2013), originally designed to support the tasks of extraction of lightweight ontologies from web pages, alignment of these ontologies using basic matching strategies, and evaluation against reference alignments (which could be manually created). This improved version of PLATAL integrates a web-based visual environment and improves some of its functionalities, making it more manipulation driven. VOAR is an environment organized in five modular operation modes, working independent of each other, namely: (a) Correspondences Edition Module, (b) Alignment Manipulation Module, (c) Matching Module, (d) Evaluation Module and (e) Visualization Module.

In order to promote interoperability and reuse, VOAR is developed on the top of established technologies in the field of ontology matching, such as the Alignment API (version 4.5) and OWL API (Horridge and Bechhofer, 2011). The OWL API is a Java library that offers a reference implementation to work with OWL ontologies, mainly OWL parsing, while the Alignment API provides a set of functions that are used as basis in the Alignment Manipulation and Evaluation modules. Furthermore, the Alignment API format[3], the most common alignment representation format adopted in the ontology matching community, is used to export and import alignments in VOAR. With respect to the development of the web-based interface, VOAR was built with JavaServer Faces (JSF) technology, compatible with most current browsers.

For the ontology matching task itself, VOAR does not implement any specific matching approach, once it does not pretend to be a matching system. Instead, VOAR offers the possibility of invoking external matching systems. In order to do so, we chose to use a well-known interface that has been proposed in the context of the SEALS project[4]. This interface has been adopted since 2011 in the OAEI campaigns, where for some tracks, participating systems have to implement it in order to be automatically invoked with the evaluation platform[5]. Therefore, VOAR is able to invoke any matching system implementing the SEALS interface.

Furthermore, VOAR works with the concept of a 'working alignment', an in-memory alignment that can be built (creating a new/blank, uploading an existing one in the Alignment API format, generating one from running matchers, etc.) and changed along the use of each module (editing correspondences, performing set operations with other alignments, etc.). The working alignment is what the user will visualize along each module. Users can as well export this generated/modified alignment for later use.

In the following, each VOAR module is presented.

