Ontology Evolution through Agent Collaboration

Heather S. Packer and **Nicholas Gibbins** and **Nicholas R. Jennings** [1]

Abstract. We present a technique that enables a software agent to augment its ontology with domain related concepts by collaborating with other agents. The collaborating agents have their own individual ontologies, they can share concepts and relationships that relate to a requested specific concept (which is known as a fragment). Thus, specifically, our technique selects the fragments that will be shared. This approach enables agents to answer queries with more range and detail, and it also enables an agent to infer new exploitable knowledge. Without this capability, an agent may be limited by its domain model, and cannot reflect changes in the environment. Through empirical evaluation, we show that our technique reduces the cost of acquiring concepts that are regularly used (compared with learning nothing) and reduces the complexity of the agent's ontology by augmenting it with selected concepts and relationships which are related to its domain (compared with learning everything).

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 augment its vocabulary used to describe its knowledge base, it can use its terminology to communicate with other agents, and answer queries that it could not previously. In contrast to augmenting an agent ontology, the agent could retrieve the entire vocabulary required for communicate with another agent. This becomes inefficient when the agent interacts with the same agent more than once. In our context, our agents use an ontology to model their vocabulary and their knowledge base. Specifically, we take an ontology to be a formal structure that models concepts, relationships and entities. Then, the ability to evolve an agent's vocabulary enables it to reflect its environment, and ensure that its ontologies do not have to be remodelled in response to environment changes. However, such augmentation may incur costs, relating to the acquisition process and the search time required for inferring logical consequences from an agent's ontology. For this reason, our proposed technique attempts to reduce this by selecting the concepts and the neighbours to share with in an informed manner.

In more detail, this approach supports the automatic exchange of knowledge to augment agent based problem solving. It provides a method that reduces the complexity of an ontology compared with augmenting an agent's ontology with all knowledge in the environment, and selects domain-specific concepts that relate to an agent's ontology. While other techniques, such as those detailed below do augment an ontology, we focus on the analysis of an agent's ontology and compare it to a fragment that contains a shared representation of a concept. A query agent can therefore evaluate the relationship between each concept from the fragment and its ontology's domain. This approach enables agents to reduce the cost of mediated transactions by augmenting their ontology on demand.

We propose that our approach is appropriate for agents that provide services about a domain, and have a small ontology (approximately 50-200 concepts) that models an incomplete domain[2]. In our context, a service is the ability to provide and complete requests related to a specific domain. Agents in an environment that provide different services can benefit from incorporating new knowledge from other agents where their interest domains intersect. For example, suppose a set of emergency services have intersecting knowledge about the domain 'rescue'. When an emergency service agent (or query agent) cannot perform a rescue task alone, it can incorporate new domain knowledge from other specialist agents which support this task. In more detail, a specialist agent can send fragments to other agents about concepts in its ontology. For example, consider a query agent that requires a vehicle that can remove heavy rubble so that the hospital emergency service can rescue casualties from a collapsed building. In this case, the hospital emergency service agent's ontology is unlikely to contain knowledge about vehicles that can remove rubble. However, it is possible to learn this new information from a specialist agent so that it can organise the removal of the rubble.

The approaches presented by Bailin and Truszkowski [1], Afsharchi et al. [2], Wiesman and Roos [3], and Soh [4] enable their agents to augment their ontologies with new knowledge, when agents have different domain models representing the same domain. In contrast to our approach, these approaches require their agents to model the same domain. In particular, Bailin and Truszkowski's approach considers semantically equivalent representations, and Afsharchi et al. and Soh focus on the validation of the knowledge to be incorporated into the agent's ontology. These approaches augment an agent's ontology, however they augment their agent's ontology one concept at a time, which increases the overhead cost of retrieving the information. In addition to agent-based research, the Semantic Web community has produced work on evolving ontologies. In particular, the techniques presented by Flouris et al. [5] enable the evaluation of coherence and consistency. The work presented by Hasse and Stojanvoic [6] further explores the issue of consistency, by proposing techniques to resolve three types of inconsistency; structural, logical, and user defined inconsistencies. These techniques can be used with our approach in order to evaluate, locate and resolve inconsistent knowledge to be incorporated into an ontology.

