

# Using Ontology Modularization for Efficient Negotiation over Ontology Correspondences in MAS

Paul Doran<sup>1</sup>, Valentina Tamma<sup>1</sup>, Terry R. Payne<sup>1</sup>, and Ignazio Palmisano<sup>2</sup>

<sup>1</sup> Department of Computer Science, University of Liverpool,  
Liverpool L69 3BX, United Kingdom

{P.Doran,V.Tamma,T.R.Payne}@liverpool.ac.uk

<sup>2</sup> School of Computer Science, University of Manchester, UK  
ignazio.palmisano@cs.manchester.ac.uk

**Abstract.** Efficient agent communication in open and dynamic environments relies on the agents ability to reach a mutual understanding over message exchanges. Such environments are characterized by the existence of heterogeneous agents that commit to different ontologies, with no prior assumptions regarding the use of shared vocabularies. Various approaches have therefore considered how mutually acceptable mappings may be determined dynamically between agents through negotiation. In particular, this paper focusses on the *meaning based negotiation* approach, proposed by Laera *et al* [1], that makes use of argumentation in order to select a set of mappings that is deemed acceptable by both agents. However, this process can be highly complex, reaching  $\Pi_2^{(p)}$ -complete. Whilst it is non-trivial to reduce this complexity, we have explored the use of ontology modularization as a means of reducing the space of possible concepts over which the agents have to negotiate. In this paper, we propose an approach that combines modularization with argumentation to generate focused domains of discourse to facilitate communication. We empirically demonstrate that we can not only reduce the number of alignments required to reach consensus by an average of 75%, but that in 41% of cases, we can identify those agents that would not be able to fully satisfy the request, without the need for negotiation.

## 1 Introduction

The ability to communicate, and hence collaborate, delegate tasks or answer queries is one of the key capabilities of an agent within a Multi-Agent System. This communication can only be facilitated if there is some shared understanding of the messages that the agents exchange. Whilst implicit assumptions regarding the terminology or vocabulary may be acceptable within small, closed environments (where all the agents are known at design time), it becomes imperative to specify explicit vocabularies or ontologies to support communication as environments open up, or the heterogeneity of large systems increases. The use of formally defined ontologies has increased significantly due to the combined emergence of optimized, description-logic reasoners, and standards for representing

ontologies [2]. Thus, agents can share the same representation language, utilize domain models that are semantically rich, and make inferences over queries in a decidable manner. However, this is predicated upon the notion of a single, shared ontology, or a set of *correspondences* or mappings<sup>1</sup> that map semantically related entities from one ontology to another.

The ability to dynamically reconcile heterogeneous ontologies within open environments (where no assumptions are made with respect to the ontologies used) is dependent on an agent's ability to agree on a set of acceptable correspondences between the elements in its ontology with those of another. Various approaches attempt to resolve ontological mismatches within open environments [3,1,4], including negotiation approaches that collaboratively search a space of correspondences to find a mutually acceptable set, and thus facilitate communication. This search can become prohibitively costly as the sizes of the ontologies grow, and thus a reduction of this search space is highly desirable. This can be particularly significant as ontologies may consist of several loosely connected or overlapping sub-domains, many of which may be irrelevant for a given task.

Several approaches for generating alignments (i.e. sets of mappings between ontologies) have emerged [5], that provide different correspondences depending on the method used, the characteristics of the ontologies themselves, and other resources (e.g. knowledge-bases). Thus, as new correspondences are generated, they can be registered with a mapping repository, to facilitate discovery and reuse by other agents at runtime. However, those agents may have different preferences for ontological alignments, based on their ontologies and their tasks, and thus a negotiation mechanism should consider these preferences when searching for possible solutions.

In this paper, we examine the *Meaning-based argumentation* approach proposed by Laera *et.al.* [1], which allows two agents to argue over a set of candidate mappings obtained from a mapping repository. We postulate that *Ontology Modularization* can be used as a filtering mechanism for reducing the size of the ontologies used, and hence the size of the search space. Ontology modularization techniques typically split an ontology into partitions, or produce a subset, an *ontology module*, of the original ontology with respect to a supplied *signature* (i.e. a set of seed concepts). We demonstrate empirically that the number of correspondences that need to be argued over is reduced by an average of 75%, thus supporting the hypothesis that modularization can be used to reduce the cost of the argumentation process. The results also demonstrate that in 41% of the cases investigated, the use of argumentation could be avoided as no satisfactory solution is possible; i.e. where an insufficient number of correspondences exist that could support viable communication. Thus, an agent can avoid costly negotiations with those agents that would ultimately be unable to satisfy a query or task.

The paper is organized as follows: the motivation for this approach and related work is presented in Section 2, followed by Section 3 which presents the

---

<sup>1</sup> The terms *correspondence* and *mapping* are equivalent, and have been used interchangeably within this paper.

*Meaning-based argumentation* approach. Section 4 introduces ontology modularization and presents the various approaches that can be used to modularize an ontology, whilst Section 4.3 describes how modularization could be used to reduce the search mechanism for suitable alignments produced by the *Meaning-based argumentation*. Modularization as a search reduction mechanism is evaluated in Section 5, and Section 6 presents some concluding remarks.

## 2 Motivation and Background

Efficient agent communication within open and dynamic environments relies on the agents ability to reach a mutual understanding over a set of messages, where no prior assumptions can be made on the vocabulary used by the agents to communicate. Requests made to other agents should be interpreted through the underlying semantics of the request itself, and thus an agent should resolve any type of mismatch that may exist due to the use of different, but conceptually overlapping ontologies. Early solutions relied on the existence of a shared ontology, or simply assumed that a canonical set of ontology mappings (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. Likewise, the emergence of different alignment-generation tools [5] has resulted in the existence of different possible alignments between ontologies, whose suitability can vary depending on the agent's tasks, goals and preferences.

A number of solutions have been proposed that attempt to resolve ontological mismatches within open environments [3,1,4]. The work by van Diggelen *et al* [3] dynamically generates a minimal shared ontology, where minimality is evaluated against the ability of the different agents to communicate with no information loss. However, this approach uses a limited ontology model whose expressivity supports only simple taxonomic structures, with no properties and few restrictions other than disjointness and partial overlap. The expressive power of this model is non-standard, in that it does not correspond to any of the OWL [6] flavours<sup>2</sup>. Therefore, its applicability to the augmentation of existing real-world, published, OWL ontologies is limited.