[3]http://alignapi.gforge.inria.fr/format.html

[4]http://www.seals-project.eu/

[5]http://oaei.ontologymatching.org/2011.5/seals-eval.html

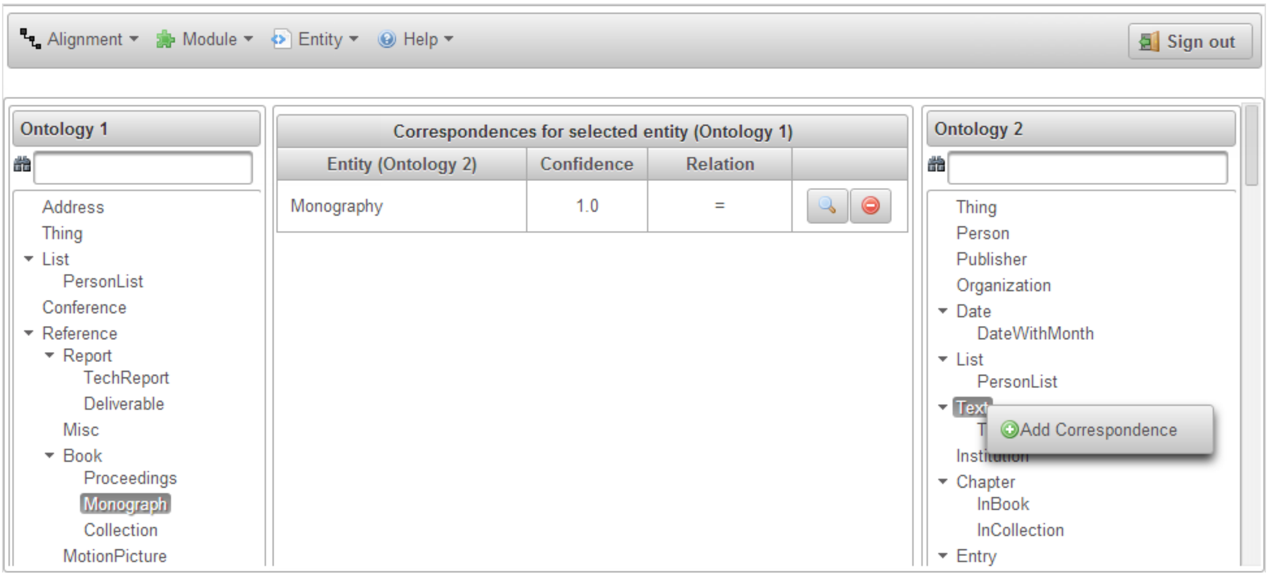

Figure 1: VOAR correspondences edition module.

3.1. Correspondences Edition

Following the definition of alignment in §2., VOAR was designed to allow users to load two ontologies o and o' and an alignment A', which can be edited. Figure 1 presents a screenshot of the VOAR correspondences edition module, where trees representing each ontology are shown, and between them, a table of correspondences of the working alignment for a given ontology entity. The 'Modules' option in the top of the Figure 1 allows for switch between the modules. The ontologies used in the examples come from the test 304 of the OAEI 2012 benchmark[6].

Correspondences in the alignment can be visualized, for a selected entity (on the tree representation), then its relation and confidence modified (in the table itself) or deleted. One can use the focus operation in each correspondence line of the table to jump to the entities in the tree visualization. A new correspondence may be created by selecting entities in both source and target ontologies, and it is displayed along with the existing ones. Finally, in each tree visualization, users can also filter out the entities whose naming (URI, label, comments) corresponds to a search criteria, what is especially useful when dealing with large ontologies. In this current version, VOAR is limited to the visualization and manipulation of correspondences involving ontology classes.

3.2. Alignment Manipulation

VOAR also provides ways to manipulate an alignment as an object, mostly by interacting with other alignments. The Figure 2 shows a screenshot of this module. The current set of available operations are: *union* (resulting in the merge of alignments), *intersection* (where only correspondences occurring in all input alignments are kept), *difference* (complement, where a sub-set of correspondences are removed from a given alignment). There are also other auxiliary operations, like the possibility to *trim* correspondences under a given threshold and *invert* the ontologies order. These operations are available in the Alignment API and Alignment Server for a pair of alignments, in their native versions, and we reuse that implementation through the Alignment API. In VOAR users can select multiple alignments at once and apply these operations (for instance, making the union of multiple alignments together), by using the API as base and composing its results. All resulting changes are incorporated to the working alignment.

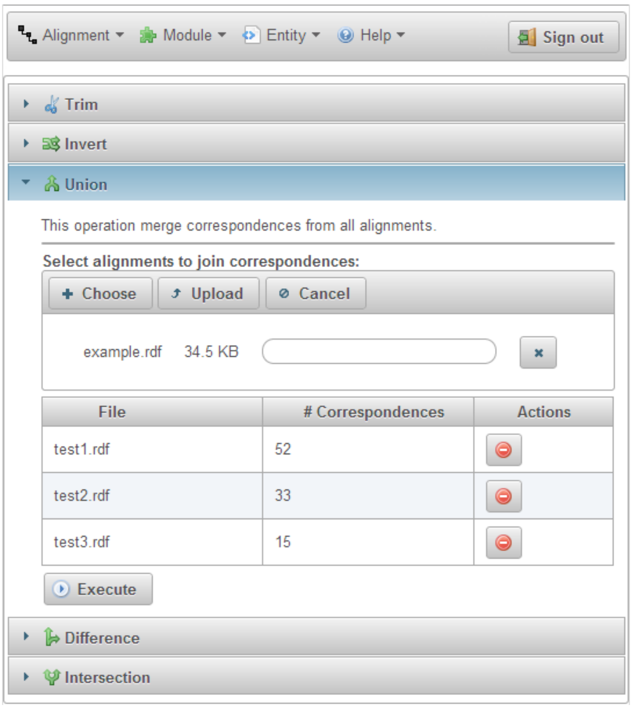

Figure 2: VOAR alignment manipulation module.

