HAL
open science

Argumentation for reconciling agent ontologies

Cassia Trojahn dos Santos, Jérôme Euzenat, Valentina Tamma, Terry Payne

To cite this version:

Cassia Trojahn dos Santos, Jérôme Euzenat, Valentina Tamma, Terry Payne. Argumentation for reconciling agent ontologies. Atilla Elçi, Mamadou Koné, Mehmet Orgun. Semantic Agent Systems, Springer, pp.89-111, 2011, 10.1007/978-3-642-18308-9_5 . hal-00781028

HAL Id: hal-00781028

https://inria.hal.science/hal-00781028v1

Submitted on 25 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Argumentation for reconciling agent ontologies

Cássia Trojahn, Jérôme Euzenat, Valentina Tamma and Terry R. Payne

Abstract Within open, distributed and dynamic environments, agents frequently encounter and communicate with new agents and services that were previously unknown. However, to overcome the ontological heterogeneity which may exist within such environments, agents first need to reach agreement over the vocabulary and underlying conceptualisation of the shared domain, that will be used to support their subsequent communication. Whilst there are many existing mechanisms for matching the agents' individual ontologies, some are better suited to certain ontologies or tasks than others, and many are unsuited for use in a real-time, autonomous environment. Agents have to agree on which correspondences between their ontologies are mutually acceptable by both agents. As the rationale behind the preferences of each agent may well be private, one cannot always expect agents to disclose their strategy or rationale for communicating. This prevents the use of a centralised mediator or facilitator which could reconcile the ontological differences. The use of argumentation allows two agents to iteratively explore candidate correspondences within a matching process, through a series of proposals and counter proposals, i.e., *arguments*. Thus, two agents can reason over the acceptability of these correspondences without explicitly disclosing the rationale for preferring one type of correspondences over another. In this chapter we present an overview of the approaches for alignment agreement based on argumentation.

Cássia Trojahn
INRIA & LIG, e-mail: cassia.trojahn@inrialpes.fr

Jérôme Euzenat
INRIA & LIG, e-mail: jerome.euzenat@inrialpes.fr

Valentina Tamma
University of Liverpool, e-mail: v.tamma@liverpool.ac.uk

Terry R. Payne
University of Liverpool, e-mail: t.r.payne@liverpool.ac.uk

1 Introduction

The problem of dynamic reconciliation of vocabularies, or *ontologies*, used by agents during interactions has recently received significant attention, motivated by the growing adoption of service-oriented and distributed computing. In such scenarios, agents are situated in open environments and may encounter unknown agents offering new services due to changes in a user's context or goal. These multi-agent systems are, by nature, distributed and heterogeneous, and as such, ontologies play a fundamental role in formalising the concepts that agents perceive, share, or encounter. However, as the heterogeneity that permeates these environments increases, fewer assumptions on the vocabulary and content of these ontologies can be made, hindering seamless interactions between the agents. Thus, mechanisms that can dynamically and autonomously reconcile the differences between ontologies are essential if agents are to communicate within such open and evolving environments.

Early systems avoided the problem of ontological heterogeneity by relying on the existence of a shared ontology, or simply assuming that a canonical set of ontology correspondences (possibly defined at design time) could be used to resolve ontological mismatches. However, such assumptions work only when the environment is (semi-) closed and carefully managed, and no longer hold in open environments where a plethora of ontologies exist. Moreover, the assumption of a common ontology forces an agent to comply with a fixed, but highly constrained view of the world, with respect to a set of predefined tasks and, as a consequence, abandon its own world view, which may have evolved due to interactions with other agents [8].

To facilitate the communication between two agents, they first need to establish a set of correspondences (or an *alignment*) between their respective ontologies. The reconciliation of heterogeneous ontologies has been investigated at length by research efforts in *ontology matching* [15], which tries to determine suitable correspondences between two ontologies. However the increased availability of mechanisms for ontology matching has facilitated the potential construction of a plethora of different correspondence sets between two ontologies, depending on the approach used. In addition, the majority of traditional matching approaches cannot be easily utilised as part of dynamic interaction protocols since they either require human intervention or they align the ontologies at design time. Even when alignments are pre-computed and stored within some alignment library, the selection of a possible correspondence that would be mutually acceptable to two transacting agents can be problematic, as the choice of correspondences can be highly dependent on the current task or available knowledge. For example, an agent may prefer correspondences which have been approved by its own institution and another one may prefer a correspondence designed for the same task. These may not be easy to reconcile. Hence, some correspondences may be preferable to some agents, but may be unsuitable or untrustworthy to others. In addition, it may not always be desirable for an agent to disclose a preference for a given type of correspondence as this may reveal its goal, and thus compromise it's ability to negotiate strategically with other agents within a competitive environment. Thus it is not always possible to utilise a

collaborative approach, or exploit the use of a third party mediator to determine a mutually acceptable set of correspondences.

The agreement on a mutually acceptable alignment is an important problem that therefore arises when different parties need to reconcile private, yet potentially conflicting preferences over candidate correspondences. Such an agreement can be achieved through a negotiation process whereby agents iteratively exchange proposals and counter-proposals [28, 20] until some consensus is reached. Argumentation can be seen as a qualitative negotiation model based on the construction and comparison of arguments [12, 29, 3], either supporting or refuting a set of possible propositions. Thus, by considering these propositions as correspondences (with justifications that support their use), agents can strategically argue in favour of (or against) possible correspondences given their individual strategies or preferences.

This chapter presents an overview on the approaches for alignment agreement based on argumentation. The different approaches are presented following two scenarios. In the first one, agents with different preferences need to agree on the alignment of their ontologies in order to communicate with each other. For the second scenario, specialised matcher agents rely on different matching approaches and argue on their individual results in order to obtain a consensual alignment.

The remainder of this chapter is structured as follows. First, we provide the basic definitions of ontology matching and argumentation frameworks (§2). Next, two argumentation frameworks for alignment agreement are introduced (§3). The different proposals on argumentation for alignment agreement are then presented (§4). The limitations and challenges in this domain are discussed (§5). Finally, related work (§6) and final remarks (§7) are presented.

2 Foundations: Alignment and Argumentation Frameworks

2.1 Ontology matching

An **ontology** typically provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary. As different agents within an open multi-agent system may be developed independently, they may commit to different ontologies to model the same domain. Whilst 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.

In order to illustrate the matching problem, let us consider an e-Commerce marketplace, where two agents, a $buyer$ and a $seller$, need to negotiate the price of a digital camera. Before starting the negotiation, they need to agree on the vocabulary to be used for exchanging the messages. They use the ontologies o and o', respectively (Figure 1). These ontologies contain subsumption statements (e.g., DigitalCamera $\sqsubseteq$ Product), property specifications (e.g., price domain Product) and instance descriptions (e.g., ThisCamera price 250$).

Ontology matching is the task of finding correspondences between ontologies. Correspondences express relationships supposed to hold between entities in ontologies, for instance, that an Electronic in one ontology is the same as a Product in another one or that DigitalCamera in an ontology is a subclass of CameraPhoto in another one. In the example above, one of the correspondences expresses an equivalence, while the other one is a subsumption correspondence. A set of correspondences between two ontologies is called an alignment. An alignment may be used, for instance, to generate query expressions that automatically translate instances of these ontologies under an integrated ontology or to translate queries with respect to one ontology in to query with respect to the other.