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. and Soh, 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 collab-

[1] School of Electronics and Computer Science, University of Southampton, United Kingdom, email:{hp07r, nmg, nrj}@ecs.soton.ac.uk

[2] An ontology that has an incomplete domain model does not model all concepts associated with its domain.

orating agents' ontologies. In contrast to the above approaches we augment our agent's ontology with a fragment of knowledge, as opposed to a single concept, in order to reduce the complexity of the knowledge that is required. This aims to satisfy our objective to reduce the overhead cost of regularly acquiring knowledge for repetitive queries. This approach is described in more detail in [7].

Our paper is structured as follows. In Section 2 we present an outline of how our query agent retrieves fragments, and in Section 3 we describe our technique to select concepts and relationships to augment a query agent's ontology. Then, in Section 4 we present our empirical evaluation of our technique. We conclude in Section 5.

2 Retrieving the Fragments

As previously discussed, our proposed approach enables an agent to evolve its ontology with new concepts related to its domain, so that it can collaborate and reduce the cost of inference associated with complex ontologies. In order to evolve a query agent's ontology, the agent is required to retrieve a fragment from a specialist agent's ontology modelling the required concept. To this end, Figure 1, shows three types of agent: 'query agent', 'specialist agent', and 'mediator'. We note three things about these agents. First, the query agent refers to its own ontology, which describes a specific domain. However, it models the domain only with partial knowledge due to its design and/or changing requirements. Second, the specialist agents also refer to their own ontologies, each of which specialises on a domain that intersects with the domain of the query agent. Third, the mediator can provide translation mappings, for the specialist agents, between a concept and a set of concepts.

In this scenario, the query agent is sent queries from a user. When the query agent receives a query that contains a concept that is not contained within its ontology, the query agent attempts to learn this new vocabulary from the specialist agents so that it can answer the query. In particular, figure 1 shows the interaction between the three agents where the query agent is required to learn a concept. In step 1 (and given our motivating example) a user sends our emergency rescue query agent a query to locate a vehicle that can move rubble from a collapsed building. This query contains a concept, in this example '`forkLiftTruck`', that the query agent ontology does not contain. In step 2, a request is broadcast to all specialist agents in the environment for knowledge about '`forkLiftTruck`'s. In step 3, the specialist agents request a translation between its concepts and the unknown concept, '`forkLiftTruck`'. This requested translation is sent in step 4. In this case only one specialist agent's ontology contains an equivalent concept to the unknown concept, '`forkLiftTruck`', and it sends this confirmation to the query agent in step 5. The query agent then requests the description in the form of a fragment based on the unknown concept, '`forkLiftTruck`', and the specialist agent responds with this fragment in steps 6 and 7. This fragment contains the concepts '`vehicle`', '`liftingCapacity`', '`building`', '`reachTruck`', '`handPalletTruck`', and '`truckMountedForklift`'. Once the query agent receives the definition of a '`forkLiftTruck`' it answers the user's query, shown in step 8. This enables the query agent to find a vehicle that can remove building rubble. Given this background, our approach focuses on the mechanism to select axioms to include into the query agent's ontology; additionally we also focused on the performance of the acquisition.

The query agent's approach is designed to select a set of concepts and relationships from a set of fragments that describes the required concept to augment into its ontology. This selection of concepts enables the agent to choose domain related knowledge, and limits the amount of knowledge augmented by the agent. The limitation of knowledge aims to reduce the potential reasoning complexity of an agent's ontology because the more complex an ontology is the more time is required to process inferred knowledge. It is important to note that this approach does not preclude the ability of the agent to complete complex tasks, as any required knowledge is never excluded. While evolving an ontology can benefit an agent as it can infer logical consequences from its ontology, a complex ontology can affect the time it takes to infer knowledge. This cost can be exponential depending on the structure and features contained within the ontology [8].

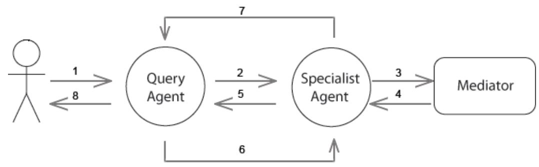

Figure 1. A sequence of messages between a user, query and specialist agent, and a mediator.