3.3. Ontology Matching

As stated in §3., VOAR does not pretend to be a matching system. Instead, VOAR allows for invoking external matching systems. In order to do so, these systems have

[6] http://oaei.ontologymatching.org/2012/benchmarks

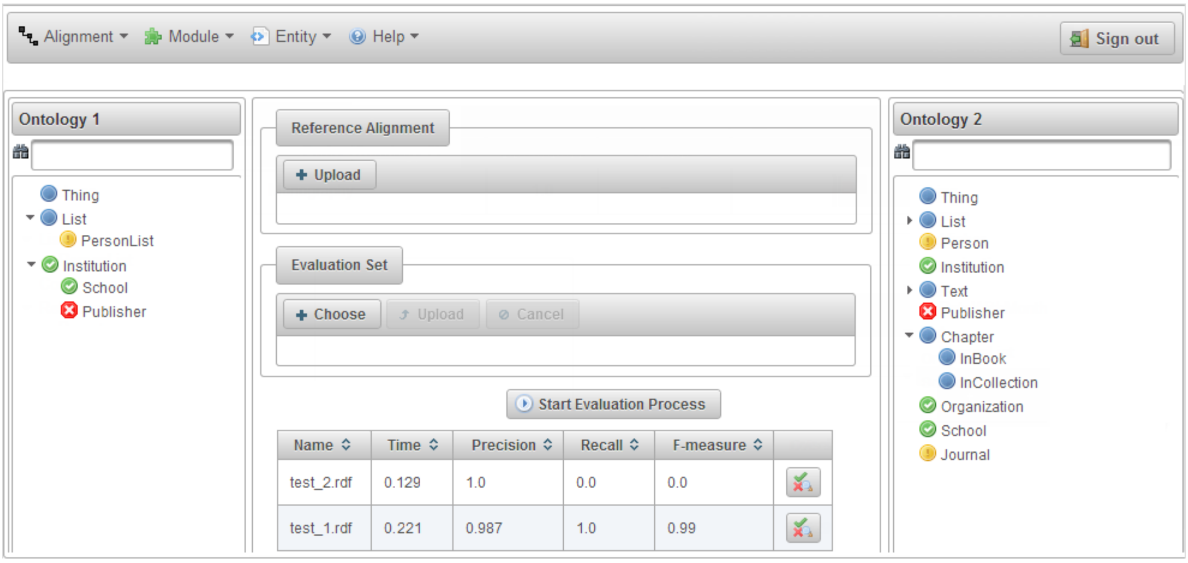

Figure 3: VOAR evaluation module.

Table 1: Correspondence evaluation statuses

Icon	Status	Description
	Incorrect correspondences	The proposed alignment contains incorrect correspondences for this entity when compared to the reference alignment (true negative)
	Missing correspondences	The proposed alignment is missing correspondences for this entity when compared to the reference alignment (false negative)
	Correct correspondences	The proposed alignment contains all possible correspondences for this entity when compared to the reference alignment (true positive)
	No correspondences	Neither the proposed alignment nor the reference alignment contains correspondences for this entity (true negative)

to implement the well-known SEALS Ontology Matching Interface. We choose to follow this interface because it has been largely adopted in the context of the OAEI evaluation campaigns. Therefore, using this common interface promotes the interoperability in the ontology community.

Basically, the SEALS format requires to bundle the matcher in a certain folder structure, and create a “bridge” implementing the SEALS Ontology Matching Interface. As most state-of-the-art matching systems participate in OAEI campaigns and have to implement the SEALS interface in order to participate in some tracks, using this interface allows users to upload most matchers (publicly) available from the OAEI campaigns (like YAM++, LogMap, etc.). One can also follow this format for generating a bundle from their own matghing system in order to run it in VOAR. In VOAR, one can upload a matcher bundle (available only for the current session), what facilitates the user task when running matchers, once some matchers usually have to be run in a console prompt.

