Evolving Ontological Knowledge Bases through Agent Collaboration

Heather Packer[1], Terry Payne[2], Nicholas Gibbins[1], and Nicholas R. Jennings[1]

[1] School of Electronics and Computer Science
University of Southampton, United Kingdom
{hp07r, nmg, nrj}@ecs.soton.ac.uk
[2] Department of Computer Science
University of Liverpool, United Kingdom
trp@liv.ac.uk

Abstract. This paper presents initial work that will enable an agent to augment its ontology to incorporate required knowledge from other agents, in order to let it answer domain related queries. Specifically, our agents are heterogeneous, whereby an agent has its own interest domain and represents this with an ontology that contains relevant conceptualisations. These agents have intersecting domain interests and their ontologies represent a set of overlapping concepts with alternative symbolic representations. In this setting, our proposed approach focuses on reducing the costs associated with acquiring knowledge through collaboration, and augmenting axioms into an agent's ontology. In order to achieve this, we consider incorporating knowledge to reduce the number of messages required to answer repetitive domain related queries that require mediation, and select a shared set of axioms that represent conceptual knowledge. We present results from our approach and identify the number of messages and axioms required for a repeated transaction. These preliminary results show that augmenting an agent's ontology can indeed reduce the number of messages and axioms required.

Key words: agent, ontology, augmentation, collaboration, fragment

1 Introduction

Agents that model a domain for the purpose of answering queries, can be limited to the knowledge instantiated in their model. However, if an agent can allows its knowledge base to evolve, it can model a domain with its own terminology, while incorporating new concepts. This enables an agent to change to reflect its environment, and ensures that its ontologies do not have to be explicitly remodelled in response to environment changes. While adding new knowledge to an agent's knowledge base can increase the range of queries it can handle, the acquisition and storage of this knowledge may incur costs. We propose an approach to reduce the cost of acquiring and incorporating knowledge. Furthermore, we propose that when an agent regularly collaborates on repeated tasks, it is beneficial to augment its ontology with the concept used in these tasks. This enables agents to communicate in a common vocabulary, and remove the need for the repeated acquisition of this knowledge, reducing the costs associated with the acquisition. Therefore, the overhead costs of acquiring knowledge and the number of times it is used, determines when it is beneficial to incorporate new concepts. Possible overheads include the computational complexity of the mediator's translation service, or a payment required by the mediator. Additionally, the agent that requires new knowledge may encounter limitations, such as bandwidth, resource (such as battery power) and computational restrictions. These overhead costs associated with these factors can be reduced by augmenting an agent's ontology. However, augmenting the agent's ontology also has associated costs, relating to the validation and incorporation of knowledge into a structure. For example, an agent may require the services of a reasoner to validate the knowledge to be incorporated (consistency checking with a DL reasoner, for example). With the addition of new knowledge an ontology can become more complex than required by an agent, with the result that inference becomes more computationally expense. Therefore, we propose that it is advantageous to incorporate a select amount of knowledge into an agent's knowledge base due to these costs.

Our agents use an ontology to model their domain, and each of these agents have an individual ontology, which represents its individual view of its domain. In more detail, Gruber [1] defines an ontology as "an explicit specification of a conceptualization", and a conceptualisation as "an abstract, simplified view of the world that we wish to represent for some purpose"; our agents' ontologies are designed to support certain domain related tasks. For example, agents a_1 and a_2 have a domain model representing an office removal company and rental company, respectively. Additionally, a_1 and a_2 handle requests for removing offices' content and renting vehicles to customers, respectively. However, if an agent cannot complete a domain related task, it can benefit from collaborating with other agents' ontologies that contain the required conceptualisations to complete the task. In our example case, agent a_1 might require a large van to move office furniture, and is unable to rent a 'Large_Van' as a_2 represents a large van with the symbol 'LWB_Van'. Hence, it is necessary to be able to communicate in a common vocabulary with these agents. In order to generate a common vocabulary between agents that represent the same conceptualisations with different logical symbols, it has been proposed by Euzenat [2] that a mediator can provide mappings between semantically equivalent concepts, with the use of ontology matching techniques. A mediator can be a centralised or decentralised service in an environment, scaling to handle the demands of the environment.

In order to reduce the cost of acquiring regularly required knowledge, we propose a novel technique that selects axioms to augment into an agent's ontology. In more detail, an axiom is a sentence in first order logic that is assumed to be irrefutably true and an ontology contains a set of such axioms which represent a specific domain. In particular, our approach aims to retrieve axioms related to a specific concept in the form of ontology fragments, we aim to reduce the cost of regularly acquiring this knowledge by augmenting an agent's ontology. In our context an ontology fragment is a set of axioms from an ontology, and a fragment represents axioms related to a specified concept. In order to select axioms from a set of fragments, our approach choses a group of axioms by analysing the concepts and their semantic similarities from the set of retrieved fragments using an ontology matching technique. This selection enables the reduction of cost compared with incorporating all the retrieved knowledge, due to the associated cost of integrating and storing domain knowledge. Our initial approach only considers augmenting an agent's ontology with new knowledge, future work will consider when to remove incorporated knowledge. Removing incorporated knowledge may be beneficial when this knowledge is not used over a period of time, because of the overhead cost of modelling this knowledge due to the complexity issues discussed above, thus providing a non-monotonic function. Our preliminary results show that our approach reduces the number of messages, and the amount of knowledge passed in these messages for a repetitive transaction, compared with acquiring this knowledge each time on demand. This approach aims to evolve an agent's ontology incrementally, focusing on incorporating knowledge that is of immediate use. The process takes an iterative approach, enacted each time an agent requires additional knowledge about a domain concept. Specifically, our approach aims to:

1. Support the automatic exchange of knowledge to augment agent-based problem solving.
2. Reduce the number of messages required to carry out repeated tasks (the number of messages required after the incorporation of all the required knowledge will be reduced to the minimum number).
3. Analyse an agent's ontology and compare it to a fragment that contains a shared representation of a conceptualisation. In order to evaluate the relationship between each axiom from the fragment and the ontology's domain.

In particular, our technique is designed for agents with small ontologies (with approximately 50 - 200 classes) that do not completely model the entirety of a domain. Furthermore, our technique lends itself to agents that provide services, such as reference catalogues, companies and public services, because these services are of interest to agents in many domains. For example, a company that handles the removal of the contents of offices describes the items that it can remove and their requirements with an ontology, this ontology can evolve as the company expands its business to handle additional items. The removal company can then expand their company so that they can remove hazardous chemicals, to aid demolition workers.

Our paper is structured as follows. In Section 2 we summarise related work, and in Section 3 we describe an overview of our technique. Then, in Section 4 we present in more detail our techniques. In Section 5 we provide an illustrative example situated in a possible use case and Section 6 presents our preliminary results. We conclude in Section 7.

2 Background Work

This section describes the state of the art, which augments an agent's ontologies with new knowledge from other agents. The approaches presented by van Diggelen et al. [3], Van Eijk et al. [4], Doherty et al. [5], Bailin and Truszkowski [6], Afsharchi et al. [7], Wiesman and Roos [8], and Soh [9] enable their agents to augment their ontologies with new knowledge, when agents have different domain models representing the same domain. However, these techniques are limited as they require their agents to model the same domain, and they do not consider the cost of acquiring and incorporating new knowledge into their agent's ontologies. In particular, the approaches proposed by Doherty et al., and Bailin and Truszkowski attempt to address the issue of semantically equivalent representations with the use of the conditions describing a concept, and alternate symbolic representations. When a group of agents describe different domain models, these approaches cannot guarantee the accuracy of the alignment. Alternatively, ontology matching techniques [2] have addressed this problem by providing a confidence rating that measures the assurance of a mapping between two concepts. In constrast, the approaches of Afsharchi et al., Wiesman and Roos, and Soh focus on the validation of the knowledge to be incorporated into the agent's ontology. These approaches are dependent on agents either containing accurate knowledge about a concept, or strong positive and negative instance examples. Although these approaches provide automated techniques that augment an agent's ontology to incorporate additional knowledge about a desired concept, and can evaluate the validity of the knowledge, they have limitations whereby they require certain conditions (for example, they require common experiences). However, the approaches presented by Afsharchi et al. [7] and Soh [9] consult other agents when there are conflicts between the definition of a concept. This enables an agent to research a common consensus on the accuracy of the knowledge.

In addition to agent based research, the Semantic Web community have produced relevant work in evolving ontologies. In particular, the techniques presented by Flouris et al. [10–12] have applied a generalisation of the AGM (Alchourrón, Gärdenfors and Makinson) [13] theory of the expansion, revision and contraction of knowledge to Description Logic ontologies. Specifically, Flouris et al. [12] present a framework that can be used to evaluate the consistency of an ontology using the AGM theory, and present the supporting propositions. This work enables the evaluation of coherence and consistency [12], the next step to evolve an ontology is to locate and manage inconsistencies. The work presented by Hasse et al. [14] presents a basic method to locate inconsistencies in an ontology and how an inconsistent ontology can yield meaningful results. Hasse and Stojanvoic [15] further explore a method to locate, and possible techniques to resolve, three types of inconsistency; structural, logical, and user defined inconsistencies. These techniques offer our approach possible methodologies that can be used to evaluate, locate and resolve inconsistent knowledge to be incorporated into an ontology. Currently our approach does not consider resolving inconsistent axioms, and this will be considered in future work.

Given this background, we have considered how to augment an agent's ontology with new knowledge, while analysing how to reduce the overhead cost involved with augmenting an ontology. Similarly to Afsharchi et al. [7] and Soh [9], we also consider how to incorporate knowledge and select which knowledge has a higher priority, by considering which knowledge is contained in the majority of collaborating agents' ontologies.

3 Evolving an Ontological Knowledge Base

This section gives an overview of our general approach, and details the interaction between the actors in our environment. Our approach considers four actors: i) an agent, a_i, that provides services related to a domain to a user, and has an ontology, o_i, modelling its domain, ii) a user that sends domain related requests to a_i, iii) an agent, a_j, that provides domain related services and models its domain with its own ontology, o_j, where o_j and o_i contain intersecting concepts, and iv) a mediator that provides mappings between concepts.

