Ontology Alignment through Argumentation

Nuno Luz and **Nuno Silva** and **Paulo Maio**
GECAD - Knowledge Engineering and Decision Support Group
School of Engineering - Polytechnic of Porto
R. Dr. António Bernardino de Almeida, 431
4200-072 Porto, Portugal

Paulo Novais
CCTC - Computer Science and Technology Center
University of Minho
Campus de Gualtar
4710-057 Braga, Portugal

Abstract

Currently, the majority of matchers are able to establish simple correspondences between entities, but are not able to provide complex alignments. Furthermore, the resulting alignments do not contain additional information on how they were extracted and formed. Not only it becomes hard to debug the alignment results, but it is also difficult to justify correspondences. We propose a method to generate complex ontology alignments that captures the semantics of matching algorithms and human-oriented ontology alignment definition processes. Through these semantics, arguments that provide an abstraction over the specificities of the alignment process are generated and used by agents to share, negotiate and combine correspondences. After the negotiation process, the resulting arguments and their relations can be visualized by humans in order to debug and understand the given correspondences.

The existence of heterogeneous data models in computer systems leads to an integration problem when two or more of these systems need to interact and exchange information. This can be due to several reasons, including differences in model representation languages, structure, constraints and semantics, where the origin is often because of a lack of consensus (Sheth and Larson 1990) between those who built the models. Model matching, which consists in finding correspondences between the entities in both representations (or models), is considered to be the first step in solutions for information integration (Euzenat and Shvaiko 2007).

With the increasing popularity of the Semantic Web, more and more data models are being published daily in the form of ontologies. This increase in the amount of models and their heterogeneity is becoming a global scale integration problem. Even so, the demand for complex ontologies in the Semantic Web is small. Actually, empirically, there seems to be a struggle to create very simple and easily shareable and reusable ontologies (as they can more easily become a consensus). However, in the case of business enterprises (Silva, Silva, and Rocha 2011) and in specific research domains such as genetics (Goble and Wroe 2004), complex and heterogeneous ontologies exist. When such ontologies need to be aligned, matches can involve different types of entities, be of different cardinality and form different complex patterns. Still, automatic alignment algorithms are not able to detect these matches, and semi-automatic approaches can be hard to handle from an user's standpoint.

Concurrently, the alignments given by matchers usually do not come with additional information of how they were extracted and formed. Not only it becomes hard to debug the alignment results, but it is also difficult to justify correspondences. This lack of semantics regarding matchers obfuscates the alignment process and constitutes an obstacle to the combination of alignment results.

Following these premises, we propose a method to generate complex ontology alignments that relies on the combination of the overall semantics of matching algorithms and human-oriented ontology alignment definition processes. These semantics is the basis for generating arguments from the techniques employed in matching algorithms, reasoning procedures, and human actions towards alignment definition and correspondences. The generated arguments provide an abstraction over the specificities of the alignment process, which will allow agents to share, negotiate and combine correspondences suggested by different algorithms and/or humans. Furthermore, agents can use additional information (e.g., correspondence patterns, domain specific background knowledge, previous experience, specific preferences and interests) to extract more complex correspondences from those already suggested Finally, using the additional arguments and their relations established during the negotiation process, a human-oriented view of the abstracted alignment process can be provided, allowing debugging and containing justifications for the given correspondences.

Along with this proposal, we envisage an overall collaborative ontology alignment solution where ontology alignments, their formation process and justifications can be shared and reused by a community of ontology engineers that participate in the negotiation process through simple interactions. The process leads to the evolution and refinement of alignments over time and allows the participation of non-expert users.

This paper is organized as follows: the next section provides a brief background on matching algorithms. Afterwards, the overall envisaged alignment solution is presented, followed by its main contributions, more specifically in the automatic extraction of complex correspondences through argumentation. Finally, a discussion on evaluation methods is given along with conclusions.

Background

Complex and heterogeneous correspondences in alignments are hard to find and establish automatically. The process not only requires information that in most cases is not available to the matcher (background knowledge), but also needs to deal with ambiguity, handle uncertainty and possibly provide partial alignments (Shvaiko and Euzenat 2008). Such a process can easily become unfeasible and non scalable.