Users are able to select which available matchers will be run for the uploaded ontologies (those that belong to the working alignment). Each selected matcher will run one after the other, sequentially. If any error occurs during execution, VOAR displays an appropriate error message and starts the execution of the the next matcher (no interaction is needed). When the executions are finished, the resulting alignments can be loaded as working alignment, once they are temporarily stored during the current session and can be edited, evaluated and visualized. VOAR is not yet able to manage systems requiring long execution times along with other modules execution, and the user has to wait until the alignment processes are completed. Hence, the alignment resulting from the execution of each matcher can be then loaded and downloaded (for other uses or later loading in VOAR).

3.4. Alignment Evaluation

With VOAR, one can compare multiples alignments (working alignment and/or others that can be uploaded) with respect to a reference alignment. Measures of precision, recall, and F-measure can be visualized for each evaluated alignment. We have used the implementations available in the Alignment API for generating these evaluation metrics. VOAR allows the visualization of the results according to each correspondence. Figure 3 shows a screenshot of VOAR where evaluation results are presented (table of results for all submitted alignments and tree representations with status by entity for a selected alignment).

The status of specific entities with regard to the reference alignment is shown on each ontology tree representation, according to four possible status (represented by icons on each entity), as presented in Table 1. These icons are shown respecting the order of precedence of values in Table 1. For example, if a given ontology entity has one or more incor-

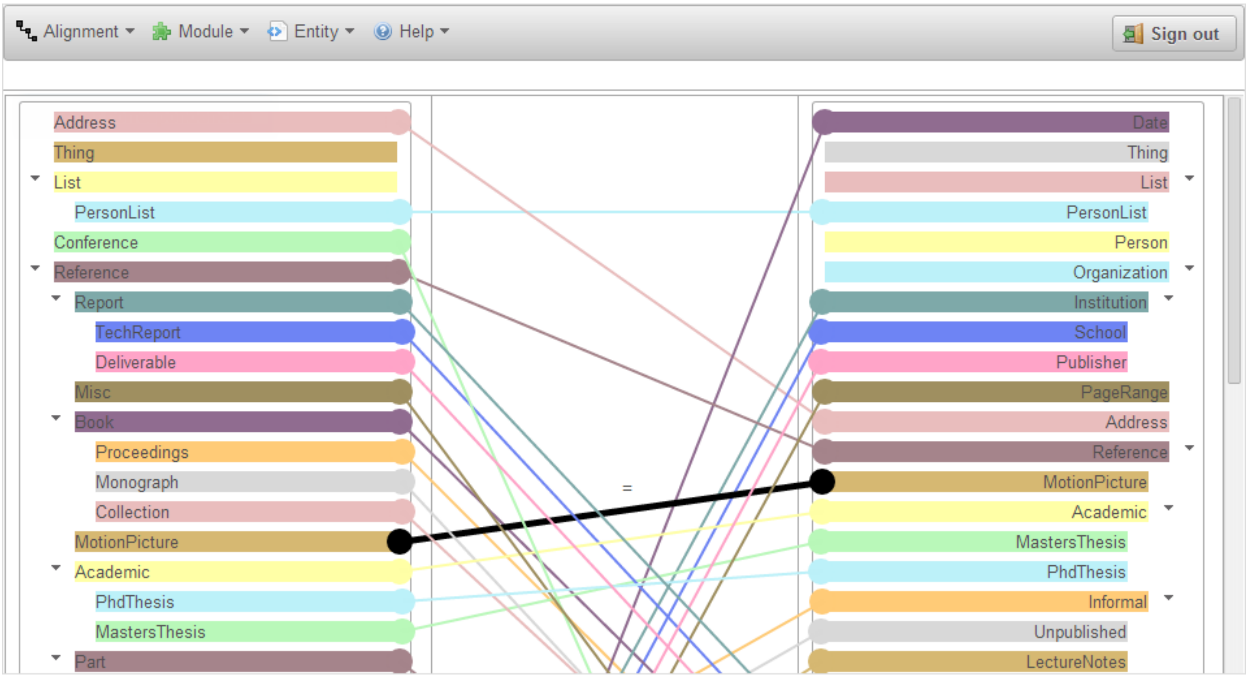

Figure 4: VOAR visualization module.

rect correspondences, it will be shown the associated icon of incorrect correspondences, even if there are other correspondences correct or missing for the same entity.