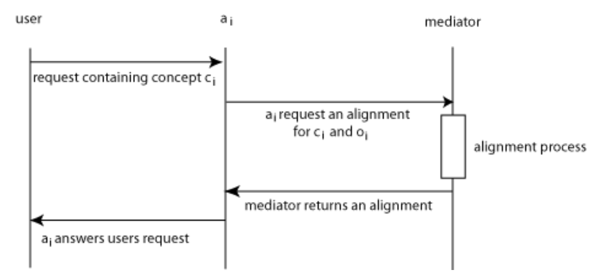

Fig. 1. General approach when an agent's ontology, o_i, contains the required knowledge but requires a translation.

In more detail, agent a_i receives requests from a user that contains a concept c_i, this situation has three possible scenarios. First, a_i can answer the request as its ontology contains the concept contained in the request. Second, a_i's ontology does not contain c_i, and consults a mediator to find out if there is a mapping between c_i and concepts contained in o_i. When the mediator identifies that o_i contains a semantically equivalent concept to c_i it answers the user's request, as shown in Figure 1. In contrast, the third scenario handles the case where o_i does not contain a semantically equivalent concept to c_i, as shown in Figure 2. In this situation, agent a_i attempts to locate fragments that represent c_i, so that it can incorporate axioms relating to c_i into o_i. Our agents pass fragments when requested by other agents to describe a conceptualisation. In this third case, an agent broadcasts a request to other agents in the environment to locate agents with potential alignments. Then the agent retrieves the fragments that potentially represent c_i, and then a_i selects axioms to incorporate from the retrieved fragments into o_i. Specifically, our technique merges these fragments and selects axioms according to our algorithms presented in the following section.

In particular, Figure 2 depicts the three sets of processes, i) the alignment process of the mediator, ii) the fragmentation process of a_j, and iii) the selection and incorporation of axioms from the retrieved fragments. Each of these processes can use different techniques to provide the required outcomes. Our approach focuses on the third process, and we consider a technique that selects axiomatic knowledge an agent will incorporate into its ontology (see Section 4). In particular, our approach broadcasts a request for fragments to a set of agents in the environment, and receives a set of fragments, which potentially represents the desired concept. We aim to retrieve a set of fragments so that an agent can incorporate the shared meaning of a concept, thus enabling it to collaborate with a set of agents. Similarly, to the techniques proposed by Afsharchi et al. [7] and Soh [9] our approach takes advantage of the collective knowledge in an environment. Although, in contrast to our approach they consult the participating agents in the case of inconsistent knowledge. The following section describes our technique to select axioms from this set of fragments.

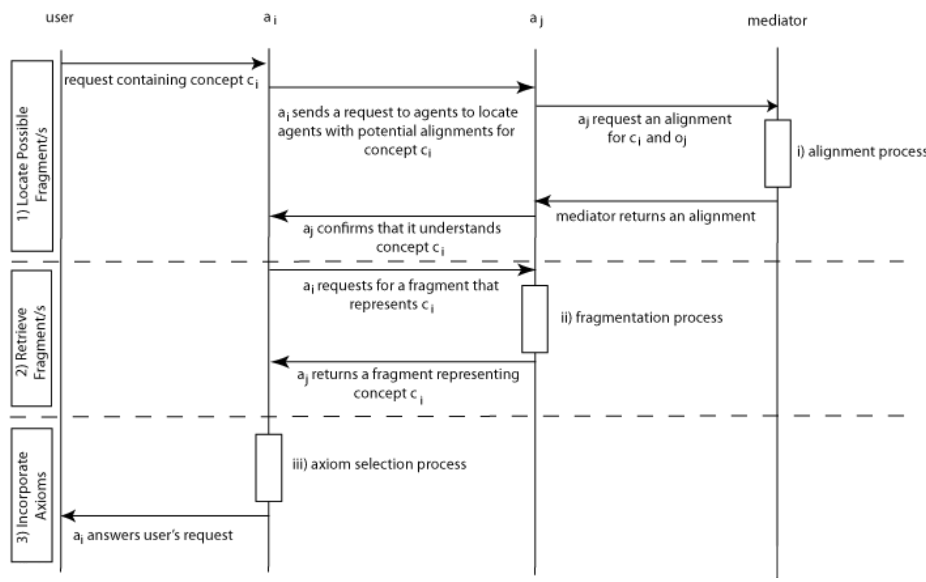

Fig. 2. General approach when an agent's ontology, o_i, does not contain the required knowledge.

4 An Agent Learning Approach

This section presents in detail our implementation that merges fragments collected from the *retrieving fragments* stage and selects which axioms to incorporate into an agent's ontology. Our main objective is to reduce the overhead cost of regularly acquiring the same knowledge for repetitive tasks, by augmenting an agent's ontology with a set of axioms representing the desired conceptual knowledge. In particular, we aim to maintain a similar granularity while augmenting an agent's ontology, to emulate the level of detail required for domain tasks. Specifically, the granularity of an ontology is dependant on an agent's purpose, and agents that contain the same conceptualisations may have a different level of detail. For example an agent's ontology contains knowledge about removal vehicles and models its vehicles with the concepts 'type', 'name', 'license plate', and 'capacity', whereas an agent designed to build these vehicles may contain 'manufacturer_location', 'vehicle_components', 'part_numbers', 'dimensions', and 'capacity'. The agent that builds these vehicles requires a greater level of detail than the agent that rents them. Therefore our approach aims to reflect the agent's purpose by emulating a similar level of detail. The following sections describe i) the merging of acquired fragments, and ii) the axiom selection technique (as shown in Figure 3).