3 Collaborative Evolution of the Ontology

As discussed in the previous section, the query agent can obtain more than one fragment that possibly represents the required concept. Our main objectives are to reduce the overhead cost of regularly acquiring the same knowledge for repetitive queries, and reduce the complexity of using all the knowledge that is acquired. In order to do this we augment the query agent's ontology with a set of selected concepts and relationships representing the desired knowledge, and these components are stored as axioms in a query agent's ontology. In particular, we aim to maintain a similar granularity to the query agent's ontology while augmenting its ontology, to relate to the level of detail required for domain queries; this technique prevents an exponential growth of the depth of the agent's ontology, and increases the chance of retaining irrelevant concepts and relationships. Relating to the purpose of the agent, the granularity of an ontology is dependant on the agent's purpose, and agents that contain the same conceptualisations may have a different level of detail. For example, in our motivating scenario where a query agent wants to remove rubble, the agent requires information about only vehicles that can be used in rescue situations and will not need information about a '`reachTruck`' and '`handPalletTruck`' as these vehicles are for commercial environments. In order to achieve this we have a two stage process: merging the fragments; and selecting concepts and relationships from the merged fragment.

3.1 Merging the Fragments

The query agent can retrieve more than one fragment containing axioms relating to the requested concept. Our aim is to merge a subset of these fragments which are semantically similar to the agent's ontology, so that the query agent can select the 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. This enables the query agent to incorporate knowledge that is contained within a greater number of agents, and therefore collaborate with a wider range of neighbours. The following process describes our chosen technique, and is used to merge the set of fragments:

1. Compare each fragment to the query agent's ontology, by requesting mappings from a mediator. If the fragment contains one other concept that matches the agent's ontology then it is deemed that the fragment and ontology have the same domain. In our example, the query agent's ontology concept 'capacity' maps to '`liftingCapacity`' which is contained in a retrieved fragment, and the concept '`vehicle`' is already contained in the query agent's ontology. Therefore the fragment detailed in Section 2 is deemed to relate to the query agent's domain. This step aims to remove fragments that do not represent the semantics of the requested concept. We assume that a fragment will contain concepts related to the ontology if they are domain related.
2. Generate the powerset of all axioms in the selected fragments. Discard sets which are inconsistent with the query agent's ontology, and select the largest set that is contained by the largest average number of agents.

This technique is used to select axioms that represent an agent's domain, and provide a set of axioms that do not conflict with the query agent's ontology.

3.2 Selecting Concept to the Merged Fragment

The above technique provides a merged fragment that represents the requested concept, so that the query agent can select a set of axioms from this fragment. From this, the selection component needs to select a set of axioms to use to augment the agent's ontology. To do this, we adopt a selection method that is similar to Seidenburg et al.'s [9] ontology segmentation technique, in that we both consider the role of hierarchical and relational classes. However, 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. This retrieves a fragment that describes the context of a concept; this context is used for validating and not all of this context is required by the query agent. In order to maintain the query agent's knowledge, we model acquired axioms in a separate ontology, which is imported into the agent's instantiated ontology. This enables an agent to infer new and exploitable knowledge from its instances, and enable it to determine its instantiated concepts. For instance, in our example augmenting the query agent's ontology with a fragment about 'forkLiftTrucks' enables the agent to infer that a vehicle in its knowledge base can also lift the same amount as a forklift truck and therefore may be offered as a substitute in a critical situation. Our approach uses two steps to analyse these axioms: (i) **hierarchical**, and (ii) **relational** axiom selection techniques.

The **hierarchical selection** technique reduces the depth of a fragment, and provides the relational axiom selection technique with a set of concepts it can select from. Our hierarchical selection technique aims to reduce the depth of the fragment only if the depth of the query agent's ontology is smaller than the fragment's depth, otherwise it enables the relational selection algorithm to select from all of the fragment's levels. This technique calculates the mean average depth (from the root node to lowest child node) of the query agent's ontology and the fragment, and uses this to select the number of levels of classes to be selected from the fragment. Once the number of required levels has been selected, we calculate the number of times each concept in the fragment is referred to in the query agent's ontology. We then calculate the average number of times the concepts are referred to in each axiom, for each depth in the fragment. The selection process selects the number of required levels by selecting those with the highest average concept rating.

Relational selection aims to limit the number of relationships connected to the required concept. The number of properties and possible concepts have already been reduced by the *hierarchical selection* technique. The properties will be 'pruned' by the distance of properties in 'hops' away from the required concept, using a set threshold. This process ensures that the properties to be incorporated into the query agent's ontology are closely related to the required concept and its domain.