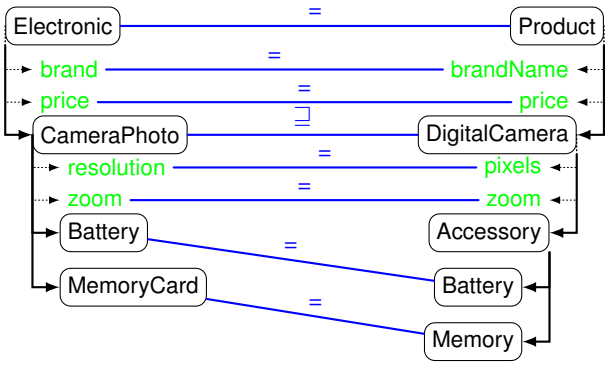

Fig. 1 Fragments of ontologies o and o' with alignment A.

Matching determines an alignment A' for a pair of ontologies o and o'. There are other parameters that can extend the definition of the matching process, namely: (i) the use of an input alignment A, which is to be completed by the process; (ii) the matching parameters, for instance, weights, thresholds, etc.; and (iii) external resources used by the matching process, for instance, common knowledge and domain specific thesauri.

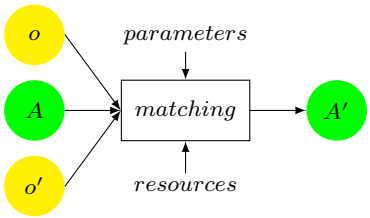

Fig. 2 The ontology matching process (from [15]).

Each of the elements featured in this definition can have specific characteristics which influence the difficulty of the matching task. As depicted in Figure 2, the matching process receives as input three main parameters: the two ontologies to be matched (o and o') and the input alignment (A). The input ontologies can be characterized by the input languages they are described (e.g., OWL-Lite, OWL-DL, OWL-Full), their size (number of concepts, properties and instances) and complexity, which indicates how deep is the hierarchy structured and how dense is the interconnection between the ontological entities. Other properties such as consistency, correctness and completeness are also used for characterizing the input ontologies. The input alignment (A) is mainly characterized by its multiplicity (or cardinality, e.g., how many entities of one ontology can correspond to one entity of the others) and coverage in relation to the ontologies to be matched. In a simple scenario, the input alignment is empty. Regarding the parameters, some systems take advantage of external resources, such as WordNet, sets of morphological rules or previous alignments of general purpose (Yahoo and Google catalogs, for instance).

Different approaches to the problem of ontology matching have emerged from the literature [15]. 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. *Terminological* methods lexically compare strings (tokens or n-grams) used in naming entities (or in the labels and comments concerning entities), whereas *semantic* methods utilise 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, many ontology matching systems rely not on a single approach.

The output **alignment** A' is a set of correspondences between o and o'. Generally, correspondences express a relation r between ontology entities e and e' with a confidence measure n. These are abstractly defined in [15]. In this chapter, we will restrict the discussion to simple correspondences.

Definition 1 (Simple correspondence). Given two ontologies, o and o', a simple correspondence is a quintuple:

$$\langle id, e, e', r, n \rangle,$$

such that:

- id is a URI identifying the given correspondence;
- e and e' are named ontology entities, i.e., named classes, properties, or instances;
- r is a relation among equivalence ($\equiv$), more general ($\sqsupseteq$), more specific ($\sqsubseteq$), and disjointness ($\perp$);
- n is a number in the $[0, 1]$ range.

The correspondence $\langle id, e, e', n, r\rangle$ asserts that the relation r holds between the ontology entities e and e' with confidence n. The higher the confidence value, the higher the likelihood that the relation holds. Alignments may have different cardinalities; 1:1 (one-to-one), 1:m (one-to-many), n:1 (many-to-one) or n:m (many-to-many). An alignment is a 1:1 alignment, if and only if no two different entities in one of the ontologies are matched to the same entity in the other ontology.

Mechanisms that facilitate the construction of alignments require access to both ontologies. Whilst it may be desirable to embed such mechanisms within agents that operate in transparent and collaborative environments, exposing one's ontology may not always be desirable in competitive or adversarial environments, as this may allow other agents to infer, and exploit this knowledge in subsequent negotiations. In addition, creating alignments can be costly, and thus the ability to cache or save previously generated alignments (possibly generated by trusted third parties) may be desirable. Thus, agents may rely on an external **alignment service**.

For example, the *Alignment server*, built on the Alignment API [13], provides functionality to facilitate ontology matching, as well as storing and retrieving alignments. In addition, it can provide assistance to agents when attempting to determine relationships between their ontologies, so that they can understand and interpret each other's messages. An agent plug-in has been developed to allow agents based on the JADE/FIPA ACL (*Agent Communication Language*) to interact with the server in order to retrieve alignments.

Such a service can provide alignments over which the agents will argue in order to choose the more suitable correspondences. Alignments, and the correspondences within such alignments, can be better qualified, through the inclusion of *metadata*, which may refer to the provenance and origin of alignments, confidence ratings, and the original purposes for which they were created. Other metadata may also include any manual (human-based) checks or endorsements provided by some authority. This type of metadata is used, for instance, by *Bioportal* [26], which is an alternative alignment web-service, where users can select correspondences based on providence-based alignment metadata.

2.2 Argumentation frameworks

Argumentation is a decentralised, peer-based negotiation model for reasoning based on the construction and comparison of arguments. The central notion in argumentation systems is the notion of *acceptability*. Different argumentation frameworks have been specified presenting different notions of acceptability. The classical argumentation framework (AF) was proposed by Dung [12], whose notion of acceptability defines that an argument should be accepted only if every attack on it is attacked by an accepted argument. Dung defines an argumentation framework as follows:

Definition 2 (Argumentation Framework [12]). An Argumentation Framework (AF) is a pair $\langle \mathcal{A}, \ltimes \rangle$, such that $\mathcal{A}$ is a set of arguments and $\ltimes$ (attacks) is a bi-

nary relation on $\mathcal{A}$. $a \ltimes b$ means that the argument a attacks the argument b. A set of arguments S attacks an argument b iff b is attacked by an argument in S.

The key question about the framework is whether a given argument $a \in \mathcal{A}$ should be accepted or not. Dung proposes that an argument should be accepted only if every attack on it is attacked by an accepted argument. This notion then leads to the definition of acceptability (for an argument), admissibility (for a set of arguments) and preferred extension:

Definition 3 (Acceptable argument [12]). An argument $a \in \mathcal{A}$ is *acceptable* with respect to set arguments S, noted $acceptable(a, S)$, iff $\forall x \in \mathcal{A}, (x \ltimes a \longrightarrow \exists y \in S, y \ltimes x)$.

Definition 4 (Conflict-free set [12]). A set S of arguments is *conflict-free* iff $\neg\exists x, y \in S, x \ltimes y$. A conflict-free set of arguments S is *admissible* iff $\forall x \in S, acceptable(x, S)$.

Definition 5 (Preferred-extension [12]). A set of arguments S is a *preferred extension* iff it is a maximal (with respect to inclusion set) admissible set of $\mathcal{A}$.

Thus, a preferred extension represents a consistent position within an argumentation framework, which defends itself against all attacks and cannot be extended without raising conflicts.