4.1 Merging the Fragments

As previously stated, agent a_i retrieves a set of fragments, $S = \{s_1 \dots s_r\}$ where there are r number of segments, that represent concepts equivalent to x from the *retrieving fragment* stage. Our aim is to merge a subset of these fragments which contains semantic similarity to the agent's ontology, so that agent a_i can select a subset of axioms from the merged fragments. This merging process removes fragments that do not relate to the agent's domain, redundant and conflicting axioms,

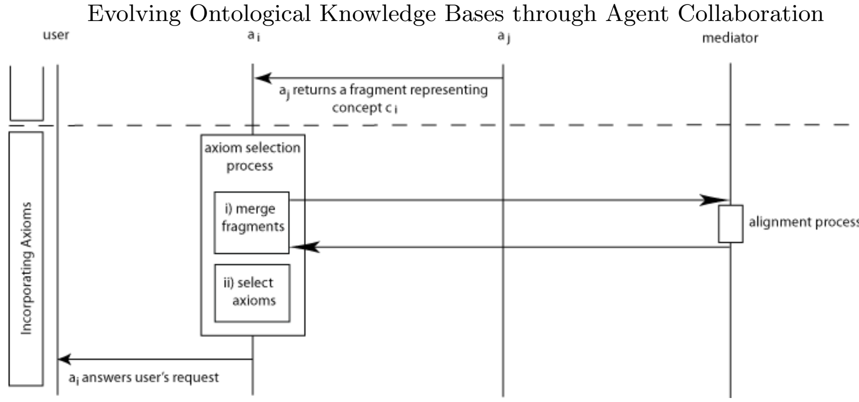

Fig. 3. Our incorporating axioms approach

and also attempts to resolve conflict with a 'majority ruling' methodology. We considered three alternative techniques: first, simply merging the fragments, although this technique does not resolve conflicts, and the fragments may by unrelated to the required domain; second, merging fragments that are determined to relate to the domain; third, merge fragments that relate to the agent's domain and select a set of axioms from all of these fragments that are consistent within itself and the agent's ontology that contain the most axioms. In contrast to the proposed third alternative, our chosen technique enables an agent to select a set of axioms, which is consistent within itself and the agent's ontology, that are represented by most agents. This enables agents to incorporate knowledge that is contained by a greater number of agents, and therefore can collaborate with a wider range of agents. The following process describes our chosen technique, and is used to merge the set of fragments representing the concept x_i:

– Each fragment $s_1 \dots s_r$ contained in S is compared with o_i, with the use of the mediator that provides a confidence rating comparing the individual semantic similarity between $s_1 \dots s_r$ and o_i. When the rating is below 0.7 then the corresponding fragment is removed from S. A threshold of 0.7 has been chosen to select strong correspondences between the two concepts. A trade-off between precision and recall determines the effect of the threshold, with a higher threshold providing a greater precision of match, at the expense of recall of further matches. This step aims to remove fragments that do not represent the semantics of x_i. For example, a_i's domain is 'animals' and requires knowledge about the species tiger, and requests for fragments about 'Tiger'. Agent a_j responds to a_i by sending a fragment representing 'Tiger_II', because a mediator mapped 'Tiger' to 'Tiger_II'. However, the fragment represents a World War II tank and not the mammal tiger, and by using the fragment and a_i's ontology it is possible to detect that the fragment contains no semantically equivalent concepts. We assume that a fragment will contain some correspondence to the ontology if they are domain related.
– The axioms from $s_1 \dots s_r$ are merged into a set of axioms M, such that $M = s_1 \cup \dots \cup s_r$. We create the powerset P of M, where $P = p_i \dots p_n$, in order to check consistency of the subsets with the agent's ontology, o_i, using a semantic reasoner. If a set from this powerset is found to conflict with the ontology, then it is removed from P. Prior to consistency checking, the sets in P are ordered to

minimise the number of consistency checks, such that if $m \subseteq M$ is inconsistent, then any other $m' \supset m$ is also likely to be inconsistent.

- The remaining sets belonging to P, are evaluated to select a set that represents x_i. Specifically, these sets are evaluated by calculating the average number of agents that contain the concepts in each set contained in P, and the set that contains the highest average will be selected. This is calculated with Equation 1, and the selected set will be used by the agent to select axioms to incorporate into its ontology.

The selection of a subset of M is dependent on the average number of agents containing the axioms L, see Equation 1. Then the powerset with the highest average L (i.e. the set which is contained by the most agents) will be selected to represent the concept x_i.

$$L_m = \sum_{x \in m} \frac{|\{a_i : a_i \in A \; and \; x \in o_i\}|}{|m|} \tag{1}$$

where A is the set of agents $\{a_1 \ldots a_n\}$, and o'_n is the ontology (set of axioms) of agent a_1.