Ontology matching approaches can be classified as either automatic or semi-automatic (Eidoon, Yazdani, and Oroumchian 2007; Shvaiko and Euzenat 2005). While the former try to extract the alignment without human intervention, the latter can provide more complex and reliable alignments at the cost of human intervention. Due to the dynamics of new emerging applications, run time alignment has become a necessity (Shvaiko and Euzenat 2005).

Currently, the majority of matchers are able to establish simple correspondences (level 0 and 1) between entities. They establish equivalence and subsumption relations between two entities of two ontologies, and provide an associated confidence degree. VBOM (Vector Based Ontology Matching) (Eidoon, Yazdani, and Oroumchian 2007) is such a matcher. It is an automatic structural-level ontology alignment technique that matches vector representations of ontology concepts, estimating their similarity degree through the cosine of the angle between the vectors.

RiMOM (Risk Minimization based Ontology Mapping) (Li et al. 2008) is a multiple strategy ontology alignment framework based in Bayesian decision theory that is able to determine, at run time, the matching methods to use based in the textual and structural ontology similarity measures. RiMOM has the particularity of establishing correspondences with multiple $n : m$ cardinality.

Similarly, the MLMA (Multi-Level MAtching) framework is capable of defining $n : m$ correspondences. The framework allows the application of one or more similarity measures per level, where a partial order is enforced to the levels. The output candidate results of one level are fed to the next level as input along with the alignment ontologies.

GLUE (Doan et al. 2002) is an instance-level and multiple strategy ontology matching framework based in machine learning. Although GLUE achieved, according to the authors experiments, a node matching accuracy of $66-97\%$, it works with a rather simple definition of ontology (taxonomy) and can only generate level 0 alignments with 1:1 cardinality.

In order to retrieve background knowledge, Quix, Roy, and Kensche propose the use of background ontologies obtained using search queries from the input ontologies to be aligned. This approach has been implemented in the semi-automatic GeRoMeSuite framework, which is not restricted to ontology alignment and features several lexical and structural matching strategies. However, the focus of this work is not to extract complex alignments but to increase the performance in terms of precision and recall.

Other approaches include the schema-level COMA++ (COmbination of MAtching algorithms) (Aumueller et al. 2005), Similarity Flooding (Melnik, Garcia-Molina, and Rahm 2002), AnchorPrompt (Noy and Musen 2001) and Falcon-AO (Hu and Qu 2008). All these and the above described approaches are not able to provide complex level 2 alignments and only a few extract n:m cardinality alignments.

In order to establish complex alignments, semi-automatic alignment approaches that involve user interaction and try to handle the drawbacks of automatic matchers exist. OLA (OWL-Lite Alignment) (Euzenat et al. 2004) is an alignment tool for ontologies expressed in OWL that provides functionalities such as (i) automated computation and manual construction of alignments, and (ii) visualization and comparison of ontologies and alignments. Others include the service-oriented MAFRA (MApping FRAmework) (Maedche et al. 2002) and FOAM (Ehrig and Staab 2004; Ehrig and Sure 2005).

Even with the wide variety of available tools and features, the responsibility of establishing complex (e.g., level 2) alignments belongs entirely to the user. This is a cumbersome task, specially when dealing with huge ontologies (Falconer and Storey 2007). In this sense, Zhdanova and Shvaiko (2006) propose a community-driven ontology matching approach where automatically generated matches can be manually edited, shared and reused between members of communities sharing similar interests or in the same collaboration environments. This reduces the initial matching effort and distributes the task of refining the final alignment throughout the community. Simultaneously, it provides an environment for the evaluation of automatic ontology matching algorithms. The matching process relies on several resources in order to solve the heterogeneity problem. These include information about users, information about communities, groups and social networks, and tools for automatic ontology matching. OntoMediate (Correndo and Alani 2008) also focuses in collaborative ontology alignment. Most specifically the impact on the alignment of ontologies of the social interactions, collaboration and user feedback in a community is studied.