3.5. Alignment Visualization

An important feature for a visual alignment tool is the visualization of an alignment in a graphical form, aiming at facilitating the interpretation and analysis of the generated correspondences. A global view, as the one available in VOAR, provides assistance in the interpretation of the alignment as a whole, showing aspects such as density. Areas of alignments concentration or lack of alignments can also be observed.

Another important aspect of the representation is related to the colors used to each ontology entity and correspondences. When a great number of correspondences are displayed on screen, is hard to see which entities and connections are the ones one want to focus on. So the visualization strategy in this module assigns a common color for entities with a correspondence, as well as their connection line (Figure 4), making easier to track the links.

During the visualization of the alignment, one can set the focus on a given correspondence to make it being shown highlight with their attributes. Selecting entities in the ontologies trees allows the user to navigate to the application module that provides ways to manipulate that specific data (also accessible in the 'Modules' option in the top of the Figure 4).

4. Related Work

Different tools have been proposed for assisting the user in the task of matching ontologies, in which we found similar operations and visual representations as the ones provided by VOAR. COMA++ (Aumueller et al., 2005) was designed for aligning XML schemas and offers a library of matchers. Users can define their own correspondences and provide feedback on automatically generated ones. Visualization of alignments is done by lines drawn between the trees of entities. HOMER (Udrea et al., 2007) makes available a radial-graph display GUI for alignment visualization that allows the user to navigate to any match decisions and to compare multiple alignments in parallel (i.e., displaying common/differences between the alignments). PROMPT (Noy and Musen, 2000) is a Protégé plugin for ontology merging and alignment that makes suggestions and, for each of its suggestions, presents a series of explanations. A series of other Protégé plug-ins are also available for similar tasks (Lanzenberger and Sampson, 2006) (Kolli and Doshi, 2008).

The YAM++ (Ngo and Bellahsene, 2012a)(Ngo and Bellahsene, 2012b) matcher, which discovers correspondences between two ontologies by using machine learning approach, offers a graphical way for visualizing the generated alignments just like COMA++. Other examples of matching systems implementing some GUI mechanism are AgreementMaker (Cruz et al., 2009), that presents information in a tree diagram, provides more information than others tools at the level of the correspondence (colors for each connected entities and confidence of the relation) and allows to focus on individuals; and TaxoMap (Hamdi et al., 2010)[7], where the interface is oriented to the system parameterization. Taxomap does not show the ontologies or its alignment in a graphical form. LogMap[8], a matching systems based on applying reasoning and repair capabilities, provides a simple web interface that allows users to browse the input ontologies and invoke the LogMap system. An environment for manipulating alignments is OLA (OWL Lite Alignment) (Euzenat et al., 2004). It supports

[7]https://www.lri.fr/ hamdi/TaxoMap/TaxoMap.html
[8]http://csu6325.cs.ox.ac.uk/

Table 2: Alignment tools comparison

Tool	Environment	Correspondences Edition	Alignment Manipulation	External Matching	Evaluation	Alignment Visualization
PROMPT	Plugin	-	-	-	-	yes
YAM++	Standalone	yes	-	-	yes	yes
COMA++	Standalone	yes	-	-	yes	yes
HOMER	Standalone	-	-	-	yes	yes
AgreementMaker	Standalone	yes	-	-	yes	yes
TaxoMap	Standalone	-	-	-	-	-
OLA	Standalone	-	-	-	-	yes
SILK workbench	Web	yes	-	-	-	yes
Alignment Server	Web	-	yes	-	yes	yes
LogMap	Web	-	-	-	-	-
VOAR	Web	yes	yes	yes	yes	yes

parsing and visualization of ontologies, automated generation of correspondences between entities, based on a set of predefined matching strategies, manual construction of alignments, and visualization of alignments. A graph-based approach is used to represent ontologies in order to make the comparisons between them and their alignments easier. This structure makes the links between classes more explicit, through the links established from their object properties, while the alignment visualization is made by loading the graph structures for both ontologies. Another similar system is the SILK workbench (Volz et al., 2009), focused on the linked data, that provides means to interactively create configuration files enabling to manage different sets of data sources and linking tasks, allowing to visualize the links found and to correct them.