The increased availability of mechanisms for ontology mapping and alignment [5] raises the challenge of discovering a number of different correspondence sets that may be mutually acceptable between two agents. This is essentially a collaborative search problem through the space of possible ontology correspondences between entities within different ontologies. To facilitate this search, a repository of (previously generated) correspondences for overlapping ontologies is required, as well as a mechanism for searching over these correspondences.

---

<sup>2</sup> The authors mention a reformulation of their model using Description Logics (the logical theory underpinning the standard ontology language OWL [6]), but no formal proof of the soundness of this reformulation is provided [3].

Mechanisms supporting the storage and provision of correspondences have already been devised; for example, Laera *et al* [1] postulate the notion of an *Ontology Alignment Server* (OAS), an agent which can supply potential mappings between two agents' ontologies. Various approaches that facilitate collaborative search have also been proposed [1,4]. A simple method might consist of a brute force approach that selects only those mappings whose level of confidence is above a certain threshold specified for each agent. More sophisticated approaches have exploited the use of *argumentation* as a negotiation mechanism to locate mappings that are mutually acceptable by both agents [1,4]. Laera *et al* [1] use argumentation as a rational means for agents to select ontology mappings from a repository [5], based on the notion of partial-order preferences over the different types of correspondences (e.g. structural vs terminological). Their approach assumed the use of OWL as a common ontology language. Dos Santos *et al* [4] proposed a variant on this idea, by representing ontology mappings as disjunctive queries in Description Logics.

The complexity of the search through the space of possible correspondences can, however, become prohibitive when complex negotiation mechanisms such as argumentation are involved, and reach  $\Pi_2^{(p)}$ -complete [7]<sup>3</sup>. This can make the search costly, especially when it is used to establish a common communication vocabulary (thus constituting the initial phase of any communication or transaction). Hence, it is important to identify ways in which the search space can be reduced *before* the argumentation process takes place.

One possible approach would be to reduce the size of the search space, by isolating only the pertinent correspondences that are relevant to some communication. This can be achieved by finding the relevant concepts in the original ontology through *ontology modularization*. Modularization refers to a set of principles and methodologies for either splitting an initial ontology into a number of partitions [8,9], or producing a subset of the input ontology with respect to a supplied signature [10,11,12,13,14,15]. Several different ontology modularization techniques have been proposed, which can broadly be classified as *traversal approaches* [10,11,12,13,14,15], which conditionally traverse the ontology represented as a graph, and *logical approaches*, [10,15] which identify modules that preserve certain logical properties, such as coverage.

In the next sections we introduce the *meaning-based argumentation* approach for dynamically selecting a set of mutually acceptable mappings from all the possible ones, and we provide a survey of ontology modularization approaches.

### 3 Arguing over Alignments

To better understand how the synergy between modularization and negotiation can be used to efficiently find alignments, we first present the *Meaning-based argumentation* [1], before then describing in the next Section how modularization can be used to reduce the space of all candidate mappings. Agents could then

---

<sup>3</sup> This is the complexity of deciding whether an argument is in every preferred extension of an agent (see Section 3).

argue over this reduced set of candidate mappings (as no rational agent should waste time arguing over possibly irrelevant alignments). The central hypothesis within this paper is that the use of modularization can be used to reduce the search space and consequently the cost of arguing over acceptable correspondences, by reducing the number of alignments, and hence arguments (this hypothesis is explored empirically in Section 5).

### 3.1 Value-Based Argumentation Framework (VAF)

This paper adopts the framework used by Laera *et al.* [16], which is based upon Bench-Capon's Value-Based Argument Framework (VAF) [17], that introduces the notions of *audience* and *preference values*. An audience represents a group of agents who share the same preferences over a set of values, with a single value being assigned to each argument. The VAF is based on the seminal work by Dung [18]. Dung showed that many forms of non-monotonic reasoning and logic programming are special forms of his argumentation theory.

In Dung's framework [18] attacks always succeed; in essence they are all given equal value. For deductive arguments this suffices, but in our scenario, ontology alignment negotiation, the persuasiveness of an argument could change depending on the audience, where an audience represents a certain set of preferences. One alternative is to use a Value-Based Argumentation Framework (VAF) [17], which assigns different strengths to arguments on the basis of the values they promote and the ranking given to these values by the audience for the argument. Thus, it is possible to systematically relate strengths of arguments to their motivations and to accommodate different audience interests.

**Definition 1.** A *Value-Based Argumentation Framework (VAF)* is defined as  $\langle AR, A, \mathcal{V}, \eta \rangle$ , where:

- $\langle AR, A \rangle$  is an argumentation framework;
- $\mathcal{V}$  is a set of  $k$  values which represent the types of arguments;
- $\eta : AR \rightarrow \mathcal{V}$  is a mapping that associates a value  $\eta(x) \in \mathcal{V}$  with each argument  $x \in AR$ .

The notion of *audience* is central to the VAF. Audiences are individuated by their preferences over the values. Thus, potentially, there are as many audiences as there are orderings of  $\mathcal{V}^4$ . The set of arguments is assessed by each audience in accordance to its preferences. An audience is defined as follows:

**Definition 2.** An audience for a VAF is a binary relation  $\mathcal{R} \subseteq \mathcal{V} \times \mathcal{V}$  whose irreflexive transitive closure,  $\mathcal{R}^*$ , is asymmetric, i.e. at most one of  $(v, v')$ ,  $(v', v)$  are members of  $\mathcal{R}^*$  for any distinct  $v, v' \in \mathcal{V}$ . We say that  $v_i$  is preferred to  $v_j$  in the audience  $\mathcal{R}$ , denoted  $v_i \succ_{\mathcal{R}} v_j$ , if  $(v_i, v_j) \in \mathcal{R}^*$

This notion allows us to consider that different agents (represented by an audience) can have different perspectives on the same candidate mapping. Thus, the VAF [17] defines what it means for an argument to be acceptable relative to some audience; it is defined as follows:

<sup>4</sup> Number of audiences =  $|\mathcal{V}|!$ .