Although alignment meta-data are provided by some ontology matchers, the process and its semantics are still obfuscated and no justifications/explanations are presented.

Overall Perspective

Ontology alignments represent knowledge, which can be "produced, consumed, refined, stored, retrieved, shipped and recycled in a continuous loop in which both humans and machines play an important role" (Tijerino, Al-Muhammed, and Embley 2004). Following this premise and the principles described in (Zhdanova and Shvaiko 2006), our overall perspective of an ontology alignment solution goes towards collaborative and trust-based reuse and refinement of complex ontology alignments through agent negotiation and argumentation.

The rationale behind this perspective is that negotiation of correspondences amongst software agents, which can represent a specific matching algorithm or a human in a dis-

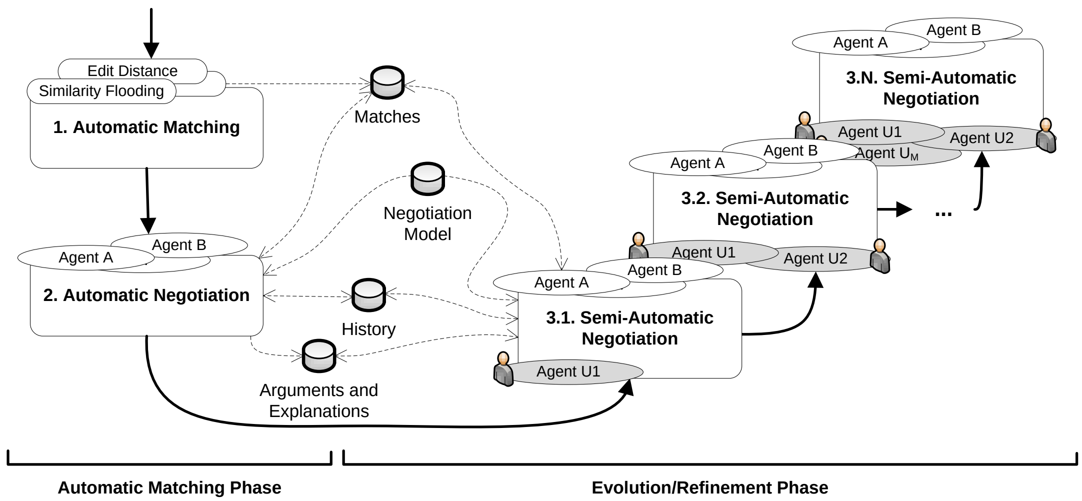

Figure 1: Complex ontology alignment process: starts with an alignment request that executes the required automatic alignment algorithms. It is followed by an automatic negotiation step that enters in an iterative semi-automatic negotiation subprocess triggered by user interaction.

tributed and extensible environment, can greatly improve the alignment result. While human communities based in trust and domain knowledge provide the necessary background knowledge and context (Zhdanova and Shvaiko 2006), software agents can easily exploit matching techniques, alignment patterns (Scharffe and Fensel 2008) and previously established alignments for new alignment construction. Also, the use of an argumentation approach not only allows agent arguments and their relations to be used as justifications for correspondences, but also as an abstract representation and visualization of the alignment and negotiation process.

The envisaged alignment process contains two phases: the automatic matching phase, and the evolution/refinement phase (see figure 1). Even though the whole process can be considered semi-automatic, an unrefined alignment is immediately obtained without user intervention from the automatic matching phase. Then, the iterative refinement phase evolves the initial alignment according to user interaction in the collaborative interface service.

In a simple example scenario, the alignment process starts with a request from the user that chooses which automatic matching algorithms and techniques must be applied and the set of trusted users that have refinement permissions. This request launches an iterative negotiation process that starts with the execution of automatic matchers, and then a negotiation of the best alignment using their output correspondences. The negotiation process also feeds on the complex alignment negotiation model and knowledge from previous alignments. After the automatic negotiation (step 2 in figure 1) is complete, an initial alignment is available to the alignment community along with justifications and a visualization of the argumentation process.