In Dung's framework, all arguments have equal strength, and therefore attacks always succeed. This is reasonable when dealing with deductive arguments, but in many domains, arguments may lack some coercive force: they provide reasons which may be more or less persuasive. For that purpose, preference-based argumentation has been designed [2] which assigns preferences to arguments, so that preferred arguments would successfully attack less preferred ones (but not vice versa). Bench-Capon [6] went one step further with the *Value Based Argumentation framework* (VAF[1]), which assigns to arguments the values they promote. Agents are distributed among different *audiences* which ascribe different preferences to such values. Hence, different audiences will have different preferences among the arguments and similarly, successful attacks for an audience are those made by arguments of highest values to the audience.

Definition 6 (Value-based AF [6]). A Value-based Argumentation Framework (VAF) is a quintuple $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq \rangle$ such that $\langle \mathcal{A}, \ltimes \rangle$ is an argumentation framework, $\mathcal{V}$ is a nonempty set of values, $v : \mathcal{A} \rightarrow \mathcal{V}$, and $\succeq$ is the preference relation over $\mathcal{V}$ ($v_1 \succeq v_2$ means that, in this framework, v_1 is preferred over v_2).

To each audience, α corresponds a value-based argumentation framework VAF_α such that $v_1 \succeq_\alpha v_2$ states that audience α prefers v_1 over v_2. Attacks are then deemed successful, based on the preference ordering on the arguments' values. This leads to re-defining the notions seen previously:

[1] We describe here as VAF what [6] calls an audience-specific value-based argumentation framework, but the result is equivalent.

Definition 7 (Successful attack [6]). In a value-based argumentation framework, $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq \rangle$, an argument $a \in \mathcal{A}$ defeats (or successfully attacks) an argument $b \in \mathcal{A}$, noted $a \dagger b$, iff both $a \ltimes b$ and $v(b) \not\succeq v(a)$.

Definition 8 (Conflict-free set [6]). A set S of arguments is *conflict-free* for an audience α iff $\forall x, y \in S,\ \neg(x \ltimes y) \vee v(y) \succeq_\alpha v(x)$.

Acceptable arguments and preferred extensions are defined as before. In order to determine preferred extensions with respect to a value ordering promoted by distinct audiences, *objective* and *subjective* acceptance are defined. An argument is *subjectively acceptable* if and only if it appears in some preferred extension for some specific audience. An argument is *objectively acceptable* if and only if it appears in all preferred extension for every specific audience.

3 Argumentation Frameworks for Alignment Agreement

In alignment agreement, arguments can be seen as positions that support or reject correspondences. Such arguments interact following the notion of attack and are selected according to the notion of acceptability. Argumentation frameworks for alignment agreement redefine the notion of acceptability, taking into account the confidence of the correspondences and the number of agents agreeing on a correspondence. In this section we first introduce the general definition of argument, which will be extended according to the scenario where argumentation is used (§4), and then we present the argumentation frameworks.

3.1 Arguments on correspondences

The different approaches presented below all share the same notion of correspondence argument originally defined in [22]. The general definition of correspondence argument is as follows:

Definition 9 (Argument [22]). An argument $a \in AF$ is a tuple $a = \langle c, v, h \rangle$, such that c is a correspondence $\langle e, e', r, n \rangle$; $v \in \mathcal{V}$ is the value of the argument and h is one of $\{+, -\}$ depending on whether the argument is that c does or does not hold.

In this definition, the set of considered values may be based on: the types of matching techniques that agents tend to prefer; the type of targeted applications; information about various level of endorsement of these correspondences, and whether or not they have been checked manually. Thus, any type of information which can be associated with correspondences (see §2.1) may be used. For example, an alignment may be generated for the purpose of information retrieval; however, this alignment may not be suitable for an agent performing a different task requiring

more precision. This agent may therefor prefer the correspondences generated by a different agent for web service composition. Likewise, another agent may prefer human curated alignments rather than alignments generated on the fly.

Arguments interact based on the notion of attack relation:

Definition 10 (Attack [22]). An argument $\langle c, v, h\rangle \in \mathcal{A}$ attacks another argument $\langle c', v', h'\rangle \in \mathcal{A}$ iff $c = c'$ and $h \neq h'$.

Therefore, if $a = \langle c, v_1, +\rangle$ and $b = \langle c, v_2, -\rangle$, $a \ltimes b$ and vice-versa (b is the counter-argument of a, and a is the counter-argument of b).

3.2 Strength-based argumentation framework (SVAF)

Bench-Capon's framework acknowledges the importance of preferences when considering arguments. However, within the specific context of ontology matching, an objection can still be raised regarding the lack of complete mechanisms for handling persuasiveness. Indeed, many ontology matchers generate correspondences with a strength that reflects the confidence they have in the similarity between the two entities. These confidence levels are usually derived from similarity assessments made during the matching process, e.g., from the edit distance measure between labels, or overlap measure between instance sets, and thus are often based on objective grounds. In order to represent arguments with *strength*, reflecting this confidence in a correspondence, [34] proposed the *Strength-based Argumentation Framework (SVAF)*, extending Bench-Capon's *VAF* by redefining the notion of acceptability.

Definition 11 (SVAF [34]). A strength-based argumentation framework (SVAF) is a sextuple $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq, s\rangle$ such that $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq\rangle$ is a value-based argumentation framework and $s : \mathcal{A} \rightarrow [0, 1]$ represents the strength of the argument.

As in value-based argumentation frameworks, each audience α is associated with its own framework in which only the preference relation $\succeq_\alpha$ differs. In order to accommodate the notion of *strength*, the notion of *successful attack* is extended:

Definition 12 (Successful attack [34]). In a strength-based argumentation framework $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq, s\rangle$, an argument $a \in \mathcal{A}$ *successfully attacks* (or *defeats*, noted $a\dagger b$) an argument $b \in \mathcal{A}$ iff

$$a \ltimes b \wedge (s(a) > s(b) \vee (s(a) = s(b) \wedge v(a) \succeq v(b)))$$

3.3 Voting-based argumentation framework (VVAF)

The frameworks described so far assume that candidate correspondences between two entities may differ due to the approaches used to construct them, and thus

these argumentation frameworks provide different mechanisms to identify correspondences generated using approaches acceptable to both agents. However, different alignment generators may often utilise the same approach for some correspondences, and thus the approach used for that correspondence may be significant. Some large-scale experiments involving several matching tools (e.g. the OAEI 2006 Food track campaign [14]) have demonstrated that the more often a given approach for generating a correspondence is used, the more likely it is to be valid. Thus, the SVAF was adapted and extended in [19], to take into account the level of consensus between the sources of the alignments, by introducing the notions of support and voting into the definition of successful attacks. Support enables arguments to be counted as defenders or co-attackers during an attack:

Definition 13 (VVAF [19]). A voting-based argumentation framework (VVAF) is a septuple $\langle \mathcal{A}, \ltimes, \mathcal{S}, \mathcal{V}, v, \succeq, s \rangle$ such that $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq, s \rangle$ is a SVAF, and $\mathcal{S}$ is a (reflexive) binary relation on $\mathcal{A}$, representing the support relation between arguments. $\mathcal{S}(x, a)$ means that the argument x supports the argument a (i.e., they have the same value of h). $\mathcal{S}$ and $\ltimes$ are disjoint relations.

A simple voting mechanism (e.g. plurality voting) can be used to determine the success of a given attack, based upon the number of supporters for a given approach.