**Definition 3.** Let  $\langle AR, A, \mathcal{V}, \eta \rangle$  be a VAF, with  $R$  and  $S$  as subsets of  $AR$ , and an audience  $\mathcal{R}$  :

- (a) For  $x, y \in AR$ ,  $x$  is a successful attack on  $y$  with respect to  $\mathcal{R}$  if  $(x, y) \in A$  and  $\eta(y) \not\prec_{\mathcal{R}} \eta(x)$ .
- (b)  $x \in AR$  is acceptable with respect to  $S$  with respect to  $\mathcal{R}$  if for every  $y \in AR$  that successfully attacks  $x$  with respect to  $\mathcal{R}$ , there is some  $z \in S$  that successfully attacks  $y$  with respect to  $\mathcal{R}$ .
- (c)  $S$  is conflict-free with respect to  $\mathcal{R}$  if for every  $(x, y) \in S \times S$ , either  $(x, y) \notin A$  or  $\eta(y) \succ_{\mathcal{R}} \eta(x)$
- (d) A conflict-free set  $S$  is admissible with respect to  $\mathcal{R}$  if every  $x \in S$  is acceptable to  $S$  with respect to  $\mathcal{R}$
- (e)  $S$  is a preferred extension for the audience  $\mathcal{R}$  if it is a maximal admissible set with respect to  $\mathcal{R}$
- (f)  $x \in AR$  is subjectively acceptable if and only if  $x$  appears in the preferred extension for some specific audience.
- (g)  $x \in AR$  is objectively acceptable if and only if  $x$  appears in the preferred extension for every specific audience.
- (h)  $x \in AR$  is indefensible if it is neither subjectively nor objectively acceptable.

### 3.2 Argumentation over Ontology Alignments

Laera *et al.* [16] adopt the VAF, allowing agents to express preferences for different mapping types, and restrict the arguments to those concerning ontology mappings allowing agents to explicate their mapping choices.

**Definition 4.** An agent,  $Ag_i$ , is characterised by the tuple  $\langle O_i, VAF_i, Pref_i, \epsilon_i \rangle$  where  $O_i$  is an ontology,  $VAF_i$  is the Value-based argumentation framework,  $Pref_i$  is the private pre-ordering of preferences over the possible values,  $\mathcal{V}$ , and  $\epsilon_i$  is the private threshold value.

Laera *et al.* define the arguments as follows:

**Definition 5.** An argument  $x \in AR$  is a triple  $x = \langle G, m, \sigma \rangle$  where  $m$  is a mapping,  $G$  is the grounds justifying the prima facie belief that the mapping does or does not hold and  $\sigma$  is one of  $\{+, -\}$  depending on whether the argument is that  $m$  does or does not hold

Laera [19] presents an algorithm for the agents to generate the arguments. The agents will argue for (+) a mapping if it is the agent's most preferred value in  $\mathcal{V}$  and the degree of confidence,  $n$ , of the mapping is greater than the agents private threshold,  $\epsilon$ ; otherwise the agent will argue against (-) the mapping.

Laera *et al.* [1] also address the notion of attack;  $x$  is attacked by the assertion of its negation,  $\neg x$ , this counter-attack is defined as follows:

**Definition 6.** An argument  $x \in AR$  attacks an argument  $y \in AR$  if  $x$  and  $y$  are arguments for the same mapping,  $m$ , but with different  $\sigma$ . For example, if  $x = \langle G_1, m, + \rangle$  and  $y = \langle G_1, m, - \rangle$ ,  $x$  counter-argues  $y$  and vice-versa.

**Table 1.** The classification of different types of ontological alignment approaches

<b>Semantic</b>	<i>M</i>	These methods utilise model-theoretic semantics to determine whether or not there is a correspondence between two entities, and hence are typically deductive. Such methods may include propositional satisfiability and modal satisfiability techniques, or logic based techniques.
<b>Internal Structural</b>	<i>IS</i>	Methods for determining the similarity of two entities based on the internal structure, which may use criteria such as the range of their properties (attributes and relations), their cardinality, and the transitivity and/or symmetry of their properties to calculate the similarity between them.
<b>External Structural</b>	<i>ES</i>	Methods for determining external structure similarity may evaluate the position of the two entities within the ontological hierarchy, as well as comparing parent, sibling or child concepts.
<b>Terminological</b>	<i>T</i>	These methods lexically compare the strings (tokens or n-grams) used in naming entities, or in the labels and comments concerning entities. Such methods may employ normalisation techniques (often found in Information Retrieval systems) such as stemming or eliminating stop-words, etc.
<b>Extensional</b>	<i>E</i>	Extension-based methods which compare the extension of classes, i.e., their set of instances. Such methods may include determining whether or not the two entities share common instances, or may use alternate similarity based extension comparison metrics.

Furthermore, in [1] a way to instantiate the set of values  $\mathcal{V}$  is also provided. These values depend on the methods used to generate the mappings; the possible values of  $\mathcal{V}$  are described in Table 1.

The agents can now express, and exchange, their arguments about ontology mappings and decide from their perspective, audience, what arguments are in their preferred extension; but the agents still need to reach a mutually acceptable position with regards to what ontology alignment they actually agree upon. Laera *et al.* define the notion of *agreed* and *agreeable* alignment as follows:

**Definition 7.** *An agreed alignment is the set of mappings supported by those arguments which are in every preferred extension of every agent.*

**Definition 8.** *An agreeable alignment extends the agreed alignments with those mappings supported by arguments in some preferred extensions of every agent.*

Thus, a mapping is *rejected* if it is in neither the agreed nor agreeable alignment. Given the context of agent communication it is rational for the agents to accept as many candidate mappings as possible[16], thus both sets of alignments are considered. The agents should only completely disagree when they want the opposite, indeed, the agents gain little by arguing and not reaching some kind of agreement.

## 4 Ontology Modularization

*Ontology modularization* [10,11,12,13,14,15] refers to the process of fragmenting existing ontologies into a set of smaller, and possibly interconnected parts, or *modules*. Broadly speaking, modularization approaches aim to identify the minimal set of necessary concepts and definitions for different parts of the original ontology.

The reasons for modularizing can be different and range from ontology reuse in order to support the work of ontology engineers [12,11,20] to information integration[10], or to support efficient agent communication [21]. Thus, whilst size is often quoted for some modularization techniques, it is unsuitable as an objective indicator of the quality of a module or the modularisation approach. This section reviews the different approaches for modularizing ontologies, focussing in particular on module extraction techniques, and presents the different techniques proposed in the literature for evaluating the result of modularization approaches.