From this point onwards, an iterative process starts where users can provide feedback (e.g., agree, disagree) on the correspondences of the (initial) alignment. When an user submits feedback, the negotiation process is restored with the new information and new arguments might emerge.

The resulting alignment service features refinement operations over alignments. As it is a social service, if user a requests (and is the author of) a specific alignment, he can choose a restricted community (a set U of trusted users) to have refinement permissions over the alignment. These operations might include the request to edit, add and remove correspondences. Their actions and opinions will then be taken into account in the refinement negotiation process through a representative agent. The impact is affected by the user's social status and domain expertise. Such a service gathers the collaborative effort of communities in order to refine initial alignments produced by the combination and negotiation of the results of several different matching algorithms. Furthermore, the collected information is relevant to assessing the quality of the (initial) alignment.

Negotiation and Argumentation

The existence of several matching techniques and algorithms has led to multiple alignment approaches that combine these algorithms in order to merge their strengths. The increased complexity associated with these new approaches, has obfuscated the ontology alignment process, making it difficult to understand by most specialists, and a black box to domain experts.

In this sense, we propose a new method to ontology alignment that provides an abstraction over the specificities of the ontology alignment process, allowing specialists and domain experts to easily visualize the reasoning process and actions behind the resulting alignment.

The semantics required for the abstraction is captured by an ontology that describes several matching algorithms and human alignment actions in the form of arguments. These arguments can be shared by software agents in order to negotiate and combine correspondences suggested by different algorithms and/or humans. After the negotiation process, the arguments along with their relationships can be presented as a visualization of the ontology alignment process that includes justifications for the resulting correspondences.

Although a common ground is required for agents to interpret the arguments, each agent can have its own interpretation of the ontology alignment domain and employ different data, techniques and algorithms (e.g., correspondence patterns, domain specific background knowledge, previous experience, specific preferences and interests) to propose correspondences and generate arguments.

A suitable argumentation framework for this purpose is the EAF (Extensible Argumentation Framework) (Maio, Silva, and Cardoso 2011a), which is a generic three-layered framework where agents adopt a generic and domain-independent argument-based negotiation process.

The Meta-model layer defines the core argumentation concepts (Argument, Statement and Reasoning Mechanism) and a set of relations holding between them. An argument applies a reasoning mechanism (such as rules, methods, or processes) to conclude a conclusion-statement from a set of premise-statements. Intentional arguments are the arguments corresponding to intentions (Bratman 1999) and are supported/attacked by both intentional and non-intentional arguments. With respect to ontology matching, an intentional argument represents a correspondence while information used to support/attack such correspondence is represented by a non-intentional argument. Yet, the existence of a correspondence may support/attack the existence of another correspondence.

The Model layer defines the entities and their relations for a specific domain (e.g. ontology matching) according to a community's perception. The resulting model is further instantiated at the Instance-pool layer. A relation R is established between two argument types (e.g. $(C, D) \in R$) when C supports or attacks D. Through R it is also determined the types of statements that are admissible as premises of an argument. Additionally, arguments, statements and reasoning mechanisms can be structured through the H_A, H_S and H_M relations respectively (vaguely similar to the subclass/superclass relation).

The Instance-Pool layer corresponds to the instantiation of a particular model layer for a given scenario (e.g. agents negotiating the alignment to be established between their ontologies).

Previously, an EAF model for ontology alignment and a process to instantiate it was proposed in (Maio, Silva, and Cardoso 2011b). However, the proposed model is simple, still lacking the semantics needed for explaining matching algorithms and for more complex correspondences to be extracted. Even so, the three-layered architecture of the EAF provides the necessary flexibility to model and represent correspondence patterns (Scharffe and Fensel 2008) and the conditions under which they manifest themselves (Ritze et al. 2010). Furthermore, as arguments in the EAF are defined according to statements playing the roles of premises and conclusions, an initial structure for generating arguments from correspondences is already present. In this sense, we focus on building an EAF model that includes different types of statements and arguments that capture the semantics of the ontology alignment domain. This includes the modeling of correspondence patterns, and defining mapping functions not only according to the current state of the art matching algorithms, but also according to the user interactions.