This technique is used to select axioms that represent an agent's domain, and provide a set of axioms that does not conflict with an agent's ontology. Merging fragments using this technique to perform consistency checking is not optimised for scale, due to being implemented using a basic brute force methodology, which is sufficient for our small-scale examples. Thus, the optimisation of this aspect of our approach is suggested as future work.

4.2 Axiom Selection

The above technique provides a fragment containing a shared set of axioms that represent x_i, this technique enables a_i to choose a set of axioms from this fragment. Our selection method is similar to Seidenburg et al.'s [16] ontology segmentation technique, in that we both consider the role of hierarchical and relational classes. In contrast to Seidenburg et al.'s approach, which focuses on reducing the overall size of an ontology by selecting those axioms relating to one specific concept, our approach aims to reduce the number of axioms used to describe a specific concept. As our approach also considers how the incorporated knowledge relates to the domain we have decided to instantiate the domain knowledge in a Domain Ontology (DO), and the knowledge that is acquired by our technique into an Incorporated Knowledge Ontology (IKO). In particular, our IKO will import the DO, our reasons for using this approach are two fold: first, it enables the agent to weight the values contained in the DO higher than the IKO during the selection process, this endeavours to maintain the agent's intended domain so that the agent's ontology does not evolve to represent a different domain; second, it enables our approach to later consider when an agent should remove incorporated knowledge due to identified costs of storing the knowledge. In order to select axioms, our technique enables an agent to analyse which axioms from the fragment relate to its domain. Our approach uses two steps to analyse these axioms, i) **hierarchical**, and ii) **relational** axiom selection techniques.

The **hierarchical selection** technique aims to reduce the number of superclasses that are used to represent x_i when s_j has a larger hierarchical depth than o_i. Specifically, agent a_i's aim is to incorporate knowledge that is useful for domain related tasks, and therefore does not require information that is too far removed, structurally, from this interest domain. Therefore a_i aims to minimise the depth of information taken from the fragment. If classes that are excluded by this process are required for later tasks, the agent reiterates the 'learning' technique to incorporate them. In particular, the agent selects which concepts to incorporate by using weightings to determine which classes relate to the agent's DO and IKO. Specifically, the agent uses a weight c to rate the classes that are equivalent to those contained in the DO, this ensures that the incorporated knowledge aligns with the domain of interest of the agent, with minimal deviation into the domain of the IKO. These classes are rated according to the number of axioms that relate to concepts in the DO and IKO. The following equation is used to rate each hierarchical concept, to determine which concepts will be incorporated into an agent's ontology.

$$concept\ rating = c * nDO + nIKO \tag{2}$$

where c is a constant that represents the weighting given to the concepts relating to DO's knowledge, nDO is the number of times the concept is related to DO's knowledge, and $nIKO$ is the number of times the concept relates to the concepts in the IKO.

In particular, in this scenario, there are two possible cases: In the first case, the hierarchical depth of s_j is greater than o_i, and the second that s_j has a hierarchical depth that is less than or equal to o_i. In the first case, the depth of the ontology has a greater difference of hierarchical depth, thus our approach aims to reduce the hierarchical depth. In order to achieve this the concepts are rated using Equation 2 and are then ordered into sets of hierarchical depth. For example, all subclasses of the root class are grouped as level one and their subclasses are grouped as level two, continuing until the leaf nodes are reached. Then the average concept rating for each hierarchical depth can be calculated. The agent then selects a number of hierarchical layers that have the highest average value to incorporate into an agent's ontology, provided that one of the selected set contains the desired knowledge. In the case that these selected sets do not contain the required knowledge then the set with the lowest average will be replaced with the highest average level that contains the desired knowledge. Specifically, the number of hierarchical layers selected is defined by calculating the median value of the hierarchical depths of o_i and s_j. This is designed to enable the agent's ontology to expand, while recognising the granularity of the detail expressed in its ontology and the granularity required to express the new knowledge (as discussed in Section 4). In the second case, where the hierarchical depth of s_j is less than or equal to o_j, the agent aims to keep that depth of hierarchy while selecting concepts related to its ontology. Subsequently, the agent rates the concepts using Equation 2, such that concepts that have a rating greater than zero will be incorporated into o_i. Pseudo-code for above selection technique is provided in Algorithm 1.

Both of these cases create a set of axioms, H, that represent x_i. This technique aims to replicate the granularity in o_i, in order to maintain the level of depth of the agent's particular interest in the domain. This enables the agent to represent the desire knowledge with a similar depth as o_i, and the incorporated knowledge acts as base for including domain knowledge. This approach minimises the fragment's depth and aims to incorporate a similar level of detail contained within o_i. For example agents interested in different subclasses will have a larger depth and granularity than those with a lower level of hierarchy. It also encourages agents to learn smaller fragments of ontologies, allowing the evolution to be controlled and maintained.