An ontology  $O$  is defined as a pair  $O = (Ax(O), Sig(O))$ , where  $Ax(O)$  is a set of axioms (intensional, extensional and assertional) and  $Sig(O)$  is the signature of  $O$ <sup>5</sup>. This signature consists of the set of entity names used by  $O$ , i.e., its vocabulary. Ontology modularization is the process of defining a module  $M = (Ax(M), Sig(M))$ , where  $M$  is a subset of  $O$ ,  $M \subseteq O$ , such that  $Ax(M) \subseteq Ax(O)$  and  $Sig(M) \subseteq Sig(O)$ . No assumptions beyond this are made here about the nature of a module.

Approaches for modularizing ontologies belong to two main categories: ontology partitioning and ontology module extraction. *Ontology partitioning* is the process of fragmenting an ontology  $O$  into a set of (not necessarily disjoint<sup>6</sup>) modules  $\mathcal{M} = \{M_1, M_2, \dots, M_n\}$ , such that the union of all the modules should be equivalent to the original ontology  $O$ ; i.e.  $\{M_1 \cup M_2 \cup \dots \cup M_n\} = O$ . Thus, a function *partition*( $O$ ) can be formally defined as follows:

$$partition(O) \rightarrow \mathcal{M} = \{\{M_1, M_2, \dots, M_n\} | \{M_1 \cup M_2 \cup \dots \cup M_n\} = O\}$$

*Ontology module extraction* refers to the process of extracting a module  $M$  from an ontology  $O$  that covers a specified signature  $Sig(M)$ , such that  $Sig(M) \subseteq Sig(O)$ .  $M$  is the relevant part of  $O$  that is said to cover the elements defined by  $Sig(M)$ , therefore  $M \subseteq O$ .  $M$  is also considered as an ontology and could elicit further modules, depending on the signatures subsequently used. Thus, a function *extract*( $O, Sig(M)$ ) can be defined as follows:

$$extract(O, Sig(M)) \rightarrow \{M | M \subseteq O\}$$

This paper focusses on ontology module extraction approaches for query answering tasks, since the concept queried can form the basis of the signature used to extract modules. Ontology partitioning approaches are independent from any specific signature used to drive the modularization process, and thus would not reflect a query answering task. The techniques for ontology module extraction in the literature can be further subdivided into two distinct groups: *traversal approaches* and *logical approaches*. Traversal approaches [11,12,13,14] represent the extraction as a graph traversal, with the module being defined by the conditional traversal, which implicitly considers the ontological semantics, of the graph. Logical approaches [10,15] focus on maintaining the logical properties of coverage

<sup>5</sup> This definition is agnostic with respect to the ontology language used to represent the ontology, but it should be noted that the modularization techniques detailed in this section assume a description logic representation.

<sup>6</sup> This is in contrast to the mathematical definition of partitioning that requires partitions to be disjoint.

and minimality; as such, they explicitly consider the ontological semantics when extracting an ontology module.

#### 4.1 Traversal Based Extraction

The following approaches perform a traversal based extraction of an ontology module. **d'Aquin et al** [11] address the specific task of extracting modules related to components found in a given web page. Their ontology module extraction technique is integrated within a larger *knowledge selection* process. The specific aim is to dynamically retrieve the relevant components from online ontologies to annotate the webpage currently being viewed in the browser. The knowledge selection process comprises of three phases: (i) selection of relevant ontologies, (ii) modularization of selected ontologies and (iii) merging of the relevant ontology modules. The principle used for the extraction of an ontology module (i.e. phase (ii)) is to include all the elements that participate in the definition of an entity, either directly being included in the definition or indirectly (similar to the approach proposed by Seidenberg and Rector [14]). There are two distinct characteristics of this approach:

- **Inferences** are used and computed during the extraction. This is in contrast with other approaches (as is the case with techniques such as Doran et al [12]) that assume that an inferred model (including all the derived inferences) is computed prior to the module extraction process. For example, the transitivity of the *subClassOf* edge allows new subclass relations to be inferred in the input ontology.
- ‘**Shortcuts**’ are taken in the class hierarchy by including only the named classes that are the most specific common super-classes of the included classes. This is done by restricting the possible values of the *Least Common Subsumer(LCS)* algorithm [22] to the classes in the ontology.

**Doran et al.** [12] tackle the problem of ontology module extraction from the perspective of an Ontology Engineer wishing to reuse part of an existing ontology. The approach extracts an ontology module corresponding to a single user-supplied concept that is self-contained, concept-centred, and consistent. This approach is agnostic with respect to the language the ontology is represented in, provided that the ontology language itself can be transformed into the *Abstract Graph Model*. A conditional traversal descends down the *is-a* hierarchy from the signature concept; two edge sets are considered: one set of edges to traverse and a second set containing edges that are not traversed. Exceptions to these traversal sets are permitted during the first iteration of the algorithm. For example, when extracting an ontology module from an OWL ontology, `owl:disjointWith` edges are not traversed during the first iteration, but are considered in subsequent iterations (to prevent relevant definitions from being skipped).

**Noy and Musen.** [13] define the notion of *traversal view extraction*, which defines an *ontology view* of a specified concept, which is analogous to an ontology module. Starting from one class of the ontology being considered, relations from

this class are recursively traversed to include the related entities. These relations are selected by the user, and for each relation selected, a depth of traversal (or *traversal directive*) is assigned. This traversal directive is used to halt the traversal of the corresponding relation when the specified depth is reached. A *traversal view* consists of a set of traversal directives. This flexible approach (which was incorporated into PROMPT [23]) allows an Ontology Engineer to iteratively construct the ontology module that they require by extending the current ‘view’. However, this can require the Ontology Engineer to have a deep understanding of the ontology that is being used.

We do not consider this approach in our evaluation since it has a high degree of interactivity with the ontology engineer, that can affect the determination of a module.