Following the described overall perspective of ontology alignment, each agent participating in the negotiation process will have access to a pool of matchers and analyzers (see figure 2). While matchers provide an initial set of correspondences (the agent's interpretation of the alignment before negotiation), the analyzers will provide additional facts important to the extraction of more complex correspondences.

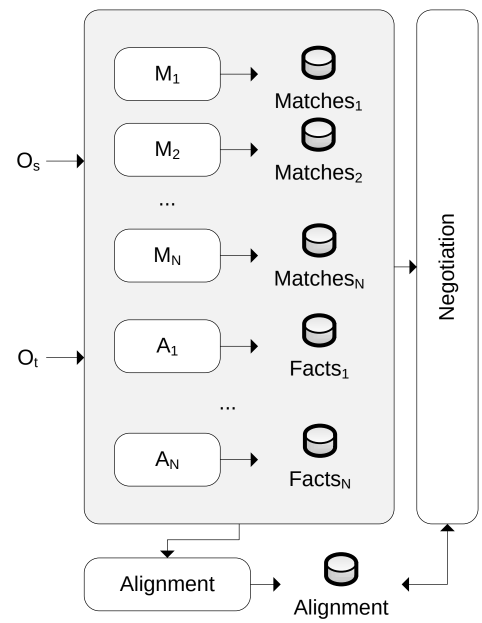

Figure 2: Agent work flow in the ontology alignment negotiation.

The ontology alignment EAF model presented in (Maio, Silva, and Cardoso 2011b) defines statements as 3-tuples (G, c, pos), where G is a matcher, c a correspondence and pos takes as value either $+$ or $-$ according to the confidence degree attributed to c by the matcher G. This definition of statement limits the argumentation process to the results of alignment algorithms. However, if an agent capable of detecting complex correspondences from patterns were to exist, embedding description logic or more expressive expressions (e.g., rules) in statements would be desirable to represent premises. Using these expressions, approaches like the one presented in (Horridge, Parsia, and Sattler 2008) could be employed to provide justifications.

Using the CAT (Class by Attribute Type) correspondence pattern, and the conditions for its detection presented in (Ritze et al. 2010), an argument in favor of the pattern instantiation can be formed using the satisfied conditions as premises and the resulting instantiation as conclusion. If an ontology O_1 contains the class

$White\_Bear_1$, and an ontology O_2 contains the axioms $Bear_2 \sqsubseteq \exists hasColour_2.Colour_2$ and $White_2 \sqsubseteq Colour_2$, a possible correspondence could be $White\_Bear_1 \equiv Bear_2 \sqcap \exists hasColour_2.White_2$. The extraction of this correspondence is triggered by the following conditions being satisfied, which can also be seen as the premises to an argument in favor of the correspondence:

1. $White\_Bear_1 \sqsubseteq Bear_2$
2. $Bear_2 \sqsubseteq \exists hasColour_2.Colour_2$
3. $Nominalization(White\_Bear_1) = White \equiv White_2$
4. $White_2 \sqsubseteq Colour_2$

Different types of statements are needed to describe these expressions (e.g., correspondence statements, ontological statements). Also several reasoning mechanisms must exist to capture the semantics of the processes employed not only by matchers but also by analyzers (e.g., nominalization). Figure 3 depicts the required argument model in order to instantiate arguments for the CAT pattern. Notice that conclusions of arguments $ArgT1$, $ArgT2$, $ArgT3$ and $ArgT4$ can automatically lead to the instantiation of (and become the premises to) the $ArgCAT$ argument. This is only possible due to the specification of the R relationship between arguments and to the patterns conditions being checked and satisfied during the negotiation process.

Although figure 3 only depicts a model for the CAT pattern, similar EAF models can be built for other correspondence patterns.