The Alignment Server[9], available with the Alignment API, provides a set of alignment operations (alignment using basic matching strategies, alignment manipulations and alignment evaluation) in a web-based interface, with the advantage of storing and retrieving alignments, enabling users to search for already generated alignments for a given pair of ontologies. However, the visualizations is raw text based and users cannot edit particular correspondences in the system. Furthermore, the manipulations of alignments are limited to pairs of alignments.

In Table 2 a comparison of the environment type and the operations available for each related tool is presented. The first point to be observed is the number of tools that are standalone (usually Java-based programs), while VOAR is a web-based application (not a plugin or a standalone application), like the Alignment Server, SILK and LogMap. This is an important aspect in the development of applications nowadays, improving compatibility with different OS, browsers, devices and making easier their access and management of versions (users dealing with tools upgrades).

For most of the evaluated tools with the functionality of editing the correspondences manually, the available actions are similar for most tools and environments: adding/removing and changing the relation type or confidence. The same scenario was found for the capability of alignment manipulation, where the only two tools with this functionality (Alignment Server and VOAR) present the available operations of the Alignment API.

[9]http://alignapi.gforge.inria.fr/aserv.html

Analyzing the matching functionality, all tools provide a way for running matchers. Some in particular allow to run existing matchers (Alignment Server – which invokes basic matchers, VOAR – which invokes systems implementing the SEALS Ontology Matching Interface), instead of implementing specific algorithms itself. The other tools usually provide customizations to their methods through their interface, like setting up thresholds and other parameterization of matchers (since their purpose is to be more matching-driven). With respect to evaluation, the most common evaluation, for those with this option, is basically the calculation of precision, recall and f-measure. A differential aspect is that some tools allow the user to manually set for each correspondence its correct status (true positive, true negative, etc.), like in YAM++ and AgreementMaker.

Regarding those tools and environments implementing visualization strategies, while matchers like COMA++ and YAM++ draw lines between the two ontology hierarchies, sometimes making difficult to distinguish relationships for large amount of alignments, VOAR proposes some simple but remarkable improvements in this type of presentation. Once a correspondence is selected, their properties and visualization gain a different appearance, making easier to evaluate them, like displaying the relation and the confidence value (similar to what is done in AgreementMaker). VOAR also determines a color, that is common for the entities on both ontologies which share a correspondence, making their correspondences more visual-driven, even if more than one line is connected to an entity. Other tools present visualizations with a focus on the evaluation of specific attributes (like in OLA), what is not yet done in VOAR, or simple interfaces for helping the user in the configuration and execution of matchers (mostly without graphical representation of alignments), like Alignment Server, LogMap and SILK. To the best of our knowledge, VOAR is the one of the first web-based integrated environment that includes a visual (graphical) representation of ontologies and their alignments, ways for manipulating multiple alignments and invoking external matching systems.

5. Conclusions and Future Work

This paper has presented the first release (1.0.0) of VOAR, a web-based and visual environment built to assist users in the main tasks involving ontology alignments. VOAR provides an integrated environment for manipulating, matching, evaluating, and visualizing alignments at schema level. Hence, making VOAR publicly available as a community resource offers an alternative way for manipulating alignments, what can be especially useful for end users.

As future work, many directions to improving and extending VOAR are envisaged. First of all, this initial release of VOAR will be submitted to evaluation by experts and non-experts of the ontology matching area. We plan also to add the possibility of visualizing and manipulating ontology alignments at the instance-level (a must in the context of Linked Open Data) – as well as visualizing data and object properties of ontologies. Third, we intend to provide a way for parameterizing matchers through the interface (in the current version, matchers run with their default configuration). Fourth, we plan to allow user to persist their data and configuration (storing ontologies, their alignments and matcher configurations). Furthermore, dealing with large ontologies and large sets of data instances may require providing alternative ways for visualizing the alignments between then. Finally, we intend to provide a way for collaborative ontology matching where multiple users can work collaboratively for generating and manipulating alignments.