Definition 14 (Successful attack [19]). In a VVAF $\langle \mathcal{A}, \ltimes, \mathcal{S}, \mathcal{V}, v, \succeq, s \rangle$, an argument $a \in \mathcal{A}$ successfully attacks (or defeats) an argument $b \in \mathcal{A}$ (noted $a \dagger b$) iff

$$a \ltimes b \wedge (|\{x|\mathcal{S}(x,a)\}| > |\{y|\mathcal{S}(y,b)\}| \vee |\{x|\mathcal{S}(x,a)\}| = |\{y|\mathcal{S}(y,b)\}| \wedge v(a) \succeq v(b)).$$

This voting mechanism is based on simple counting. As some ontology matchers include confidence values with correspondences, a voting mechanism can exploit this confidence value, for example by simply calculating the total confidence value of the supporting arguments. However, this relies on the questionable assumption that all values are equally scaled (as is the case with the SVAF). In [19], a voting framework that normalised these confidence values (i.e. strengths) was evaluated, but was inconclusive. Another possibility would be to rely on a deeper justification for correspondences and to have only one vote for each justification. Hence, if several matchers considered two concepts to be equivalent because WordNet considers their identifier as synonyms, this would be counted only once.

4 Argumentation over Alignments

The use of argumentation has been exploited in two different scenarios presented below. In the first, agents attempt to construct mutually acceptable alignments based on existing correspondences to facilitate communication, based on their alignment preferences (which may be task specific). They therefore argue directly over candidate correspondences provided by an alignment service, with each agent specifying an ordered preference of correspondence types and confidence thresholds.

The second scenario focuses on the consensual construction of alignments involving several agents, each of which specialises in constructing correspondences using different approaches. These matching agents generate candidate correspondences and attempt to combine these to produce a new alignment through argumentation. Thus, whilst the first scenario utilises argumentation as a negotiating mechanism to find a mutually acceptable alignment between transacting agents, this latter scenario could be viewed as offering a service for negotiating alignments.

4.1 Argumentation over alignments for communication in multi-agent systems

4.1.1 Meaning-based argumentation

Laera et al. proposed the meaning-based argumentation approach [22, 23, 21], to allow agents to propose, attack, and counter-propose candidate correspondences according to the agents' preferences, in order to identify mutually acceptable alignments. Their approach utilises Bench-Capon's VAF [6] to support the specification of preferences of correspondent types (as discussed in §2.1) within each argument. Thus, when faced with different, candidate correspondences who's type differ, each agents' preference ordering can be considered when determining if an argument for one correspondence will successfully attack another. Different audiences therefore represent different sets of arguments for preferences between the categories of arguments (identified in the context of ontology matching).

Each agent is defined as follows:

Definition 15 (Agent). An agent Ag_i is characterised by a tuple $\langle O_i, F, \epsilon_i \rangle$, such that O_i is the ontology used by the agent, F is its (valued-based) argumentation framework, and ϵ_i is the private threshold value.

Candidate correspondences are retrieved from an alignment service (see §2.1) which also provides the justifications G (described below) for each correspondence, based on the approach used to construct the correspondence. The agents use this information to exchange arguments supplying the reasons for their choices. In addition, as these grounds include a confidence value associated with each correspondence, each agent utilises a private threshold value ϵ to filter out correspondences with low confidence values[2]. This threshold, together with the pre-ordering of preferences, are used to generate arguments for and against a correspondence. It extends the notion of argument presented in §3.1:

Definition 16 (Argument [22]). An argument is a triple $\langle G, c, h \rangle$, where c is a correspondence $\langle e, e', r, n \rangle$, G is the grounds justifying a prima facie belief that the cor-

[2] The use of confidence profiles has since been explored to specify correspondence-type specific thresholds, resulting in the agreement over a greater diversity of agreed correspondences, and consequently more inclusive alignments [9].

respondence does, or does not hold; and h is one of $\{+, -\}$ depending on whether the argument is that c does or does not hold.

The grounds G justifying a correspondence between two entities are based on the five categories of correspondence types (as discussed in §2.1) - namely *Semantic* (S), *Internal Structural* (IS), *External Structural* (ES), *Terminological* (T), and *Extensional* (E). These classes are used as types for the values $\mathcal{V}$, i.e., $\mathcal{V} = \{M, IS, ES, T, E\}$, that are then used to construct an agent's partially-ordered preferences, based on the agents ontology and task. Thus, an agent may specify a preference for terminological correspondences over semantic correspondences if the ontology it uses is mainly taxonomic, or vice versa if the ontology is semantically rich. Preferences may also be based on the type of task being performed; extensional correspondences may be preferred when queries are about instances that are frequently shared. The pre-ordering of preferences $\succeq$ for each agent Ag_i is over $\mathcal{V}$, corresponding to the specification of an audience. Specifically, for each candidate correspondence c, if there exists one or more justifications G for c that corresponds to the highest preferences $\succeq$ of Ag_i (with the respect of the pre-ordering), assuming n is greater than its private threshold ϵ, an agent Ag_i will generate arguments $x = (G, c, +)$. If not, the agent will generate arguments against: $x = (G, c, -)$. The arguments interact based on the notion of attack, as specified in §3.1.

The argumentation process takes four main steps: (i) each agent Ag_i constructs an argumentation framework VAF_i by specifying the set of arguments and the attacks between them; (ii) each agent Ag_i considers its individual frameworks VAF_i with all the argument sets of all the other agents and then extends the attack relations by computing the attacks between the arguments present in its framework with the other arguments; (iii) for each VAF_i, the arguments which are undefeated by attacks from other arguments are determined, given a value ordering – the global view is considered by taking the union of these preferred extensions for each audience; and (iv) the arguments in every preferred extension of every audience are considered – the correspondences that have only arguments for are included in the a set called *agreed alignments*, whereas the correspondences that have only arguments against them are rejected, and the correspondences which are in some preferred extension of every audience are part of the set called *agreeable alignments*.

The dialogue between agents consists of exchanging sets of arguments and the protocol used to evaluate the acceptability of a single correspondence is based on a set of speech acts ($Support$, $Contest$, $Withdraw$). For instance, when exchanging arguments, an agent sends $Support(c, x_1)$ for supporting a correspondence c through the argument $x_1 = (G, c, +)$ or $Contest(c, x_2)$ for rejecting c, by $x_2 = (G, c, -)$. If the agents do not have any arguments or counter-arguments to propose, then they send $Withdraw(c)$ and the dialogue terminates.

To illustrate this approach, consider the two agents buyer b and seller s, using the ontologies in Figure 1. First, the agents access the alignment service that returns the correspondences with the respective justifications:

- m_1: $\langle zoom_o, zoom_{o'}, \equiv, 1.0\rangle$, with $G = \{T, ES\}$
- m_2: $\langle Battery_o, Battery_{o'}, \equiv, 1.0\rangle$, with $G = \{T\}$

- m_3: $\langle MemoryCard_o Memory_{o'}, \equiv, 0.54\rangle$, with $G = \{T\}$
- m_4: $\langle brand_o, brandName_{o'}, \equiv, 0.55\rangle$, with $G = \{T, ES\}$
- m_5: $\langle price_o, price_{o'}, \equiv, 1.0\rangle$, with $G = \{T, ES\}$
- m_6: $\langle CameraPhoto_o, DigitalCamera_{o'}, \equiv, 1.0\rangle$, with $G = \{ES\}$
- m_7: $\langle resolution_o, pixels_{o'}, \equiv, 1.00\rangle$, with $G = \{ES\}$

Agent b selects the audience R_1, which prefers terminology to external structure ($T \succ_{R_1} ES$), while s prefers external structure to terminology ($ES \succ_{R_2} T$). All correspondences have a degree of confidence n that is above the threshold of each agent and then all of them are taken into account. Both agents accept m_1, m_4 and m_5. b accepts m_2, m_3, while s accepts m_6 and m_7. Table 1 shows the arguments and corresponding attacks.

