Generating Arguments for Ontology Matching

Paulo Maio
GECAD – ISEP - IPP
School of Engineering
Polytechnic of Porto
Porto, Portugal
pam@isep.ipp.pt

Nuno Silva
GECAD – ISEP - IPP
School of Engineering
Polytechnic of Porto
Porto, Portugal
nps@isep.ipp.pt

José Cardoso
University of Trás-os-Montes
and Alto Douro
Vila Real, Portugal
jcardoso@utad.pt

Abstract

Agents embedded on open, dynamic and decentralized environments adopt different ontologies to describe their domain of discourse. Yet, agents have no prior knowledge of the other agents with whom they will interact. Therefore, a consistent and compatible communication relies on the agents' ability to reconcile in run-time the vocabulary used in their ontologies whose result is a set of correspondences. Since each party might have its own perspective about what are the best correspondences, conflicts arise. To address such conflicts, it is envisaged as a suitable approach that agents adopt the generic argument-based negotiation process presented in [1]. *A critical issue of that process concerns the arguments generation process according to the argumentation framework that it relies on. In this paper it is proposed an automatic process to generate arguments regarding the ontologies reconciliation through semantic interpretations of third party generated correspondences.*

1. Introduction

The FIPA agent communication model [2] relies on the assumption that two agents, wishing to converse, share a common ontology for the domain of discourse, i.e. the agents assign the same meaning to the symbols used in the messages [2]. Such assumption may be acceptable in applications scenarios running on controlled environments, but agents embedded in open, dynamic and decentralized environments such assumption is not feasible because different parties (i.e. people and agents) adopt different ontologies for their descriptions, making heterogeneity problems arise between communication partners. Agents operating in an open, ill-specified environment, often have no prior knowledge of the other agents with whom they will interact, compelling the agents to decide at run-time about each and all correspondences they will adopt in the conversation. Therefore, the ability to reconcile ontologies is a corner stone for agents' interoperability. In the literature this reconciliation problem is usually called *Ontology Matching* [3]. This reconciliation relies on establishing a set of correspondences (i.e. an alignment) between the agents' ontologies which are further exploited to interpret or translate exchanged messages and their content.

Research initiatives in ontology matching have developed many algorithms and systems able to generate (semi-) automatic correspondences between two different but overlapping ontologies [3]. However, different systems have contradictory and inconsistent perspectives about candidate correspondences. Additionally, agents pursuing their own goals might have different preferences and interests due, for instance, to the subjective nature of ontologies, the context and the alignment requirements. Consequently, conflicts arise between agents about which are the best correspondences.

Approaches relying on an argument-based negotiation such as [4], [5] were proposed to allow agents to reach a consensus about the correspondences that must be part of the alignment enabling them to communicate and mutually understand each other. However, on those approaches the correspondences are provided by a single Ontology Alignment Service (OAS) that is common to all agents. This constraint somehow implies that agents had previously agreed on which OAS to use. Moreover, since all agents are using the same OAS and therefore implying agents' ability to understand all provided data, it also means that the main difference between agents' individual knowledge with respect to the ontology alignment domain rely exclusively on the preferred audience [6] . Given that, argumentation outcome basically corresponds to the intersection of the alignments proposed by each agent. In that sense, agents do not have the chance to exploit existing differences between agents' knowledge in order to persuade opponent agents.

To overcome these and others (e.g. lack of quantitative or opinion factors) limitations it is envisaged as a suitable approach that agents adopt the generic and domain-independent argument-based negotiation process presented in [1], which relies on the Extensible Argumentation Framework (EAF) [1], [7], to address conflicts about correspondences. For that, the EAF Instantiation phase which concerns the arguments generation is seen as a critical issue. In that sense, this paper proposes an automatic process to generate arguments according to the

EAF structure for the ontology matching domain through semantic interpretations of third party generated correspondences.

The rest of this paper is organized as follows: the next section describes the main structures and concepts of the EAF on which the argument-based negotiation process to be adopted by communicating agents rely on. Section 3 describes the proposed arguments generation process for the ontology matching domain concerning the EAF instantiation phase of the negotiation process. Finally, section 4 draws conclusions and comments on future work.

2. EAF

This section describes briefly and informally the main features of the Extensible Argumentation Framework (EAF) [1], [7]. The EAF comprehends three modeling layers as depicted in Figure 1.