Algorithm 1 Hierarchical Selection technique, where N is the hierarchical depth of the segment, I is the number of depths of classes to incorporate and c is the concept to incorporate.

```
if depth of segment > depth of ontology then
  for each class in segment do
    calculate concept rating for class
    put class and concept rating into set(N) that corresponds to its depth level
  end for
  for each depth level set(N) do
    calculate average rating of depth level
  end for
  order sets by average rating, descending
  subsort sets by number of classes in each depth level, descending
  select highest I number of sets
  if highest I number of sets does not contain c then
    replace lowest value level with highest value level that contains c
  end if
  incorporate each class and their properties in the highest I into a fragment
else
  for each class in segment do
    calculate concept rating for class
    if concept rating of class > 0 then
      incorporate class and its properties into a fragment
    end if
  end for
end if
return  fragment containing classes and properties
```

Relational selection aims to limit the number of relationships connected to concept x_i, to be incorporated from s_j into the IKO of o_i. The number of properties have already been reduced by the *hierarchical selection* technique; and the *relational selection* process will reduce the number of incorporated properties from H, the concepts selected during the *hierarchical selection*. The properties relating to axioms from H will be 'pruned' by the distance of properties in 'hops' away from x_i; the threshold t will be used to remove concepts that are more than t 'hops' away from x_i. This process ensures that the properties to be incorporated into o_i are closely related to x_i, and the domain of interest of the agent.

Once the merging of the sourced axioms from the other agents, and hierarchical and relational selection process have been performed, the selected axioms represent a shared set of axioms describing the concept x_i. This set of axioms is a fragment,

f_i, and it will be merged with a_i''s ontology, such that $o_i' \equiv o_i \cup f_i$. Pseudo-code for this selection process is provided in Algorithm 2.

Algorithm 2 Relational Selection technique, where c is the concept to incorporate, t is the distance threshold of axioms to incorporate and *Nodes* is a set of concepts to be returned.

```
Nodes = {c}
Add the path from c to the root concept, to Nodes
D = {c}
for n = 1...t do
   Retrieve all properties that relate to D, and find all classes D relates to
   Store classes that are related to D in an array
   tmp = ∅
   for all classes Class that related to D do
      Get the path from Class to the root
      Add the classes into Nodes
      Add Class to tmp
   end for
   D = tmp
end for
return Nodes
```

5 Illustrative Example

This section provides a use case scenario, in which we walk through each step of our approach. Agent a_1 has been designed to answer domain queries about 'removal services', and its ontology contains concepts such as 'desk', 'filing_cabinet' and 'office_chair'. In our example case, a_1 receives a request from a science park to remove items from several laboratories, and move them into a new biology laboratory, including a set of biological samples. Specifically, the agent receives the request `REMOVE biomedical_samples refrigerate`, in the format `REMOVE <item> <vehicle_requirements>`. The concept *refrigerate* is not contained in a_1's ontology, and a_1 requires this knowledge to complete this task. The system contains two other agents, a_2 and a_3, which represent two vehicle hire companies that rent vans used by the removal services provided by a_1. In order for a_1 to organise transportation for biological samples, a_1 can incorporate the required additional knowledge contained in agents a_2 and a_3's ontologies to reduce the number of messages sent to the mediator. The following list details the actions taken by a_1, as shown in Figure 2:

1. Agent a_1 queries all agents in the environment with a message that contains the concept *refrigerate*. This query aims to ascertain which agents contain knowledge related to the concept *refrigerate*.
2. Agents a_2 and a_3 consult the mediator as their ontologies do not contain the concept 'refrigerate', and the mediator returns a mapping between 'refrigerate' and 'Refrigerated_Van', and 'refrigerate' and 'Refrigerated_Vehicle'.

3. Agents a_2 and a_3 reply to a_1 because their ontologies contain the concept *refrigerated_van* and *refrigerated_unit*, respectively. These concepts are lexically similar to the concept *refrigerate* because they both contain the lexical stem 'refrigerate'; this match is determined by the mediator.
4. Agent a_1 requests a set of axioms that represent *refrigerated_van* and *refrigerated_unit*, from Agents a_2 and a_3 respectively.
5. Agent a_2 sends a message to a_1 containing a fragment, s_1, representing *refrigerated_van.*
6. Agent a_3 sends a message to a_1 containing a fragment, s_2, representing *refrigerated_unit.*
7. Agent a_1, receives the fragments s_1, and uses the mediator to compare its ontology, o_1, with s_1 and s_2. Both of these fragments are found to relate to o_1, as o_1 contains the concepts *van* and *vehicle*. More specifically, s_1 contains the symbol 'van' in *refrigerated_van*, thus creating a lexical match with o_1. Additionally, s_1 and s_2 contain the concept *vehicle*. Thus, both fragments are determined to relate to the domain of o_1.
8. Agent a_1 then checks that the axioms contained in s_1 and s_2 do not conflict with its own ontology. Once the agent has collated the axioms that do not conflict a_1 merges the fragments. In our case, none of the axioms contained in these fragments conflict. Therefore, a_1 merges s_1 and s_2.
9. Agent a_1 then selects which axioms to incorporate into its ontology that represent *refrigerate.*
   (a) Firstly, a_1 uses the hierarchical selection technique (as described in Section 4.2), to reduce the number of concept levels in a fragment. Specifically, a_1 calculates the highest node depth in o_i and Ax_1, where o_i's depth is four and Ax_1's depth is six. The median of these values is five, therefore a_1 aims to incorporate a set of axioms that have a depth of five. Subsequently, each hierarchical concept is associated with a *concept rating*, which is a measure that is calculated by the number of times that the concept is related to items in the DO and IKO. In this case, the $DO \cup IKO$ is equivalent to o_1, as a_1 has not incorporated new knowledge into its ontology. These ratings are then organised into sets of concepts with the same hierarchical depth. This enables an agent to determine the average concept rating for each layer of the hierarchy. Then a_1 selects the five hierarchical layers with the highest average concept rating to incorporate into o_1. Given the average concept ratings presented in Table 1 the concepts from levels 0, 1, 2, 3 and 4 have been selected to be incorporated into o_1.
   (b) Secondly, a_1 attempts to limit the number of relational axioms, although in our case 'refrigerated_van' and 'refrigerated_unit' do not have any relationships. Therefore it cannot decrease the number of axioms used to represent 'refrigerate'.
10. Agent a_1 incorporates the axioms selected into o_1.