Table 1 Arguments and attacks.

id	argument	attack	agent
A	$\langle T, m_1, +\rangle$		b, s
B	$\langle ES, m_1, +\rangle$		b, s
C	$\langle T, m_2, +\rangle$	D	b
D	$\langle ES, m_2, -\rangle$	C	s
E	$\langle T, m_3, +\rangle$	F	b
F	$\langle ES, m_3, -\rangle$	E	s
G	$\langle T, m_4, +\rangle$		b, s

id	argument	attack	agent
H	$\langle ES, m_4, +\rangle$		b, s
I	$\langle T, m_5, +\rangle$		b, s
J	$\langle ES, m_5, +\rangle$		b, s
L	$\langle ES, m_6, +\rangle$	M	s
M	$\langle T, m_6, -\rangle$	L	b
N	$\langle ES, m_7, +\rangle$	O	s
O	$\langle T, m_7, -\rangle$	N	b

The arguments A, B, G, H, I, and J are not attacked and then are acceptable for both agents (they form the *agreed alignment*). The arguments C and D are mutually attacked and are acceptable only in the corresponding audience, i.e., C is acceptable for the audience b and D is acceptable for the audience s. The same occurs for the arguments E, F, L, M, M, and O. The correspondences in such arguments are seen as the *agreeable alignments*.

4.1.2 The approach by Trojahn and colleagues

In order to provide translations between messages in agent communication, [33] formally defines an alignment as a set of correspondences between *queries* over ontologies. The alignment is obtained by specialised matcher agents that argue in order to agree on a globally acceptable alignment. The set of acceptable arguments is then represented as conjunctive queries in OWL-DL [18].

A conjunctive query has the form $\bigwedge$ ($\mathrm{P}_i(\mathrm{s}_i)$), where each $\mathrm{P}_i(\mathrm{s}_i)$ represents a correspondence. For instance, $\langle CameraPhoto_o, DigitalCamera_{o'}, \equiv, 1.0\rangle$ is represented as $Q(x) : CameraPhoto(x) \equiv DigitalCamera(x)$.

Consider the example where the agents "buyer b" and "seller s" interact to agree on the price of a digital camera, using the ontologies o and o' of Figure 1, respectively. Before the agents can agree on the price, they need to agree on the terms used to communicate to each other. This task can be delegated to a matcher agent m,

that receives the two ontologies and sends them to an argumentation module. This module, made up of different specialised agents $a_1, ..., a_n$ (which can be distributed on the web), receives the ontologies and returns a set of DL queries representing the acceptable correspondences. These interactions are loosely based on the Contract Net Interaction Protocol [16]. The argumentation process between the specialised matchers is detailed in Section 4.2. Table 2 describes the steps of the interaction between the agents.

Table 2 Interaction steps [33].

Step	Description
1	Matcher agent m requests the ontologies to be matched to agents b and s
2	Ontologies are sent from m to the argumentation module
3	Matchers $a_1, ..., a_n$ apply their algorithms
4	Each matcher a_i communicate with each others to exchange their arguments
5	Preferred extensions of each a_i are generated
6	Objectively acceptable arguments o are computed
7	Correspondences in o are represented as conjunctive queries
8	Queries are sent to m
9	Queries are sent from m to b and s
10	Agents b and s use the queries to communicate with each other

In fact, only one of the agents should receive the DL queries, which should be responsible for the translations. We consider that the set of objectively acceptable arguments has the correspondences shown in Figure 3, with the respective queries.

Query ID	Correspondences
$Q_{b\,1}$(x)	b:CameraPhoto(x)
$Q_{s\,1}$(x)	s:DigitalCamera(x)
m_1	$Q_{b\,1} \equiv Q_{s\,1}$
$Q_{b\,2}$(y)	b:zoom(y)
$Q_{s\,2}$(y)	s:zoom(y)
m_2	$Q_{b\,2} \equiv Q_{s\,2}$
$Q_{b\,3}$(y)	b:resolution(y)
$Q_{s\,3}$(y)	s:pixels(y)
m_3	$Q_{b\,3} \equiv Q_{s\,3}$

Fig. 3 Conjunctive queries.

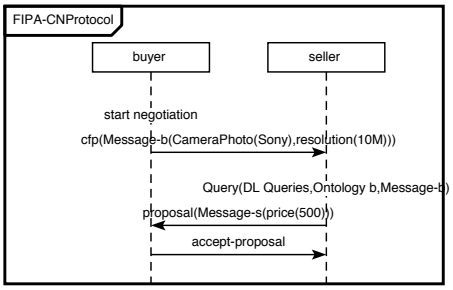

Fig. 4 Interaction between buyer and seller agents.

Figure 4.1.2 shows an AUML[3] interaction diagram with the messages exchanged between the agents b and s during the negotiation of the price of the camera. The agents use the queries to search for correspondences between the messages sent from each other and the entities in the corresponding ontologies. In the example, the agent b sends a message to the agent s, using its vocabulary. Then, the agent s converts the message, using the DL queries.

[3] AUML – Agent Unified Modelling Language [17].

4.1.3 Reducing the argumentation space through modularization

Doran et al. [11] utilised modularization to identify the ontological descriptions relevant to the communication, and consequently reduce the number of correspondences necessary to form the alignment. The use of argumentation can be computationally costly, as the complexity can reach $\Pi_2^{(p)}$-complete in some cases. Thus, by reducing the number of arguments, the time required for generating the alignments can be significantly reduced; even when taking into account the time necessary for the modularization process itself. In an empirical study, the authors found that the use of modularization significantly reduced the average number of correspondences presented to the argumentation framework, and hence the size of the search space – in some cases by up to 97%, across a number of different ontology pairs. They also noted that three patterns emerged: i) where no reduction in size occurred (in 4.84% of cases within the study); ii) where the number of correspondences was reduced (55.14%); and iii) where modules of size zero were found (40.02%), corresponding to failure scenarios; i.e. where the subsequent transaction would fail due to insufficient alignment between the ontologies.

An ontology modularization technique extracts a consistent module M from an ontology A that covers a specified signature $Sig(M)$, such as $Sig(M) \subseteq Sig(O)$. M is the part of O that is said to cover the elements defined by $Sig(M)$. The first agent engaging in the communication specifies the $Sig(M)$ of its ontology O where M is an ontology concept relevant for a task. The resulting module contains the entities considered to be relevant for its task, including the subclasses and properties of the concepts in $Sig(M)$. The step-by-step interaction between two agents, following an argumentation based on modularization is presented in Table 3.

Table 3 Ontology modularization and argumentation for alignment agreement [10].