6. References

Aumueller, David, Do, Hong-Hai, Massmann, Sabine, and Rahm, Erhard. (2005). Schema and ontology matching with coma++. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pages 906–908.

Cruz, Isabel F., Antonelli, Flavio Palandri, and Stroe, Cosmin. (2009). Agreementmaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589.

David, Jérôme, Euzenat, Jérôme, Scharffe, François, and dos Santos, Cássia Trojahn. (2011). The Alignment API 4.0. *Semantic Web*, 2(1):3–10.

Euzenat, Jérôme and Shvaiko, Pavel. (2007). *Ontology matching*. Springer, Heidelberg (DE).

Euzenat, Jérôme and Shvaiko, Pavel. (2013). *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition.

Euzenat, Jérôme, Loup, David, Touzani, Mohamed, and Valtchev, Petko. (2004). Ontology alignment with OLA. In *Proceedings of the 3rd EON Workshop at 3rd International Semantic Web Conference*, pages 59–68.

Euzenat, Jérôme, Meilicke, Christian, Stuckenschmidt, Heiner, Shvaiko, Pavel, and dos Santos, Cássia Trojahn. (2011). Ontology alignment evaluation initiative: Six years of experience. *Journal of Data Semantics*, 15:158–192.

Gruber, Thomas R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6).

Hamdi, Fayçal, Safar, Brigitte, Niraula, Nobal B., and Reynaud, Chantal. (2010). TaxoMap alignment and refinement modules: results for OAEI 2010. In *Proceedings of the 5th International Workshop on Ontology Matching, Shanghai, China, November 7*.

Horridge, Matthew and Bechhofer, Sean. (2011). The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2(1).

Jiménez-Ruiz, Ernesto and Grau, Bernardo Cuenca. (2011). Logmap: Logic-based and scalable ontology matching. In *Proceedings of the 10th International Semantic Web Conference, Bonn, Germany*, pages 273–288.

Kalfoglou, Yannis and Schorlemmer, Marco. (2003). Ontology mapping: the state of the art. *Knowledge Engineering Review*, 18(1):1–31.

Kolli, Ravikanth and Doshi, Prashant. (2008). Optima: Tool for ontology alignment with application to semantic reconciliation of sensor metadata for publication in sensormap. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 484–485.

Lanzenberger, Monika and Sampson, Jennifer. (2006). Alviz - a tool for visual ontology alignment. In *Proceedings of the 10th Conference On Information Visualization*, pages 430–440.

Ngo, Duy Hoa and Bellahsene, Zohra. (2012a). YAM++ : (not) Yet Another Matcher for Ontology Matching Task. In *28e Journées Bases de Données Avancées, Clermont-Ferrand, France*.

Ngo, DuyHoa and Bellahsene, Zohra. (2012b). YAM++ : A multi-strategy based approach for ontology matching task. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management, Galway City, Ireland*, pages 421–425.

Noy, Natalya Fridman and Musen, Mark A. (2000). Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455.

Rahm, Erhard and Bernstein, Philip A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350.

Severo, Bernardo, Trojahn, Cassia, and Vieira, Renata. (2013). PLATAL - a tool for web hierarchies extraction and alignment. In *Proceedings of the 8th International Workshop on Ontology Matching, Sydney, Australia, October 21*.

Udrea, Octavian, Miller, Renee, and Getoor, Lise. (2007). Homer: Ontology visualization and analysis. In *Demo Session at the 6th International Semantic Web Conference, Busan, Korea*.

Volz, Julius, Bizer, Christian, Gaedke, Martin, and Kobilarov, Georgi. (2009). Silk - a link discovery framework for the web of data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web, Madrid, Spain, April 20*.