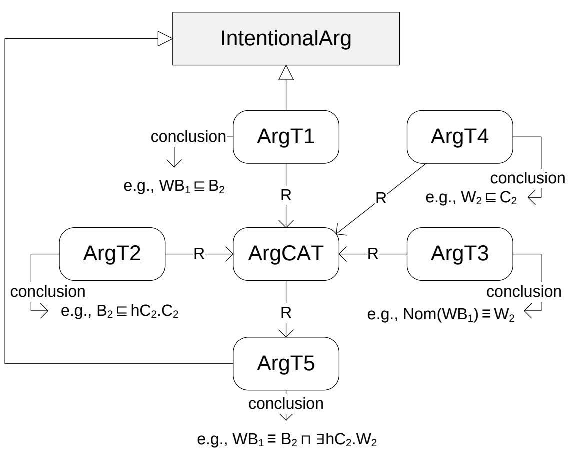

Figure 3: An argumentation model for negotiation of the CAT pattern.

Evaluation

Specific work regarding the benchmarking of Semantic Web and ontology alignment tools has been done over the last decade. In (García-Castro and Gómez-Pérez 2009), a benchmarking methodology for testing Semantic Web implementations is presented, with a life cycle of three phases: (i) planning, (ii) experimentation and (iii) improvement. Reliability issues like scalability, robustness, interoperability, performance and usability are taken into account by this methodology. The OAEI extended this methodology and built a specific benchmarking suite for ontology alignment tools (Euzenat, Ehrig, and Castro 2005).

Besides the OAEI benchmarking suite and the existent evaluation measures (e.g., precision, recall, F-measure), the envisaged collaborative ontology alignment solution provides an environment for result evaluation according the different case studies where experts and non-experts provide refinement feedback on initial alignments. The retrieved feedback can be exploited in assessing the quality and reliability of the initial automatic alignment.

In order to perform an adequate case study evaluation, three different kinds of scenarios were identified:

- Semantic Web ontology integration scenario: simple ontologies built towards consensus;
- Academic/research ontology integration scenario: complex ontologies requiring scalability;
- Business enterprise integration scenario: complex ontologies often with inconsistencies.

In Semantic Web integration scenarios, most correspondences are simple (level 0/1). Still the automatic alignment process must be able to extract them.

A possible research scenario is that of a community of biology scientists building several ontologies that they need to align in order to exchange information.

A concrete business scenario is that of the integration of a legacy system with an ERP (Enterprise Resource System) for a textile and garment enterprise presented in (Silva, Silva, and Rocha 2011). In this scenario, both legacy and ERP systems must be integrated and operate simultaneously. Their integration not only requires complex (level 2) alignments but also bi-directionality. As an example, although a correspondence between birth date and age can be established, it can only be specified through a transformation function (e.g., birthdate2age). Defining an inverse function (e.g., age2birthdate) is also difficult since additional data is required.

Conclusions

This paper proposes a collaborative approach to ontology alignment based in agent negotiation of correspondences through argumentation that includes the detection of complex matches. Being collaborative, the effort of defining and refining alignments is distributed through a community of users and experts. This manual effort is also reduced due to the automatic extraction of complex correspondences. Furthermore, using an argumentation process allows an easy extraction of simple justifications that can be presented to users.

Following the overall proposal, special attention is given to the argumentation process, where the EAF is exploited in order to define an adequate negotiation process and model the ontology alignment domain specific arguments.

Acknowledgments

This work is partially supported by the FCT PhD grant SFRH/BD/70302/2010, and by the Portuguese projects COALESCE (PTDC/EIA/74417/2006) of MCTES-FCT and World Search (QREN11495) of FEDER.

References

Aumueller, D.; Do, H.; Massmann, S.; and Rahm, E. 2005. Schema and ontology matching with coma++. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 906–908. ACM.

Bratman, M. 1999. Intention, plans, and practical reason.

Correndo, G., and Alani, H. 2008. Collaborative support for community data sharing. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 17–20. IEEE.

Doan, A.; Madhavan, J.; Domingos, P.; and Halevy, A. 2002. Learning to map between ontologies on the semantic web. In *Proceedings of the 11th international conference on World Wide Web*, 662–673. ACM.

Ehrig, M., and Staab, S. 2004. Qom–quick ontology mapping. *The Semantic Web–ISWC 2004* 683–697.