Step	Description
1	Ag_1 asks a query, $query(A \in Sig(O))$ to Ag_2.
2	Ag_2 does not understand the query, $A \notin Sig(O'))$ and informs Ag_1 they need to use a server.
3	Ag_1 produces, $om(O, Sig(A))$, an ontology module, M, to cover the concepts required for its task.
4	Ag_1 and Ag_2 invoke the server. Ag_1 sends its ontology, O and the signature of M, $Sig(M)$.
5	The alignment service aligns the two ontologies and filters the correspondences according to M. Only those features an entity from M are returned to both agents.
6	The agents begin the process of argumentation, with each agent generating arguments and counter-arguments.
7	The iteration terminates when the agents agree on a set of correspondences.
8	Ag_1 asks again Ag_2, using the agreed correspondences, $query(A \in Sig(O) \wedge B \in Sig(O'))$ where A and B are aligned.
9	Ag_2 answers the query using the agreed correspondences.

For communicating, only the initiating agent (Ag_1) is aware of its task and, consequently, which concepts are relevant to this task (Steps 1 and 2). These concepts will be included in $Sig(M)$, the signature of the resulting ontology module (Step 3). The set of candidate correspondences (Step 4) is filtered (Step 5) according to the filtering function $filter()$. $filter$ returns a subset Z of correspondences, where the entities e in these correspondence are in $Sig(M)$. The set Z is then used within the argumentation process. Modularization is therefore used to filter the correspondences that are passed to the argumentation process. The agents then argue (Steps 6-7) to reach an acceptable alignment.

The combination of argumentation and modularization reduces the cost of reaching an agreement over an alignment, by reducing the size of the set of correspondences argued over, and hence the number of arguments required. This greatly contributes to reduce the consumed time, at a minimal expense in accuracy.

Following the example of the buyer and seller agents, the buyer agent knows which concepts will be used for communicating and then a module of the ontology o is extracted containing such concepts (i.e., $CameraPhoto$, $resolution$, $zoom$, and $price$). The buyer agent then filters the correspondences in order to retrieve the subset containing only these concepts.

4.2 Solving conflicts between matcher agents

In [34], alignments produced by different matchers are compared and agreed via an argumentation process. The matchers interact in order to exchange arguments and the SVAF model (§3.2) is used to support the choice of the most acceptable of them. Each correspondence can be considered as an argument because the choice of a correspondence may be a reason against the choice of another correspondence. Correspondences are represented as arguments, extending the notion of argument specified in §3.1:

Definition 17 (Argument). An argument $x \in AF$ is a tuple $x = \langle c, v, s, h \rangle$, such that c is a correspondence $\langle e, e', r, n \rangle$; $v \in \mathcal{V}$ is the value of the argument; s is the strength of the argument, from n; and h is one of $\{+, -\}$ depending on whether the argument is that c does or does not hold.

The matchers generate arguments representing their alignments following a *negative arguments as failure* strategy. It relies on the assumption that matchers return complete results. Each possible pair of ontology entities which is not returned by the matcher is considered to be at risk, and a negative argument is generated ($h = -$).

The values v in $\mathcal{V}$ correspond to the different matching approaches and each matcher m has a preference ordering $\succeq_m$ over $\mathcal{V}$ such that its preferred values are those it associates to its arguments. For instance, consider $\mathcal{V}$= $\{l, s, w\}$, i.e., *lexical*, *structural* and *wordnet-based* approaches, respectively, and three matchers m_l, m_s and m_w, using such approaches. The matcher m_l has as preference order $l \succeq_{m_l} s \succeq_{m_l} w$. The basic idea is to obtain a consensus between different matchers,

represented by different preferences between values. Arguments interact based on the notion of attack presented in §3.1.

The argumentation process can be described as follows. First, each matcher generates its set of correspondences, using some specific approach and then the set of corresponding arguments is generated. Next, the matchers exchange with each others their set of arguments – the dialogue between them consists of the exchange of individual arguments. When all matchers have received the set of arguments of each others, they instantiate their SVAFs in order to generate their set of acceptable correspondences. The consensual alignment contains the correspondences represented as arguments that appear in every set of acceptable arguments, for every specific audience (objectively acceptable).

In order to illustrate this process, consider two matchers, m_l (lexical) and m_s (structural), trying to reach a consensus on the alignment between the ontologies in Figure 1. m_l uses an edit distance measure to compute the similarity between labels of concepts and properties of the ontologies, while m_s is based on the comparison of the direct super-classes of the classes or classes of properties. Table 4 shows the correspondences and arguments generated by each matcher. The matchers generate complete alignments, i.e., if a correspondence is not found, an argument with value of $h = -$ is created. It includes correspondences that are not relevant to the task at hand. For the sake of brevity, we show only the arguments with $h = +$ and the corresponding counter-arguments (Table 5). We consider 0.5 as the confidence level c for negative arguments ($h = -$). Considering $\mathcal{V} = \{l, v\}$, m_l associates to its arguments the value l, while m_s generates arguments with value s. m_l has as preference ordering: $l \succ_{m_l} s$, while m_s has the preference: $s \succ_{m_s} l$.

Table 4 Correspondences and arguments generated by m_l and m_s.

id	correspondence	argument	matcher
A	$c_{l,1} = \langle zoom_o, zoom_{o'}, \equiv, 1.0\rangle$	$\langle c_{l,1}, l, 1.0, +\rangle$	m_l
B	$c_{l,2} = \langle Battery_o, Battery_{o'}, \equiv, 1.0\rangle$	$\langle c_{l,2}, l, 1.0+\rangle$	m_l
C	$c_{l,3} = \langle MemoryCard_o Memory_{o'}, \equiv, 0.33\rangle$	$\langle c_{l,3}, l, 0.33, +\rangle$	m_l
D	$c_{l,4} = \langle brand_o, brandName_{o'}, \equiv, 0.22\rangle$	$\langle c_{l,4}, l, 0.22, +\rangle$	m_l
E	$c_{l,5} = \langle price_o, price_{o'}, \equiv, 1.0\rangle$	$\langle c_{l,5}, l, 1.0, +\rangle$	m_l
F	$c_{s,1} = \langle CameraPhoto_o, DigitalCamera_{o'}, \equiv, 1.0\rangle$	$\langle c_{s,1}, s, 1.0, +\rangle$	m_s
G	$c_{s,2} = \langle zoom_o, zoom_{o'}, \equiv, 1.0\rangle$	$\langle c_{s,2}, s, 1.0, +\rangle$	m_s
H	$c_{s,3} = \langle brand_o, brandName_{o'}, \equiv, 1.0\rangle$	$\langle c_{s,3}, s, 1.0, +\rangle$	m_s
I	$c_{s,4} = \langle resolution_o, pixels_{o'}, \equiv, 1.0\rangle$	$\langle c_{s,4}, s, 1.0, +\rangle$	m_s
J	$c_{s,5} = \langle price_o, price_{o'}, \equiv, 1.0\rangle$	$\langle c_{s,5}, s, 1.0, +\rangle$	m_s

Having their arguments $\mathcal{A}$, the matchers exchange them. m_l sends to m_s its set of arguments $\mathcal{A}_l$ and vice-versa. Next, based on the attack notion, each matcher m_i generates its attack relation $\ltimes_i$ and then instantiates its $SVAFs_i$. The arguments A, D, E, G, H and J are acceptable in both SVAFs (they are not attacked by counter-arguments with $h = -$). F, I, and B ($h = +$) successfully attack their counter-arguments ($h = -$) L, M and N, respectively, because they have high-