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 ([8], [9]). Yet, intentional arguments 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. The R relation 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). A statement instance B_1 is said to be in conflict with another statement instance B_2 when B_1 states something that implies or suggests that B_2 is not true. The statement conflict relation is asymmetric (in Figure 1 B_2 conflicts with B_1 too). The support and attack relationships (R_{sup} and R_{att} respectively) between argument instances are automatically inferred exploiting (i) the R relations defined at the model layer and (ii) the existing *premise* relations and the statements conflicts at this level.

An EAF model may reuse and further extend the argumentation conceptualizations of several existing EAF models. Inclusion of an EAF into another EAF is governed by a set of modularization constraints ensuring that no information of included EAF is lost.

3. Ontology matching' EAF instantiation

This section describes the developed process for the EAF Instantiation phase (introduced in [1]) concerning the ontology matching field. Yet, the developed process is not committed to any particular EAF model for the ontology matching domain.

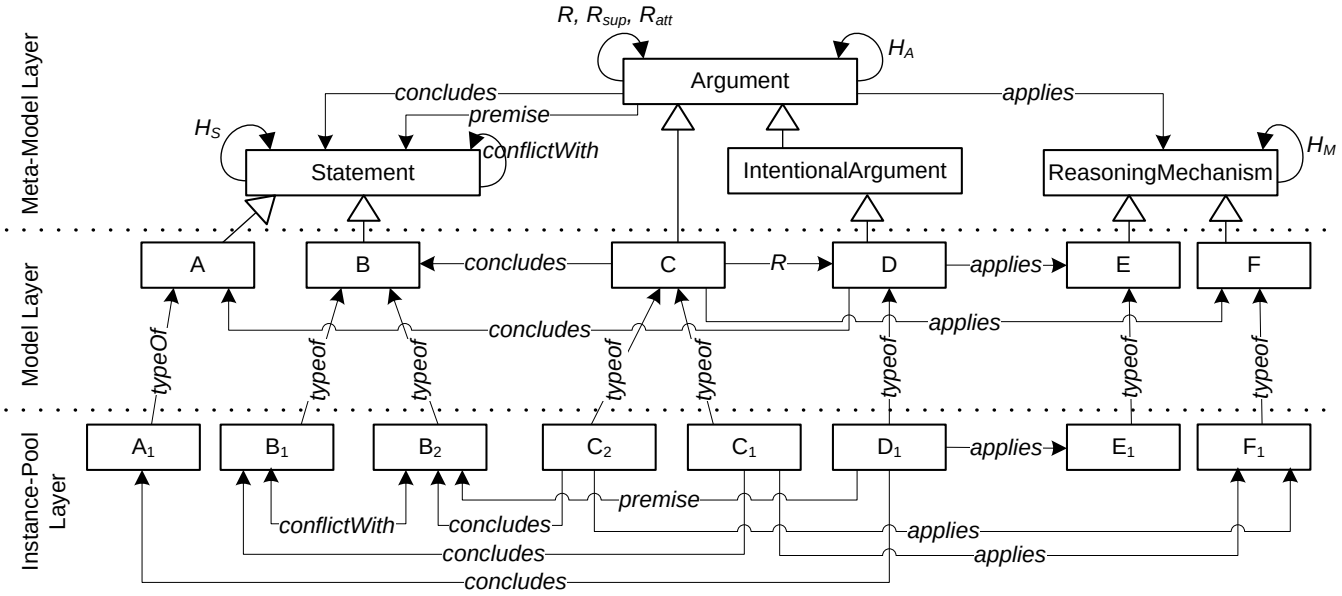

Figure 1. The three EAF modeling layers

Despite all the existing (conceptual and practical) differences between matching algorithms (matchers), all of them have as output a set of correspondences. Each correspondence (or match) is a 4-uple: $c = (e, e', R, n)$ where e and $e'$are the source and target ontology entities respectively, R is a relation (e.g. equivalence, more specific) and n is a confidence value, typically in the [0-1] range. Correspondences and matchers generating them are the units of information that are used to generate argument instances.

Contrary to arguments, EAF do not establish any structure for statements. In that sense, we have defined a matching statement as 3-uple $s = (G, c, pos)$ where c is a correspondence, G is a univocal matcher identification and $pos \in \{+, -\}$ stating if G is in favor $(+)$ or against $(-)$ c.