Ehrig, M., and Sure, Y. 2005. Foam–framework for ontology alignment and mapping results of the ontology alignment evaluation initiative. In *Integrating Ontologies Workshop Proceedings*, 72.

Eidoon, Z.; Yazdani, N.; and Oroumchian, F. 2007. A vector based method of ontology matching. *Third International Conference on Semantics, Knowledge and Grid* 378–381.

Euzenat, J., and Shvaiko, P. 2007. *Ontology matching*. Springer-Verlag New York Inc.

Euzenat, J.; Loup, D.; Touzani, M.; and Valtchev, P. 2004. Ontology alignment with ola. In *Proceedings of the 3rd International Workshop on Evaluation of Ontology based Tools (EON), Hiroshima, Japan*. Citeseer.

Euzenat, J.; Ehrig, M.; and Castro, R. 2005. Towards a methodology for evaluating alignment and matching algorithms.

Falconer, S., and Storey, M. 2007. A cognitive support framework for ontology mapping. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, 114–127. Springer-Verlag.

García-Castro, R., and Gómez-Pérez, A. 2009. Benchmarking in the semantic web.

Goble, C., and Wroe, C. 2004. The montagues and the capulets. *Comparative and functional genomics* 5(8):623–632.

Horridge, M.; Parsia, B.; and Sattler, U. 2008. Laconic and precise justifications in owl. *The Semantic Web-ISWC 2008* 323–338.

Hu, W., and Qu, Y. 2008. Falcon-ao: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(3):237–239.

Li, J.; Tang, J.; Li, Y.; and Luo, Q. 2008. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering* 1218–1232.

Maedche, A.; Motik, B.; Silva, N.; and Volz, R. 2002. Mafra - a mapping framework for distributed ontologies. *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web* 69–75.

Maio, P.; Silva, N.; and Cardoso, J. 2011a. Eaf-based negotiation process. In *The 4th International Workshop on Agent-based Complex Automated Negotiation (ACAN) at AAMAS*.

Maio, P.; Silva, N.; and Cardoso, J. 2011b. Generating arguments for ontology matching. *10th International Workshop on Web Semantics (WebS)*.

Melnik, S.; Garcia-Molina, H.; and Rahm, E. 2002. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, 117–128. IEEE.

Noy, N., and Musen, M. 2001. Anchor-prompt: Using non-local context for semantic matching. In *Proceedings of the workshop on ontologies and information sharing at the international joint conference on artificial intelligence (IJCAI)*, 63–70. Citeseer.

Quix, C.; Roy, P.; and Kensche, D. 2011. Automatic selection of background knowledge for ontology matching. *3rd International Workshop on Semantic Web Information Management (SWIM 2011)*.

Ritze, D.; Völker, J.; Meilicke, C.; and Šváb-Zamazal, O. 2010. Linguistic analysis for complex ontology matching. *Ontology Matching* 1.

Scharffe, F., and Fensel, D. 2008. Correspondence patterns for ontology alignment. *Knowledge Engineering: Practice and Patterns* 83–92.

Sheth, A., and Larson, J. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)* 22(3):183–236.

Shvaiko, P., and Euzenat, J. 2005. A survey of schema-based matching approaches. *Journal on Data Semantics IV* 146–171.

Shvaiko, P., and Euzenat, J. 2008. Ten challenges for ontology matching. *On the Move to Meaningful Internet Systems: OTM 2008* 1164–1182.

Silva, J.; Silva, N.; and Rocha, J. 2011. Online, iterative and interactive integration of legacy system and up-to-date erp: a case study and systematization. *INFORUM 2011 - Informatics Symposium*.

Tijerino, Y.; Al-Muhammed, M.; and Embley, D. 2004. Toward a flexible human-agent collaboration framework with mediating domain ontologies for the semantic web. *Meaning Coordination and Negotiation (MCN-04)* 131.

Zhdanova, A., and Shvaiko, P. 2006. Community-driven ontology matching. *The Semantic Web: Research and Applications* 34–49.