Table 5 Counter-arguments (attacks) for the arguments in Table 4.

id	correspondence	counter-argument	matcher
L	$c_{l,6} = \langle CameraPhoto_o, DigitalCamera_{o'}, \equiv, 0.5\rangle$	$\langle c_{l,6}, l, 0.5, -\rangle$	m_l
M	$c_{l,7} = \langle resolution_o, pixels_{o'}, \equiv, 0.5\rangle$	$\langle c_{l,7}, l, 0.5, -\rangle$	m_l
N	$c_{s,6} = \langle Battery_o, Battery_{o'}, \equiv, 0.5\rangle$	$\langle c_{s,6}, s, 0.5, -\rangle$	m_s
O	$c_{s,7} = \langle MemoryCard_o, Memory_{o'}, \equiv, 0.5\rangle$	$\langle c_{s,7}, s, 0.5, -\rangle$	m_s

est confidence in their correspondences. C ($h = +$) is successfully attacked by its counter-argument O.

The arguments in the preferred extension of both matchers m_l and m_s are: A, D, E, F, G, H, J, F, I, B and O. While $\langle resolution_o, pixels_{o'}, \equiv, 1.0\rangle$, $\langle Battery_o, Battery_{o'}, \equiv, 1.0\rangle$ and $\langle CameraPhoto_o, DigitalCamera_{o'}, \equiv, 1.0\rangle$ have been accepted, $\langle MemoryCard_o, Memory_{o'}, \equiv, 0.33\rangle$ has been discarded.

5 Weaknesses and Challenges

As discussed above, argumentation for alignment agreement has been exploited in different ways, for different scenarios. However, there are still various challenges ahead for achieving a fully satisfying approach. We briefly consider some of them.

Confidence of arguments

In [34], the notion of attack between the arguments highly depends on the confidence associated to the correspondences. Such confidence levels are usually derived from similarity assessments made during the matching process, e.g., from edit distance measure between labels, or overlap measure between instance sets. However, there is no objective theory nor even informal guidelines for determining such confidence levels. Using them to compare results from different matchers is therefore questionable especially because of potential scale mismatches. For example, a same strength of 0.8 may not correspond to the same level of confidence for two different matchers.

Complete alignments

Generating complete alignments is at first sight quite unrealistic, but it can nevertheless be supported by the observation that most matchers try to provide as much correspondences as possible. However, dealing with a large number of arguments can become prohibitively costly. Following the approach from [10], the search space within the argumentation process can be reduced, by isolating only the correspondences that are relevant to the communication. Other authors isolate the subpart of the ontologies to be matcher relevant for the communication before matching only these pieces of ontologies instead of the whole ontologies [27]. These approaches have to be developed with guarantees that the isolated items are the relevant ones.

Inconsistent alignments

An important issue in such argumentation for alignment agreement is related to the potential inconsistency in the agreed alignment. Indeed, even if the initial alignments are consistent, selected sets of correspondences may generate concepts that are not satisfiable.

Solving the inconsistency problem in alignments has two possible alternatives:

- express the inconsistency within the argumentation framework [1];
- deal alternatively with the logical and argumentative part of the problem.

Integrating the logics within the argumentation framework seems a more elegant solution and it can be achieved straightforwardly when correspondences are arguments and incompatible correspondences can mutually attack each others. However, this works only when two correspondences are incompatible. When the set of incompatible correspondences is larger, the encoding is not so straightforward and may lead to the generation of an exponential amount of argument and attack relations. On the other side, alternating logical and argumentative treatments may also lead to prohibitive computational costs.

In this case, the solution seems to be a trade-off between the computational costs and the expected consistency.

Availability of justifications

The presented approaches argue for or against a correspondence based on justifications for the arguments. They are thus highly dependent on justifications for the arguments provided with the alignments. Although, alignment servers provide the necessary metadata for storing such justification with alignments (see §2.1), it is not common for people or for matchers to provide this information.

Ideally, matchers should provide such justifications, as a way to understand why a particular alignment is found or why a certain match is ranked higher than another. However, this is not common practice.

The development of such methods may therefore be slowed by the unavailability of justification metadata. It seems necessary to provide incentive to both automatic and manual matchers to generate this information. One such incentive could be, of course the ability to be involved in an argumentation process and then to provide better alignments. Another incentive would be to better help explain matcher results to users [30].

6 Other Related Work

This chapter has covered all the work carried out in the domain of alignment argumentation *per se*. However, in order to find alignments between ontologies used by agents, some work have proposed different techniques that we consider here.

[31] has proposed alignment negotiation to establish a consensus between different agents using the MAFRA alignment framework [24]. The approach is based on

utility functions used to evaluate the confidence in a particular correspondence in the context of each agent. These confidence values are combined in order to decide if the correspondence is accepted, rejected or need to be negotiated. A meta-utility function is also applied to evaluate if the effort necessary to negotiate is beneficial or not; it may so automatically change the thresholds so that some correspondences are directly rejected or accepted. The approach is highly dependent on the MAFRA framework and cannot be directly applied to other environments.

Schemes for obtaining ontology alignments through the working cycles of agents have been developed. They either observe failure or success of the communication and statistically learn the alignments [7] or they use the interaction protocol of each agent for reducing the possible meaning of concepts used as performative [4].

[8] presents an approach for agents to agree on a common ontology in a decentralised way. The approach assumes that each agent adopts a private ontology and shares an intermediate ontology. The private ontology is used for storing and reasoning with operational knowledge, i.e., knowledge relevant to a particular problem or task at hand. The intermediate ontology is used for communication. Communication proceeds by translating from the speaker's private ontology to the intermediate ontology which the hearer translates back again into its own private ontology. The authors show how to establish such an intermediate ontology, which is the common goal for every agent in the system. In the approaches we have presented, on the other hand, the result of the negotiation is a set of correspondences between the terms of the different ontologies.

[5] presents an ontology negotiation protocol to provide semantic interoperability in multi-agent systems in an automated fashion at run-time. The ontology negotiation protocol enables agents to discover ontology conflicts or unknown terms. Then, it goes through (i) incremental interpretations of the unknown terms with the help of external resources, (ii) clarification, by proposing putative correspondences, (iii) evaluation, through the impact of such correspondences on some tasks, and (iv) update of the ontology with the correspondence. The final result of this process is that each agent will converge on a single, shared ontology. In contrast, in the approaches presented in this chapter, agents keep their own ontologies that they have been designed to reason with, whilst generating alignments with other agent's ontologies.

In [25], the authors propose an argumentation framework for inter-agent dialogue to reach an agreement on terminology, which formalizes a debate in which the divergent representations (expressed in description logic) are discussed. The proposed framework is stated as being able to manage conflicts between claims, with different relevancies for different audiences, in order to compute their acceptance. However, no detail is given about how agents will generate such claims.

[32] proposes a cooperative negotiation model, where agents apply individual matching algorithms and negotiate on a final alignment. Basically, the negotiation process involves the exchange of proposal and counter-proposals that represents correspondences. Each correspondence is negotiated individually. Three kinds of agents interact, lexical, structural, and semantic, and the communication is managed by a mediator agent.