Ultimately, as a_1 has augmented its ontology with additional knowledge, it no longer requires the mediator to organise a rental of a refrigerated van from either a_2 or a_3. In order to achieve this, a_1, a_2 and a_3 send a total of six messages to the mediator, however no additional translations are required from the mediator

Level	Concepts	Values	Average
0	{ transportation , documents , employees, customers }	{ 1 , 0 , 1 , 1}	0.75
1	{ vehicles , vehicle }	{ 1 , 1 }	1
2	{ road_vehicles , van , lorry , roof_rack , trailer , car ,motorcycle , van , lorry , roof_rack , trailer , car, motorcycle , road_vehicles }	{ 1 , 1 , 1 , 0 , 0 , 1 , 0 , 1 , 1 , 0 , 0 , 1 , 0 , 1 }	0.57
3	{ commercial_vehicles , refrigerated_unit , large_capacity , medium_capacity, small_capacity , heavy_goods, medium_van, small_van , heavy_goods_van , large_van , refrigerated_van , rrefrigerated_unit , large_capacity , medium_capacity , small_capacity , heavy_goods medium_van , small_van , heavy_goods_van , large_van, efrigerated_van , commercial_vehicles }	{ 1 , 1 , 0 , 0 , 0 , 0 , 1 , 1 , 1 , 1 , 1, 0 , 0 , 0 , 0 , 1 , 1 , 1 , 1 , 1 , 1}	0.64
4	{ van , van }	{ 1 , 1 }	1
5	{ medium_van , small_van, heavy_goods_van, large_van, refrigerated_van , refrigerated_unit, large_capacity , medium_capacity, small_capacity, heavy_goods, medium_van , small_van, heavy_goods_van , large_van , refrigerated_van, refrigerated_unit, large_capacity , medium_capacity, small_capacity , heavy_goods }	{ 1 , 1 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 }	0.5

Table 1. For each depth level in the hierarchy of s_j, the average of all of the values has been calculated, for use in the hierarchical selection technique.

to perform repeated requests for another refrigerated vehicle. To obtain additional knowledge the collaboration required a total of ten messages, as the agent initiated four collaborations (as there are four other agents in the environment), to which two agents responded. Therefore, subsequent requests for a 'refrigerated' vehicle requires two messages.

6 Results

Our preliminary results are presented in this section, the above scenario and approach have been developed using Java and OWL-API[3]. These preliminary results are initial observations, and do not analyse the possible overheads of acquiring knowledge, such as the cost of acquiring knowledge, and the incorporated knowledge model's complexity and its inference costs. However, these results provide details about the number of messages and axioms that are required to acquire knowledge that enables an agent to answer an domain query, and show that augmenting an agent's ontology can reduce these measures. , We consider two cases: first where an agent augments its ontology using our approach (the three stages presented in Figure 2); and second acquiring knowledge without augmenting an agent's ontology (represented by the first two stages in Figure 2). Specifically, our domain query focuses on acquiring knowledge related to 'refrigerate', to enable an agent to rent a refrigerated vehicle from two agents. This query requires agent a_1 to acquire knowledge related to 'refrigerate' and the acquisition process yields the number of messages and axioms sent during this transaction. In order to show the trends, we have presented these results in two cumulative results tables (see Tables 2 and 3). In particular, these results are based on collaboration between three agents (as described in Section 5) and it is assumed that the mediator responds with one alignment between concepts.

Table 2 shows the cumulative number of message pairs required to augment an agent's ontology for ten iterations of the transaction described above. The cumula-

[3] The OWL API, `http://owlapi.sourceforge.net/`

	Number of iterations of task									
	1	2	3	4	5	6	7	8	9	10
Cumulative number of pairs of messages using our approach to augment an ontology	10	12	14	16	18	20	22	24	26	28
Cumulative number of pairs of messages without augmenting an ontology	6	12	18	24	30	36	42	48	52	58

Table 2. Cumulative results showing the number of messages required to acquire knowledge.

tive number of message pairs is shown when the agent incorporates new knowledge using our approach, and when the agent does not incorporate knowledge.