**Seidenberg and Rector.** [14] developed a technique specifically for extracting an ontology module for a given signature,  $Sig(M)$ , from the Galen ontology. Their technique identifies all elements that participate (even indirectly) to the definition of the signature, or other elements in the extracted module. The algorithm can be decomposed based on the assumption that assuming we have a  $Sig(M) = \{A\}$ . Firstly the hierarchy is upwardly traversed (analogous to Upper Cotopy defined in [24]), so all of the  $A$ ’s superclasses are included. Next the hierarchy is downwardly traversed so that all the  $A$ ’s subclasses are included. It should be noted that the sibling classes of  $A$  are not included. The restrictions, intersection, union and equivalent classes of the already included classes can now be added to the module. Lastly, links across the hierarchy from the previously included classes are traversed; the target of these links are also upwardly traversed.

Whilst the degree of generality for this approach is high (with respect to other ontologies), the focus on GALEN introduces certain features that may be less suitable for other ontologies. For example, result of property filtering can lead to class definitions becoming equivalent, whilst this is not incorrect it does introduce unnecessary definitions that can be transformed into primitive classes.

## 4.2 Logical Based Extraction

The logical based extraction techniques are based on the notion of conservative extension [25] whereby an ontology module extracted from a given ontology is considered a conservative extension if the entailments regarding the ontology module are captured totally within its signature. More formally Lutz *et al* [25] present the following definition:

**Definition 9.** *Let  $\mathcal{T}_1$  and  $\mathcal{T}_2$  be TBoxes formulated in a Description Logic (DL)  $\mathcal{L}$ , and let  $\Gamma \subseteq sig(\mathcal{T}_1)$  be a signature. Then  $\mathcal{T}_1 \cup \mathcal{T}_2$  is a  $\Gamma$ -conservative extension of  $\mathcal{T}_1$  if for all  $C_1, C_2 \in \mathcal{L}(\Gamma)$ , we have  $\mathcal{T}_1 \models C_1 \sqsubseteq C_2$  iff  $\mathcal{T}_1 \cup \mathcal{T}_2 \models C_1 \sqsubseteq C_2$ .*

Thus, all the entailments regarding the signature of the ontology module are equivalent to using the ontology module with the ontology it was taken from. Unfortunately, Lutz *et al* [25] also show that deciding if an ontology  $O$  is a conservative extension is undecidable for OWL DL. However, Konev *et al* [15] have

developed an algorithm, MEX, for extracting conservative extensions from acyclic terminologies formulated in more expressive types of DL ( $\mathcal{ALCT}$  or  $\mathcal{ELT}$ )<sup>7</sup>. Whilst these restrictions limit the use of this approach, it can be successfully applied to large, real world ontologies such as SNOMED CT.

Grau *et al* [10] overcome the limitations of conservative extensions for more expressive description logics by utilizing approximations; they term these modules as locality-based modules. Coverage and safety are the properties that locality-based modules can guarantee, but this is done at the expense of minimality which is also guaranteed by conservative extensions. Coverage and safety [27] are defined in terms of a module being imported by a local ontology ( $\mathcal{L}$ ) as follows:

**Coverage.** Extract everything the ontology defines for the specified terms. The module  $O'$  covers the ontology  $O$  for terms from, some signature,  $X$  if for all classes  $A$  and  $B$  built from terms in  $X$ , such that if  $\mathcal{L} \cup O \models A \sqsubseteq B$  then  $\mathcal{L} \cup O' \models A \sqsubseteq B$ .

**Safety.** The meaning of the extracted terms is not changed.  $\mathcal{L}$  uses the terms from  $X$  safely if for all classes  $A$  and  $B$  built from terms in  $X$ , such that if  $\mathcal{L} \cup O' \models A \sqsubseteq B$  then  $O' \text{ models } A \sqsubseteq B$ .

Two different variants of locality are described by Grau *et al* [28]. Syntactic locality can be computed in polynomial time, but semantic locality is PSPACE-complete. Syntactic locality is computed based on the syntactic structure of the axiom whereas semantic locality is computed based on the interpretation ( $\mathcal{I}$ ) of the axiom. The issue concerning the syntactic locality is that syntactically different (but semantically equivalent) axioms can be treated differently. For example, Borgida and Giunchiglia [29] raise this issue of the syntactic approximation via the following example; consider the two sets of axioms  $\{A \sqsubseteq (B \sqcap C)\}$  and  $\{A \sqsubseteq B, A \sqsubseteq C\}$ . These axioms are semantically equivalent, but the syntactic difference will effect the extraction process. The syntactic locality also can not handle tautologies, but this is unlikely to affect real world applications as ontologies with tautologies would be considered badly engineered.

### 4.3 Combining Ontology Modularization and Argumentation

Ontology modularization can be used as a pre-processing step to improve the efficiency of an argumentation framework, when used to search the space of all candidate ontology mappings. When two agents communicate, only the initiating agent ( $Ag_1$ ) is aware of its task, and consequently, what concepts are relevant to this task. It can therefore select these relevant concepts within the signature of the desired ontology module. The signature of the resulting ontology module can then be used to filter the correspondences, and consequently the number of arguments necessary within the argumentation process. The steps in Table 2

<sup>7</sup> The expressivity of a DL is determined by the constructors allowed by the language, such as negation, existential restriction, etc. For more details we invite the reader to refer to [26].

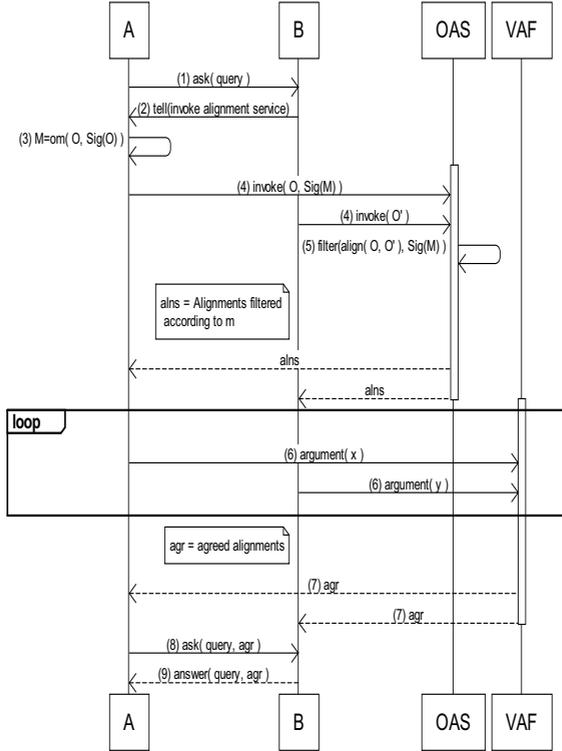


Fig. 1. UML Sequence Diagram of Ontology Modularization and Argumentation

describe this process, whilst Figure 1 depicts the process as a UML Sequence Diagram. It is assumed that two agents,  $Ag_1$  and  $Ag_2$  have ontologies  $O$  and  $O'$  respectively.