Once the hierarchical and relational selection processes have been performed, the selected axioms represent a shared set of axioms describing the required concept. These selected axioms are then added to the query agent's ontology.

4 Empirical Evaluation

In order to evaluate our approach, we performed an empirical investigation. We modelled a scenario where a query agent aims to answer a user's queries which relate to its ontology's domain about 'emergency rescue'. Our investigation compares four alternative approaches for the query agent, each of which processes the retrieved fragments (see Section 2). Our testing environment comprised of five query agents, one of which uses our approach (see Section 3), and four other query agents. These four query agents utilise the following four learning techniques:

1. *Learn-repeated* approach: learns all concepts and their relationships which are required more than once. This aims to offset the cost of learning concepts by considering how much the required concept is used.
2. *Learn-connected* approach: learns all axioms from fragments that are directly connected to the concept being queried. This is a comparable technique with the agent approaches discussed in Section 1.
3. *Learn-all* approach: learns all axioms in all of the fragments it locates. This technique is used to show that our approach's query agent has a lower ontology complexity than a query agent that learns everything.
4. *Learn-nothing* approach: learns nothing. This technique is used to show that our approach's query agent has a lower retrieval cost than a query agent that learns nothing.

Each of the query agents is instantiated with its own copy of the same ontology, and is given the same list of queries. This list of queries is generated by selecting a set of concepts related to a specific concept in the domain of 'emergency rescue' in sets of five, until there are one hundred queries. Also in the environment are ten specialist agents, five of which contain their own ontologies with the domain of 'emergency rescue', and the other five contain ontologies with domains unrelated to 'emergency rescue'.

The emergency rescue domain ontologies were derived from the AKTiveSA ontology[3] by the e-response project[4], which publishes ontologies for emergency rescue in the domains of Fire, Ambulance and Police. The non-emergency rescue ontologies are Beer[5], Brain

[3] AKTiveSA Ontology: `http://sa.aktivespace.org/aktivesa`
[4] e-Response Ontologies: `http://e-response.org/ontology/`
[5] Beer ontology: `http://www.purl.org/net/ontology/beer#`

Atlas[6], Bug reports[7], Charly Air Service[8] and Food[9]. After our agents have processed the one hundred queries, we calculate the costs involved for completing each query. These costs are measured in milliseconds, and include the time taken to: send messages across a network with a bandwidth of 2Mb, process an alignment, generate a fragment, and augment an agent's ontology. We also measure the complexity of the agent's ontology, this measure is calculated by the number of CPU clock ticks required to load and consistency check an ontology, using the Pellet reasoner[10]. In order to evaluate our approach we repeat our investigation fifty times to obtain a statistically significant average of each iteration. The results of our investigation are illustrated in Figures 2 and 3, which show the total cost (as described above) and ontology complexity, respectively. A confidence interval of 95% is indicated with error bars for each query.