Each statement instance has an EAF-model statement type which depends on two dimensions:

- the match's content of the statement-instance, namely:
  - the type of the related entities. Typically, an ontology entity is classified as (i) concept or class (C), (ii) property or relation and (iii) instance (I). A property entity is further sub-classified either as object-property (OP) or as data-property (DP);
  - the relation holding between them (e.g. equivalence, synonym, broader, narrower);
- the matcher generating the match.

Agents may interpret the same correspondence differently. For example, a correspondence stating that an equivalence relation holds between concepts c_1 and c_2 may be interpreted as:

- the equivalence relation holds between the labels of concept c_1 and concept c_2, if the proposing matcher only exploits (i.e. compares) the entities' labels to propose such match;
- the internal structure of both concepts is equivalent, if the proposing matcher only exploits and compares the internal structure elements of each concept.

Despite these two possible interpretations, many others can be done.

The position of a matcher about a correspondence (i.e. for or against) is determined by the degree of confidence (i.e. n) that the matcher has on that correspondence. So, a matcher G_1 is in favor of a correspondence c_1 if its confidence value on c_1 is equal or greater than a given threshold value (i.e. $n \geq tr_+$). Similarly, G_1 is against c_1 if its confidence value on c_1 is below than another threshold value (i.e. $n < tr_-$), otherwise (i.e. $tr_- \leq n < tr_+$) G_1 is neither in favor nor against c_1 and therefore c_1 is ignored. Typically, instead of a single value to tr_+ and tr_- common to all matchers, it is settled one value to tr_+ and tr_- by matcher. When a matcher does not propose any correspondence about a given pair of ontology entities, there are two mutually exclusive alternatives: (i) consider that matcher is against any correspondence relating those entities or (ii) consider that matcher have no opinion about that and therefore is neither for nor against any correspondence relating those entities. Moreover, matchers are classified based on the internal algorithm applied to generate correspondences. Such classification relates to one possible reasoning mechanisms of an argument. Notice that, a matcher may apply different algorithms to discover different types of correspondences (e.g. based on the relation holding between entities or based on the entities type). As a result, a semantic interpretation of matchers and correspondences must be done by a domain expert (or third party approaches) in order to define a mapping function that provides the required information to create statements and arguments. An abstract mapping function is depicted in Table 1. Formally, a mapping function is defined as $mf: G \times c \rightarrow S \times M \times pos$ where S is a statement type and M a reasoning method of the EAF model.

Table 1. Abstract mapping function

Matcher	Match Content			Stat. Type	Reasoning Method	tr_+	tr_-
	e	e'	R				
G_1	any			S_1	M_1	0.90	0.70
G_2	C	C	any	S_2	M_2	0.85	0.60
	DP	DP		S_3			
	OP	OP		S_4			
G_3			=	S_1	M_1	0.75	0.50
			syn.	S_5	M_3		

Since a mapping function represents the knowledge and perception that a particular agent has on the ontology matching domain, it might differ between agents. Notice that, each agent may use a distinct set of matchers to instantiate the EAF model. Yet, two agents using the same matcher may have a distinct interpretation.

Consider that D is the set of data/information collected by an agent from the environment such that $d \in D$ is a pair (G, c) where c is a correspondence and G is the identification of the matcher/agent from where c was collected. Also consider that (i) for an argument type x the function $instA(x)$ returns all argument instances of type x, (ii) for a statement type s the function $instS(s)$ returns all statement instances of type s, (iii) for an argument instance a the function $iconcl(a)$ returns the statement instance concluded by a and (iv) ST as the set of all statement types defined in the EAF model. Given that, an EAF model for the ontology matching domain is instantiated as follows.

First, through the mapping interpretation function, every collected element give rise to (i) a statement instance s of type S whose content is (G, c, pos) according to the defined matching statement structure, (ii) a reasoning method instance m of type M representing the algorithm used by G and (iii) an argument instance a concluding s applying m (see Algorithm 1).

Since no argument type is specified, a is classified as $Argument$ (the root type). Therefore, in the next step all argument instances are reclassified to a more specific type, by means of an automatic process relying on the EAF

model information only. For example, a rule-based process whose rules capture the following sentence: set argument type at to any argument instance whose conclusion-statement is of type st and the reasoning method used is of type mt. Notice that, according to the EAF meta-model an argument type defines the type of statement it concludes and its reasoning mechanism.