It is assumed that two agents,  $Ag_1$  and  $Ag_2$  have ontologies  $O$  and  $O'$  respectively.

The set of ontology correspondences are filtered at Step 5 according to the following function:

**Definition 10.** A filtering function,  $filter()$ , filters the set of candidate mappings prior to argumentation  $Z$  into a subset  $Z' \subseteq Z$  such that:

$$filter(Z, Sig(M)) : Z \rightarrow Z' \mid \forall m \in Z', m = \langle e, e', n, R \rangle \text{ and } e \in Sig(M).$$

Steps 6 and 7 represent a black-box process, which is the argumentation process. Modularization is therefore used to filter the correspondences that are passed to this process. The combination of these two processes reduces the cost of reaching an agreement over the set of correspondences, by reducing the size of the set of correspondences, and hence the number of arguments.

**Table 2.** Steps involved in Ontology Modularization and Argumentation

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. <math>Ag_1</math> asks a query, <math>query(A \in Sig(O))</math>, to <math>Ag_2</math>.</li> <li>2. <math>Ag_2</math> does not understand the query, <math>A \notin Sig(O')</math>, and informs <math>Ag_1</math> they need to use an Ontology Alignment Service (OAS)</li> <li>3. <math>Ag_1</math> produces, <math>om(O, Sig(A))</math>, an ontology module, <math>M</math>, to cover the concepts required for its task.</li> <li>4. <math>Ag_1</math> and <math>Ag_2</math> invoke the OAS. <math>Ag_1</math> sends its ontology, <math>O</math> and the signature of <math>M</math>, <math>Sig(M)</math>.</li> <li>5. The OAS aligns the two ontologies and filters the correspondences according to <math>M</math>. Only those correspondences featuring an entity from <math>M</math> are returned to both agents.</li> <li>6. The agents begin the Meaning-Based Argumentation process, and iterate it, with each agent generating arguments and counter-arguments.</li> <li>7. The iteration terminates when the agents reach an agreement on a set of correspondences, and this set is returned to both agents.</li> <li>8. <math>Ag_1</math> asks a query to <math>Ag_2</math> but uses the correspondences so that <math>Ag_2</math> understands, <math>query(A \in Sig(O) \wedge B \in Sig(O'))</math> where <math>A</math> and <math>B</math> are aligned.</li> <li>9. <math>Ag_2</math> answers the query making use of the resulting alignment.</li> </ol> |
|--|

## 5 Evaluation

The aim of the evaluation is to show that using modularization to filter the set of alignments reduces the cost of the argumentation process. The evaluation considers how modularization affects the number of correspondences that are argued over and subsequently agreed upon. The aim is to demonstrate that using ontology modularization prior to invoking the argumentation lessens the effort required by the agent to negotiate acceptable correspondences without compromising the agent's ability to perform its task.

### 5.1 Evaluation Setup

The ontologies used in the evaluation are listed in Table 3, complete with details of the number of named classes, number of properties and the level of expressivity of the ontology. This is the same set as that available for the OAEI 2007 Conference track<sup>8</sup>, with the exclusion of three ontologies<sup>9</sup>. This track has been selected since it is the one with the larger number of real world ontologies, allowing for more pairwise alignments than the other tracks (apart from the benchmark track, that uses artificial ontologies).

To the best of our knowledge, there is no canonical, hand-crafted gold standard for the alignments between the chosen ontologies. To explore the effectiveness of modularization with different alignment techniques and a baseline of valid alignments, the results of the Falcon-AO system<sup>10</sup> were used to provide a *gold standard* in the experiments. Falcon-AO is currently the best performing ontology alignment system, as it provides the best compromise between precision and recall in the alignment task performed for this track.

<sup>8</sup> <http://oaei.ontologymatching.org/2007/conference/>

<sup>9</sup> These ontologies have memory requirements of >1.5GB.

<sup>10</sup> <http://iws.seu.edu.cn/projects/matching/>

**Table 3.** Classes, properties, expressivity, and average % reduction in module size for the test datasets

Name	# Cl.	# Prop.	DL expressivity	% Reduction
Cmt	36	59	<i>ALCIF(D)</i>	43
ConfTool	38	36	<i>SIF(D)</i>	28
Crs	14	17	<i>SHLN</i>	39
Edas	104	50	<i>ALCIF(D)</i>	28
Ekaw	77	33	<i>SHLN(D)</i>	34
Sofsem	60	64	<i>ALCHIF(D)</i>	28
Micro	32	26	<i>ALCIOF(D)</i>	14
Pcs	23	38	<i>ELUIF(D)</i>	28
OpenConf	62	35	<i>ALCIO(D)</i>	37
Paperdyne	47	82	<i>ALCHIOF(D)</i>	0
Sigkdd	49	28	<i>ELI(D)</i>	13

The experimental setup consisted of two main tracks:

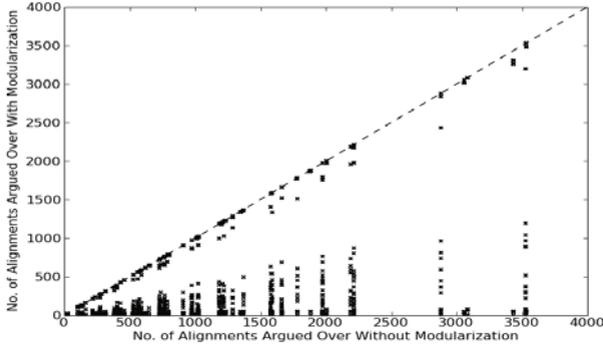
1. **Builtin** - The possible alignments were determined by using simple textual and structural similarity between concepts and properties; essentially a combination of alignment techniques.
2. **Gold standard** - The possible alignments were those obtained by the Falcon-AO system.