The results show that the messages required to incorporate knowledge has a higher cost than only acquiring the knowledge from the mediator. However, in contrast the cost is reduced in the case of augmenting an ontology if the same transaction is repeated. Our approach does not need the services of a mediator after it has incorporated new knowledge. However, the agent which does not incorporate knowledge requires regular collaboration with the mediator. This increases the cost of collaboration, as the number of messages approximately doubles for each repeated transaction.

	Number of iterations of task									
	1	2	3	4	5	6	7	8	9	10
Cumulative number of axioms in messages using our approach to augment an ontology	539	539	539	539	539	539	539	539	539	539
Cumulative number of axioms in messages without augmenting an ontology	199	398	597	796	995	1194	1393	1592	1791	1990

Table 3. Cumulative results showing the number of axioms that are passed in messages required to acquire knowledge.

A similar trend can also been seen in Table 3, whereby with each additional transaction requires a greater number of axioms to interpret the desired knowledge when an agent does not augment its ontology. Specifically, our ontologies used in this investigation contain an average of ninety seven axioms and we assume that these figures would increase proportionality with the number of axioms contained in the agent's ontologies.

Given these results, we have shown that augmenting an agent's ontology with additional results can reduce the number of messages and number of axioms sent in these messages, for repeated tasks.

7 Summary

We have presented an approach to incorporate additional knowledge to augment an agent's ontology. Our results have shown that incorporating knowledge reduces the number of messages required to acquire knowledge that is needed to repeat regular transactions. Additionally, our results identify the number of axioms sent in each of these messages and has shown that by incorporating additional knowledge we reduce the number of axioms sent in messages.

Our preliminary results and scenario will be used in future work to investigate how augmenting an agent's ontology with a set of axioms representing desired knowledge can decrease the number of axioms sent in messages. We want to make the selection process more sophisticated so that agents can reduce the cost of incorporating knowledge. Thus, we will be investigating how to search for related concepts to those incorporated that might be useful for future tasks, and that if by incorporating those related concepts, an agent can avoid having to learn them in future tasks. We hypothesise that our agents' vocabularies will converge and therefore will require a small number of axioms to define desired knowledge. In this paper we have also identified two other areas which we would like to investigate:

first, we will look at optimising the consistency checking methodology, such that the current brute force checking technique described in Section 4.1 can be reduced in complexity; and second, consider when to remove knowledge that has been incorporated into an agent's ontology. In particular, we will investigate the overhead costs trade-offs involved with storing knowledge, against the possible costs of removing and reincorporating knowledge that has not been used of a period of time.

References

1. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition **5(2)** (1993) 199–199
2. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag New York, Inc. Secaucus, NJ, USA (2007)
3. van Diggelen, J., Beun, R., Dignum, F., van Eijk, R., Ch, J.: Meyer, Optimal Communication Vocabularies and Heterogeneous Ontologies. Agent Communication: International Workshop on Agent Communication (AC 2004)(RM van Eijk, M.-Ph. Huget & F. Dignum,, eds.), LNAI **3396** (2004) 76–90
4. Van Eijk, R., De Boer, F., Van Der Hoek, W., Meyer, J.: On dynamically generated ontology translators in agent communication. International Journal of Intelligent Systems **16**(5) (2001) 587–607
5. Doherty, P., Łukaszewicz, W., Szałas, A.: Approximative query techniques for agents with heterogeneous ontologies and perceptive capabilities. The 9th International Conference on Principles of Knowledge Representation and Reasoning (2004)
6. Bailin, S., Truszkowski, W.: Ontology negotiation between intelligent information agents. The Knowledge Engineering Review **17**(01) (2002) 7–19
7. Afsharchi, M., Far, B., Denzinger, J.: Ontology-guided learning to improve communication between groups of agents. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan (2006) 923–930
8. Wiesman, F., Roos, N.: Domain Independent Learning of Ontology Mappings. Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, New York City, New York, USA (2004) 846–853
9. Soh, L.: Multiagent, Distributed Ontology Learning. Working Notes of the 2nd AAMAS OAS Workshop, Bologna, Italy (2002)
10. Flouris, G., Plexousakis, D., Antoniou, G.: Updating Description Logics using the AGM Theory. In: Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning. (2005) 69–76
11. Flouris, G., Plexousakis, D., Antoniou, G.: Evolving Ontology Evolution. LNCS **3831** (2006) 14
12. Flouris, G., Huang, Z., Pan, J., Plexousakis, D., Wache, H.: Inconsistencies, Negations and Changes in Ontologies. In: Proceedings of the National Conference on Artificial Intelligence. Volume 21 (2006) 1295
13. Alchourron, C., Gaerdenfors, P., Makinson, D.: On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. The Journal of Symbolic Logic **50**(2) (1985) 510–530
14. Haase, P., Stojanovic, L.: Consistent Evolution of OWL Ontologies. Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece (2005)
15. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A Framework for Handling Inconsistency in Changing Ontologies. LNCS **3729** (2005) 353
16. Seidenberg, J., Rector, A.: Web ontology segmentation: analysis, classification and use. In: Proceedings of the 15th International Conference on World Wide Web, New York, NY, USA, ACM (2006) 13–22