Algorithm 1 Generating statements and arguments

Require: The EAF model to instantiate, a mapping interpretation function mf and a set of collected data D
Ensure: a set of arguments, statements and reasoning methods

for all $d \in D$ **do**
  $(G, c) = d$
  $(S, M, pos) = mf(G, c)$
  $s = createStatementInst(G, c, pos, S)$
  $m = createReasoningMethodInst(G, M)$
  $a = createArgumentInst(s, m)$
end for

Yet, since just a few ontology matching algorithms are able to justify why a given correspondence is suggested (e.g. [10]), argument instances are initially set with no statements as premise. To overcome such issue, the premises of argument instances are defined by exploring the R-ships between argument types defined in the EAF model: a statement-instance s_a is premise of an argument instance b if:

- the statement concluded by b (say s_b) satisfies a given condition (i.e. $condition(s_a, s_b)$) (cf. bellow for details),
- both have the same position (i.e. $pos_a = pos_b$) and
- s_a is concluded by an argument instance a whose type (say x) is related with the argument type of b (say y) through R (i.e. $(x, y) \in R$).

Algorithm 2 captures these rules.

Algorithm 2 Setting argument premises

Require: An EAF model instantiated with argument and statement instances
Ensure: Establishes the premises of the argument instances

for all $(x, y) \in R$ **do**
  for all $a \in instA(x)$ **do**
    $(G_a, c_a, pos_a) = s_a = iconcl(a)$
    for all $b \in instA(y)$ **do**
      $(G_b, c_b, pos_b) = s_b = iconcl(b)$
      if $condition(s_a, s_b)$ *and* $(pos_a = pos_b)$
        Add s_a *as premise of* b
      end if
    end for
  end for
end for

Typically, the function $condition(s_a, s_b)$ checks if both statements are about the same correspondence (i.e. $c_a = c_b$). However, depending on the type of statements being verified such condition may be different. For example, consider the following statements:

- $s_a = (G_a, c_a, +)$ and $c_a = (e_1, e_2, =, 0.9)$;
- $s_b = (G_b, c_b, +)$ and $c_b = (e_{11}, e_{22}, =, 1.0)$;

such that s_a states that an equivalence relation holds between concepts e_1 and e_2 while s_b states that the super-concepts of concepts e_{11} and e_{22} respectively are similar. In that sense, s_a may be set as premise of an argument concluding s_b if e_1 is a super-concept of e_{11} and e_2 is a super-concept of e_{22}. Similar conditions may exist based on others ontological notions such as sub-concepts, siblings, domain and range. This kind of conditions usually permits that conclusions of intentional arguments are used as premises of arguments promoting others intentions.

At last, it established the existing conflicts between statements in order to further derive all existing support and attack relationships between argument instances. Therefore, a statement-instance s_a is in conflict with a statement instance s_b if at least one of the following conditions holds (see Algorithm 3):

- both statement instances have the same statement type and s_a is about the same correspondence that s_b (i.e. $c_a = c_b$) but their positions are contradictory (i.e. $pos_a \neq pos_b$);
- s_a is concluded by an argument a of type x and s_b is concluded by an argument b of type y and x is related with y through R (i.e. $(x, y) \in R$) and both statements satisfies a given condition (i.e. $condition(s_a, s_b)$) but their positions are contradictory (i.e. $pos_a \neq pos_b$).

Algorithm 3 Setting conflicts between statements

Require: An EAF model instantiated with argument and statement instances
Ensure: Establishes set of conflicts between statements

for all $s \in ST$ **do**
  for all $s_a \in instS(s)$ **do**
    for all $s_b \in instS(s)$ **do**
      if $(s_a \neq s_b)$ *and* $(c_a = c_b)$ *and* $(pos_a \neq pos_b)$
        Set s_a *is in conflict with* s_b
      end if
    end for
  end for
end for
for all $(x, y) \in R$ **do**
  for all $a \in instA(x)$ **do**
    $(G_a, c_a, pos_a) = s_a = iconcl(a)$
    for all $b \in instA(y)$ **do**
      $(G_b, c_b, pos_b) = s_b = iconcl(b)$
      if $condition(s_a, s_b)$ *and* $(pos_a \neq pos_b)$
        Set s_a *is in conflict with* s_b
      end if
    end for
  end for
end for