For each track, each ontology has been compared to all other ontologies (excluding itself); for a total of 110 distinct pairs, giving us 220 tests in total. For each pair in the builtin track, the possible alignments have been computed, whilst for the gold standard track the alignments generated by the Falcon-AO system were used. In this evaluation we used Doran *et al*'s approach [12] to modularize the ontologies, as the method has been demonstrated to generate accurate modules for query answering tasks and its performance is comparable to several of the other methods described earlier [30]. The argumentation procedure was started, whereby two agents were created, each one adopting an ontology and using its own preferences. The number of correspondences argued over, broken down into accepted and rejected, was recorded for each test.

The result of the argumentation process between the ontologies when no modularization occurred was used as a baseline result for each pair. Table 3 presents the average reduction in size due to the modularization process; with the overall average being 26.5%. The Paperdyne ontology was not affected by the modularization process as it has a very shallow hierarchy with respect to the number of concepts, and has numerous properties which result in a highly interconnected ontology that is not amenable to modularization.

## 5.2 Results

The results indicate that ontology modularization has a considerable impact on the number of correspondences that are argued over. On average, this number is reduced by 75% (69% for the built-in and for the gold standard 79%). Figure 2 shows a scatter plot of all the test cases, with the *x*-axis being the baseline without modularization and the *y*-axis being that with modularization.



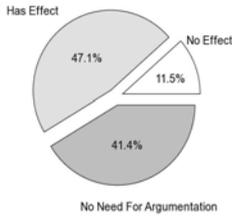
**Fig. 2.** A scatter plot showing the total number of correspondences argued over without and with modularization

The plot shows that there is a significant number of points where modularization has a considerable effect; it also shows in certain cases it has no effect, the cases where the point lies on the dotted line. Interestingly, and somewhat unexpectedly, there are a number of cases where the modularization (those cases where  $y$  is zero) effectively stops the argumentation process from taking place. This is an interesting result as it suggests that the agents are able to identify the cases where without modularization they would produce an alignment that would not help with their task. Thus, it is reasonable to postulate that combining modularization with argumentation should prevent cases where the use of argumentation is redundant.

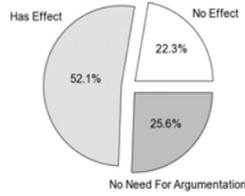
The experiments identify three cases: those where modularization has no effect, where it reduces the number of correspondences, and where argumentation is redundant. Figure 3 shows the distribution of these three cases as a pie chart. This shows that the case where modularization has an effect (47.1%) is considerably larger than that where it has no effect (11.5%). More interestingly, perhaps, is that in 41.4% of the trials, the agents are stopped from entering an unnecessary argumentation process.

The two pie charts in Figures 4 and 5 show these three types of point split between the two tracks, builtin and gold standard. These show that when using the gold standard (i.e., the set of correspondences that has less noise), the cases where unnecessary argumentation is avoided are greater. Conversely the cases where modularization has no effect are greater in the builtin track. This would seem to suggest that OAS should aim to provide gold standard alignments, but in the likely case that this is not possible, then combining modularization with argumentation would be of benefit.

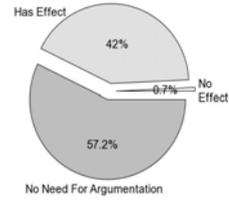
It is also interesting to see the effect that modularization has on the number of correspondences that are accepted by the argumentation process. Figure 6 shows



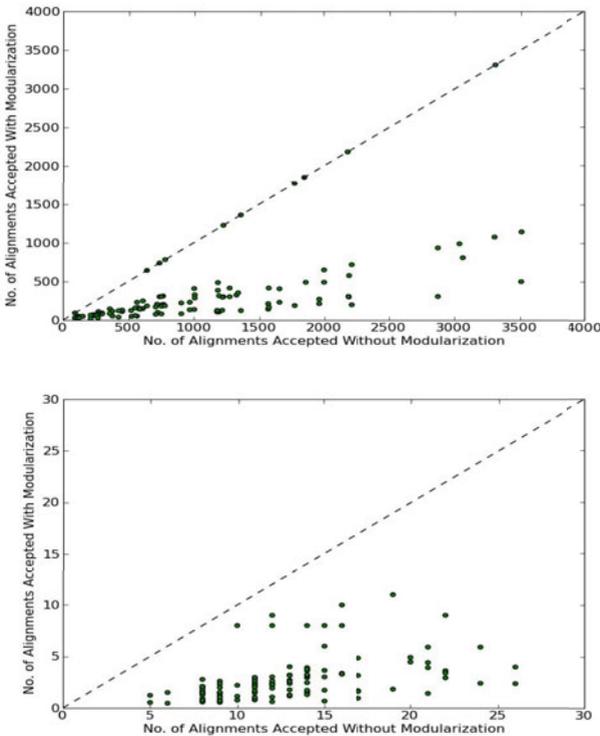
**Fig. 3.** Pie chart showing three different point types



**Fig. 4.** Pie chart showing the different point types for the builtin approach



**Fig. 5.** The different point types for the gold standard approach



**Fig. 6.** Two scatter plots showing the number of correspondences accepted without and with modularization: top for the builtin and the bottom for the gold standard

the number of accepted correspondences without using the argumentation process against the number accepted when performing the argumentation process. These graphs support the intuition that since the agents are arguing over fewer correspondences then they eventually *agree* on fewer correspondences.

### 5.3 Effect on Cost

The analysis of the efficiency of the combined processes of modularization and argumentation depends on the effect on the cost of the combination of these mechanisms, in the remainder of this section we discuss this issue.

The six decision problems of the VAF and their associated complexity [7] are presented in Table 4, where  $S$  is a finite set of arguments and  $H$  is an argument system.

**Table 4.** Decision complexity in a VAF

Type of Decision	Complexity
Is $S$ admissible?	P
Is $S$ stable?	P
Is $S$ preferred?	CO-NP complete
Is $x$ in a preferred $S$ ?	NP-complete
Has $H$ a stable extension?	NP-complete
Is $x$ in <i>every</i> preferred $S$	$\Pi_2^{(p)}$ -complete

All of the above decision problems are used by Laera *et al* [1] for each run of the argumentation over ontology correspondences. Due to this high complexity, there is a substantial motivation to reduce the size of the input; the input being the number of arguments which is correlated to the number of ontology correspondences. If the number of ontology correspondences can be reduced, then fewer arguments can be drawn, and whilst the complexity of the decision problems remains the same, the cost of the argumentation process can be reduced.