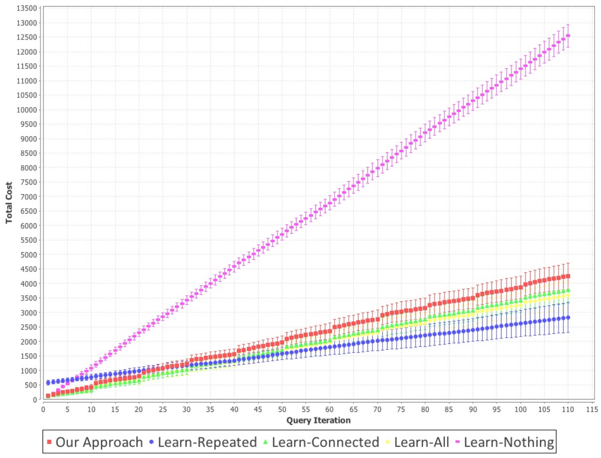

Figure 2. Cumulative total cost of alignment

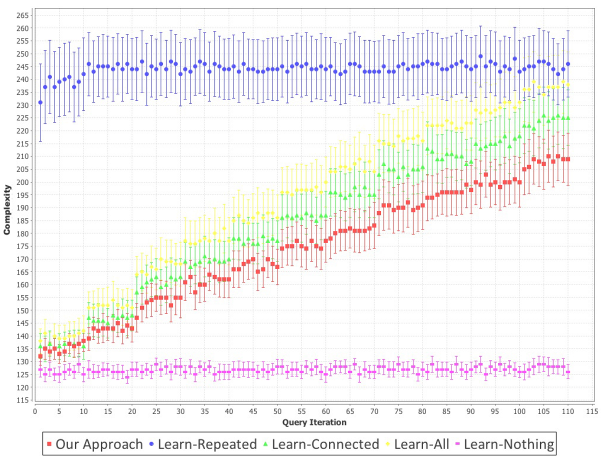

Figure 3. Ontology complexity

These figures identify four notable results. First, the results for the 'Learn-Nothing' approach demonstrate that while this technique keeps the ontology complexity consistently low, the total cost is highest of all techniques. Second, after all queries are completed, the ordering of the alternative query agent approaches is exactly reversed between the total cost and the ontology complexity. This illustrates that these two metrics are inversely linked, therefore we note that there is a trade off between the cost of selecting a fragment to augment a query agent's ontology and the complexity of a query agent's ontology. Third, that the overhead cost of learning specific repeated fragments, as used in the 'Learn-Repeated' approach, becomes profitable after approximately twenty five queries. This is because our approach does not know which queries will be repeated and the 'Learn-Repeated' knows of all repeated queries. Fourth, that the costs resulting from alternative approaches 2 and 3 are similar because their query agents communicate the same number of times.

When taken together, this shows that the 'Learn-Connected' and 'Learn-All' approaches have similar mean costs. More over, these alternative approaches have a lower mean total cost than our method. However, our approach's cost is within their confidence internal. Also, our approach has a lower mean complexity than these approaches because it selects concepts to augment into the query agent's ontology that relate to the its interest domain. This means our approach enables an agent to reduce the cost to acquire concepts that are regularly required, when compared to the 'Learn-Nothing' approach. It also reduces the complexity of our query agent's ontology by augmenting it with selected concepts and relationships, compared with the 'Learn-Repeated', 'Learn-Connected' and 'Learn-All' approaches.

5 Conclusions

We have presented a technique that enables an agent to automatically retrieve knowledge and augment its ontology in order to reduce the cost of acquiring regularly required concepts. Our technique enables a query agent to select concepts and relationships from a fragment to augment into its ontology. We hypothesised that with this selection process we could reduce the complexity of augmenting a query agent's ontology, compared with a number of standard alternatives. Our investigation shows that a query agent can indeed reduce the cost to acquire concepts that are regularly required, compared with learning nothing. It also reduces the complexity of the query agent's ontology by augmenting it with selected concepts and relationships which are related to its domain. We also hypothesise that to decrease the agent's ontology's complexity further we can discard irrelevant information, and this will be the focus of our future work. Specifically, this focus aims to identify the value of the knowledge contained in an agent's ontology so that it can select which concepts and relationships to 'forget'.

REFERENCES

[1] Bailin, S., Truszkowski, W.: Ontology negotiation between intelligent information agents. The Knowledge Engineering Review **17**(01) (2002) 7–19

[2] 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

[3] 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

[4] Soh, L.: Multiagent, Distributed Ontology Learning. Working Notes of the 2nd AAMAS OAS Workshop, Bologna, Italy (2002)

[5] 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

[6] Haase, P., Stojanovic, L.: Consistent Evolution of OWL Ontologies. Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece (2005)

[7] Packer, H., Gibbins, N., Payne, T., Jennings, N.: Evolving Ontological Knowledge Bases through Agent Collaboration. European Workshop on Mulit-Agent Systems, Bath, UK (2008)

[8] Reynolds, D. and Thompson, C. and Mukerji, J. and Coleman, D.: An assessment of RDF/OWL modelling. Technical Report HPL-2005-189, October 28, 2005

[9] 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

[6] Brain Atlas Domain ontology: `http://twiki.ipaw.info/bin/view/Challenge/OntoGrid`

[7] Bug Reports ontology: `http://www.cs.cmu.edu/ãnupriya/bugs`

[8] Charly Air Service ontology: `http://www.fo-ss.ch/simon/DiplomaThesis/Ontologies/CharlyAirService.owl#`

[9] Food ontology: `http://www.hut.fi/t̃omik/food#`

[10] Pellet: `http://clarkparsia.com/pellet/`