4. Conclusions

This paper proposed that communicating agents adopt the generic EAF-based negotiation process presented in [1] to resolve existing conflicts about which correspondences must be established between their ontologies. One of the critical phases of that process is the instantiation and classification of arguments. Arguments are domain dependent and in the case of ontology alignment it is suggested to exploit correspondences generated by third party algorithms/agents. For that, we proposed the adoption of a matching-to-statement mapping function, that provides each agent with a private interpretation of matches and matchers. Based on the mapping function, arguments and statements are generated. Concerning the definition of argument' premises and the conflict relation between statements the process relies on a very simple yet configurable approach based on a condition function. At the end of the instantiation phase the support and attack relations holding between arguments are automatically established through the EAF features.

The proposed arguments generation process has two main advantages when compared to the argument generation process of related works (e.g. [4], [5]). First, it makes possible through the condition function that the fact of accepting/rejecting a given correspondence influences positively or negatively the acceptance/rejection of others correspondences. This is not possible on the other works. Second, contrary to other works it is not mandatory that two arguments concluding statements about the same correspondence but with contradictory positions (i.e. in favor/against) attack each other. For example, this permits an argument concluding that two concepts have similar labels does not attack an argument concluding that the comments of the same two concepts are completely different and vice-versa. Instead, according to an EAF model, one may view the first argument as a reason to support a third argument concluding that equivalence holds between the two concepts and the second argument as a reason to attack the third argument.

Experiences (not reported in this paper due lack of space) performed on a set of commonly used ontology alignment cases show that the EAF-based argumentation process together with the match-statement-argument generation process leads to a substantial improvement in the quality of the alignment (in terms of precision and recall) in comparison with the intersection of agents' private alignments. Experiences also showed that after the argumentation process runs agents remain with some conflicts about correspondences. To resolve remaining conflicts, the team will investigate the combination of the proposed approach with the concession-based approach presented in [11]. Another interesting research direction is related with (i) the generalization of the specification process of an EAF model for the ontology alignment domain, (ii) measuring the impact of the EAF model used by agents in the results of the overall argumentation process and (iii) providing agents with the ability to learn and improve their argumentation strategies based on their past experiences.

ACKNOWLEDGMENTS

This work is partially supported by the Portuguese MCTES-FCT project COALESCE (PTDC/EIA/74417/2006).

5. References

[1] P. Maio, N. Silva, and J. Cardoso, "EAF-based Negotiation Process," in *The 4th International Workshop on Agent-based Complex Automated Negotiation (ACAN) at AAMAS*, 2011.

[2] FIPA, *FIPA ACL Message Structure Specification*. 2002.

[3] J. Euzenat and P. Shvaiko, *Ontology Matching*, 1st ed., vol. 1, 1 vols. Heidelberg, Germany: Springer-Verlag, 2007.

[4] L. Laera, I. Blacoe, V. Tamma, T. R. Payne, J. Euzenat, and T. Bench-Capon, "Argumentation over Ontology Correspondences in MAS," in *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, p. 228, 2007.

[5] Paul Doran, T. Payne, V. Tamma, and Ignazio Palmisano, "Deciding Agent Orientation on Ontology Mappings," in *9th International Semantic Web Conference (ISWC)*, 2010.

[6] T. J. M. Bench-Capon, "Persuasion in Practical Argument Using Value-based Argumentation Frameworks," *J Logic Computation*, vol. 13, no. 3, pp. 429-448, Jun. 2003.

[7] P. Maio and N. Silva, *Technical Report: The Extensible Argumentation Framework*. Porto, Portugal: ISEP, 2011.

[8] M. Bratman, *Intention, Plans and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.

[9] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.

[10] P. Shvaiko, F. Giunchiglia, P. P. D. Silva, and D. McGuinness, "Web Explanations for Semantic Heterogeneity Discovery," in *Proc. 2nd European Semantic Web Conference (ESWC)*, vol. 3532, p. 303—317, 2005.

[11] N. Silva, P. Maio, and J. Rocha, "An approach to ontology mapping negotiation," in *Workshop on Integrating Ontologies of the Third International Conference on Knowledge Capture*, 2005.