Consider the function  $om(O, Sig(X))$  that implements an ontology module extraction algorithm and outputs an ontology module  $M$ . By definition (Section 4) an ontology module must be a subset or equal to the ontology it was derived from ( $M \subseteq O$ ). If  $M$  has  $p$  entities,  $O$  has  $n$  entities and  $O'$  – the ontology we are aligning to – has  $m$  entities then in the worst case we now have  $p \times m$  correspondences, where ( $p \times m \leq n \times m$ ). However, this worst case is unlikely to occur and even less likely when considering ontologies of non trivial size (*i.e.*, whose size  $s > 2$ ). Another implication is that there is an incentive to reduce  $M$  as much as possible, whilst ensuring that  $M$  is still suitable for the task required.

## 6 Conclusions

Agents need to reconcile ontological differences, especially within the context of open and dynamic environments where no *a priori* assumptions about the nature of the ontology can be made. Negotiation frameworks (such as the *Meaning-based argumentation* framework), allows agents to negotiate over different ontology correspondences, and identify those alignments that are mutually acceptable.

However, this collaborative search is computationally costly, as the complexity of the decision problems involved range from  $P$  to  $\Pi_2^{(p)}$ -complete. However, *Ontology Modularization* can be exploited to reduce the size of the search space, and hence cost of finding acceptable alignments. The use of ontology modularization as a filter-based pre-processing stage was evaluated empirically, and found to reduce the average number of correspondences (and hence size of the search space) presented to the argumentation framework by 75%, across a number of different ontology pairs. In addition, three patterns emerged: i) where no reduction in size occurred (in 11.5% of cases on average); ii) where the number of correspondences was reduced (47.1%); and iii) where modules of size zero were found (41.4%). We found that this latter case corresponded to failure scenarios; i.e. where the subsequent transaction could fail due to insufficient alignment between the ontologies. Thus, this paper demonstrates that ontology modularization not only reduces the cost of negotiating over correspondences and establishing communication, but can be used to identify cases where negotiation will fail to identify relevant correspondences to support meaningful queries.

## References

1. Laera, L., Blacoe, I., Tamma, V.A.M., Payne, T.R., Euzenat, J., Bench-Capon, T.J.M.: Argumentation over ontology correspondences in mas. In: Durfee, E.H., Yokoo, M., Huhns, M.N., Shehory, O. (eds.) AAMAS, p. 228 (2007)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* 284 (2001)
3. van Diggelen, J., Beun, R.J., Dignum, F., van Eijk, R.M., Meyer, J.J.: Ontology negotiation in heterogeneous multi-agent systems: The anemone system. *Applied Ontology* 2, 267–303 (2007)
4. dos Santos, C.T., Quaresma, P., Vieira, R.: Conjunctive queries for ontology based agent communication in mas. In: 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, vol. 2, pp. 829–836 (2008)
5. Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer, Heidelberg (2007)
6. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: *OWL Web Ontology Language Semantics and Abstract Syntax* (2004)
7. Dunne, P., Bench-Capon, T.: Complexity in value-based argument systems. In: Alferes, J.J., Leite, J. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 360–371. Springer, Heidelberg (2004)
8. Cuenca-Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Automatic Partitioning of OWL Ontologies Using E-Connections. In: Proceedings of the 2005 International Workshop on Description Logics, DL 2005 (2005)
9. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. In: Proceedings of the 3rd International Semantic Web Conference, Hiroshima, Japan (2004)
10. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research (JAIR)* 31, 273–318 (2008)

11. d'Aquin, M., Sabou, M., Motta, E.: Modularization: a key for the dynamic selection of relevant knowledge components. In: ISWC 2006, Athens, Georgia, USA (2006)
12. Doran, P., Tamma, V.A.M., Iannone, L.: Ontology module extraction for ontology reuse: an ontology engineering perspective. In: Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, Ø.H., Falcão, A.O. (eds.) CIKM, pp. 61–70. ACM, New York (2007)
13. Noy, N.F., Musen, M.A.: Specifying ontology views by traversal. In: International Semantic Web Conference, pp. 713–725 (2004)
14. Seidenberg, J., Rector, A.: Web ontology segmentation: analysis, classification and use. In: WWW 2006: Proceedings of the 15th international conference on World Wide Web, pp. 13–22. ACM Press, New York (2006)
15. Konev, B., Lutz, C., Walther, D., Wolter, F.: Semantic modularity and module extraction in description logics. In: Proceedings of ECAI 2008: 18th European conference on Artificial Intelligence (2008)
16. Laera, L., Blacoe, I., Tamma, V., Payne, T., Euzenat, J., Bench-Capon, T.: Argumentation over Ontology Correspondences in MAS. In: Proc. of the 6th Int. Joint Conf. on Autonomous Agents and Multiagent Systems, pp. 1285–1292 (2007)
17. Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* 13, 429–448 (2003)
18. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77, 321–357 (1995)
19. Laera, L.: Toward shared understanding - an argumentation based approach for communication in open multi-agent systems (2008)
20. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 450–455. AAAI Press/The MIT Press (2000)
21. Doran, P., Tamma, V., Palmisano, I., Payne, T.R.: Dynamic selection of ontological alignments: a space reduction mechanism. In: Twenty-First International Joint Conference on Artificial Intelligence, IJCAI 2009 (2009)
22. Cohen, W.W., Borgida, A., Hirsh, H.: Computing least common subsumers in description logics. In: AAAI, pp. 754–760 (1992)
23. Noy, N.F., Musen, M.A.: The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies* 59, 983–1024 (2003)
24. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 251–263. Springer, Heidelberg (2002)
25. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In: [31], pp. 453–458
26. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge (2007)
27. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: Extracting modules from ontologies. In: WWW 2007, Proceedings of the 16th International World Wide Web Conference, Banff, Canada, May 8–12, pp. 717–727 (2007)
28. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In: [31], pp. 298–303

29. Borgida, A., Giunchiglia, F.: Importing from functional knowledge bases - a preview. In: Cuenca-Grau, B., Honavar, V., Schlicht, A., Wolter, F. (eds.) WOMO (2007)
30. Palmisano, I., Tamma, V., Payne, T.R., Doran, P.: Task oriented evaluation of module extraction techniques. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 130–145. Springer, Heidelberg (2009)
31. Veloso, M.M. (ed.): IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12 (2007)