7 Final Remarks

This chapter has presented an overview of the approaches for alignment agreement based on argumentation. Such approaches provide a way for agents, with different ontologies, to agree upon mutually acceptable ontology alignments to facilitate communication within a dynamic environment.

We have discussed how two agents commiting to different ontologies can align their ontologies in order to interoperate and how agents relying on different matching approaches can agree on a common alignment. The approaches for both scenarios are not fully satisfying and there are still various challenges ahead for achieving such maturity.

References

1. Amgoud, L., Besnard, P.: Bridging the gap between abstract argumentation systems and logic. In: Proceedings of the 3rd International Conference on Scalable Uncertainty Management, pp. 12–27. Springer-Verlag, Berlin, Heidelberg (2009)
2. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation. In: G. Cooper, S. Moral (eds.) Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence (1998)
3. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence **34**(1-3), 197–215 (2002)
4. Atencia, M.: Semantic alignment in the context of agent interaction. Ph.D. thesis, Universita Autonoma de Catalunya, Barcelona (SP) (2010)
5. Bailin, S.C., Truszkowski, W.: Ontology negotiation between intelligent information agents. Knowledge Engineering Review **17**(1), 7–19 (2002). DOI http://dx.doi.org/10.1017/S0269888902000292
6. Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation **13**(3), 429–448 (2003)
7. Besana, P., Robertson, D.: How service choreography statistics reduce the ontology mapping problem. In: Proceedings 6th International Semantic Web Conference, no. 4825 in Lecture Notes in Computer Science, pp. 44–57 (2007)
8. van Diggelen, J., Beun, R.J., Dignum, F., van Eijk, R.M., Meyer, J.J.: ANEMONE: An effective minimal ontology negotiation environment. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 899–906. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10.1145/1160633.1160794
9. Doran, P., Payne, T.R., Tamma, V., Palmisano, I.: Deciding agent orientation on ontology mappings. In: Proceedings of the 9th International Semantic Web Conference. Shanghai, China (2010)
10. Doran, P., Tamma, V., Palmisano, I., Payne, T.R.: Efficient argumentation over ontology correspondences. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, pp. 1241–1242. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2009)
11. Doran, P., Tamma, V., Payne, T., Palmisano, I.: Dynamic selection of ontological alignments: A space reduction mechanism. In: International Joint Conference on Artificial Intelligence (2009). URL http://www.aaai.org/ocs/index.php/IJCAI/IJCAI-09/paper/view/551
12. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n–person games. Artificial Intelligence **77**(2), 321–357 (1995)

13. Euzenat, J.: An API for ontology alignment. In: Proceedings of the 3rd International Semantic Web Conference, pp. 698–7112. Hiroshima, Japan (2004)
14. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W.R., Yatskevich, M.: Results of the ontology alignment evaluation initiative 2006. In: First International Workshop on Ontology Matching. Athens, GA, US (2006)
15. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (DE) (2007)
16. FIPA: Contract net interaction protocol specification. Tech. Rep. SC00029H, Foundation for Intelligent Physical Agents (2002)
17. FIPA: Modeling: Interaction diagrams. Tech. rep., Foundation for Intelligent Physical Agents (2003)
18. Haase, P., Motik, B.: A mapping system for the integration of OWL-DL ontologies. In: Proceedings of the 1st International Workshop on Interoperability of Heterogeneous Information Systems, pp. 9–16. ACM, New York, NY, USA (2005). DOI http://doi.acm.org/10.1145/1096967.1096970
19. Isaac, A., dos Santos, C.T., Wang, S., Quaresma, P.: Using quantitative aspects of alignment generation for argumentation on mappings. In: P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt (eds.) OM, *CEUR Workshop Proceedings*, vol. 431. CEUR-WS.org (2008)
20. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., Sierra, C.: Automated negotiation: Prospects methods and challenges. Group Decision and Negotiation **10**(2), 199–215 (2001)
21. Laera, L., Blacoe, I., Tamma, V., Payne, T., Euzenat, J., Bench-Capon, T.: Argumentation over ontology correspondences in MAS. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1–8. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1329125.1329400
22. Laera, L., Tamma, V., Euzenat, J., Bench-Capon, T., Payne, T.R.: Reaching agreement over ontology alignments. In: Proceedings of the 5th International Semantic Web Conference, *Lecture Notes in Computer Science*, vol. 4273/2006, pp. 371–384. Springer Berlin / Heidelberg (2006). DOI 10.1007/11926078
23. Laera, L., Tamma, V.A.M., Euzenat, J., Bench-Capon, T.J.M., Payne, T.R.: Agents arguing over ontology alignments. In: B. Dunin-Keplicz, A. Omicini, J.A. Padget (eds.) Proceedings of the 4th European Workshop on Multi-Agent Systems, *CEUR Workshop Proceedings*, vol. 223. CEUR-WS.org (2006)
24. Maedche, A., Motik, B., Silva, N., Volz, R.: MAFRA – a MApping FRAmework for distributed ontologies. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, pp. 235–250. Springer-Verlag, London, UK (2002)
25. Morge, M., Routier, J.C., Secq, Y., Dujardin, T.: A formal framework for inter-agents dialogue to reach an agreement about a representation. In: R. Ferrario, N. Guarino, L. Prevot (eds.) Proceedings of the Workshop on Formal Ontologies for Communicating Agents (2006)
26. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A.D., Chute, C.G., Musen, M.A.: Bioportal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research **37**(Web-Server-Issue), 170–173 (2009)
27. Packer, H., Payne, T., Gibbins, N., Jennings, N.: Evolving ontological knowledge bases through agent collaboration. In: Proceedings 6th European Workshop on Multi-Agent Systems, Bath UK. Springer (2008)
28. Parsons, S., Jennings, N.: Negotiation through argumentation-A preliminary report. In: Proceedings of the 2nd International Conference Multi-Agent Systems, pp. 267–274. Kyoto, Japan (1996)
29. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal Applied Non-Classical Logics **7**(1), 25–75 (1997)
30. Shvaiko, P., Giunchiglia, F., da Silva, P.P., McGuinness, D.L.: Web explanations for semantic heterogeneity discovery. In: A. Gómez-Pérez, J. Euzenat (eds.) Proceedings of the 2nd European Semantic Web Conference, *Lecture Notes in Computer Science*, vol. 3532, pp. 303–317. Springer (2005)

31. Silva, N., Maio, P., Rocha, J.: An approach to ontology mapping negotiation. In: Proceedings of the Third International Conference on Knowledge Capture Workshop on Integrating Ontologies. Banff, Canada (2005)
32. Trojahn, C., Moraes, M., Quaresma, P., Vieira, R.: Using cooperative agent negotiation for ontology mapping. In: Proceedings of the 4th European Workshop on Multi-Agent Systems, *CEUR Workshop Proceedings*, vol. 223, pp. 1–10. CEUR-WS.org (2006)
33. Trojahn, C., Quaresma, P., Vieira, R.: Conjunctive queries for ontology based agent communication in MAS. In: Proceedings of the 7th international joint conference on Autonomous Agents and Multiagent Systems, pp. 829–836. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2008)
34. Trojahn, C., Quaresma, P., Vieira, R., Moraes, M.: A cooperative approach for composite ontology mapping. LNCS Journal on Data Semantic X (JoDS) **4900**(1), 237–263 (2008). DOI 10.1007/978-3-540-77688-8