POAF: Portable Ontology Aligned Fragments

Yannis Kalfoglou[1], Paul Smart[1], Dave Braines[2], and Nigel Shadbolt[1]

[1] School of Electronics and Computer Science (ECS), University of Southampton, SO17 1BJ, UK
{y.kalfoglou,ps02v,nrs}@ecs.soton.ac.uk
[2] Emerging Technology Services, IBM United Kingdom Ltd., Hursley Park, Winchester, SO21 2JN, UK
dave_braines@uk.ibm.com

Abstract In today's semantic web, ontology fragmentation and modularization are considered as important tasks due to the size and complexity of prime ontologies. At the same time, the ontology alignment community thrives with solutions for discovering and producing alignments between semantically related concepts. However, these are seldom used in bulk and their subsequent re-use is somewhat problematic. In this paper we set to explore these issues from a practical viewpoint: use ontology alignments to inform an ontology fragmentation strategy for the benefit of exposing and distributing rich ontology aligned fragments.

1 Introduction

Semantic integration and interoperability solutions are becoming increasingly important in dynamic and distributed environments where information originates from various stakeholders. Ontology mapping is seen as one of the core technologies in this space [7] and has gained a lot of attention and momentum in recent years[3]. One of the issues with ontology mapping though, is the relatively sparse and problematic uptake of ontology mapping products: ontology alignments. Despite the advances in describing alignments in rich[4] and standardized ways for programmatic access [3], storing and sharing alignments [9], their use in applications remains sparse.

One of the reasons for this phenomenon is the relatively high computational cost for processing ontology alignments. W3C's omnipresent `owl:sameAs` statement is only a starting point for what it turns out to be a high computational cost and time consuming reasoning task to be undertaken by a DL reasoner. That is because the `owl:sameAs` statement merely reflects the semantic similarity of the two referenced terms, but does not provide any further information about their provenance and semantics. To take full advantage of the semantic similarity more than just the factual semantic similarity between the two terms is needed in an application.

Some proposals to overcome the high computational cost and high latency in performing an inference cycle, is to fragment or modularize the original ontologies. Others are considering enhancing the existing OWL vocabulary with richer

[3] An up-to-date overview of the field is reflected in http://www.ontologymatching.org/
[4] See, for example, the SKOS vocabulary: http://www.w3.org/TR/skos-reference/

constructs that enable more provenance information and semantics to be exposed when semantic similarity is reported. However, these are heavy weight approaches that pose certain assumptions on the domain and applications.

We advocate a practical and ready-to-use with current technology solution to this problem: use existing alignments and OWL taxonomic reasoning to identify fragments from the original ontologies that capture the immediate provenance and semantics of the aligned terms. We then propose to extract those fragments using standard W3C rules and query technology that is easy to reuse and replicate in different scenarios. The extracted fragments are bundled together in self-contained and well defined portable OWL fragments. These can then be accessed and re-used at a lower cost than that of accessing and re-using the original ontologies.

In the next section we elaborate on the requirements for operational ontology fragments drawn from our experience in an international research collaboration programme (section 2). Our proposed Portable Ontology Aligned Fragments (POAF) work is described in section 3 along with a concise example case (section 3.1). We report on related work in section 4 before conclude the paper in section 5.

2 The case for operational ontology fragments

The work described in this paper is part of the International Technology Alliance project Semantic Integration and Collaborative Planning[5]. The overall aim of this project is to enable the integration of heterogeneous and physically distributed information content in semantically-coherent ways. This need arises from the information demands of coalition operations. A key concern in these operations relates to the physical distribution and semantic heterogeneity of relevant information content: physical distribution makes information difficult to search, retrieve and manage, while semantic heterogeneity makes information difficult to integrate and understand.

Semantic integration and interoperability solutions have been studied and applied to a variety of domains [8] but coalition operations pose strict availability and access constraints to knowledge assets. There is a perceived operational time frame associated with each operation and any semantic integration solution that aims to have an impact will have to respect that. In today's semantic integration practice, time constraints and availability of knowledge assets is not a high priority. To bridge this gap, we set to explore the use of advanced semantic integration solutions, like ontology mapping, that complete within an operationally useful time frame.

One way of achieving this, is to fragment the original knowledge assets involved in a semantic integration solution. Our experiences from ontology mapping shows that only a small fraction of the referenced ontologies is used in the produced alignments. That opens the road for an ontology alignment informed

[5] More information available from: http://usukita.org/

fragmentation task that meets the strict operational constraints in a coalition context. In this scenario, the ontology alignments become the focal point, both as enablers of semantic integration solutions as well as triggers for fragmenting the original knowledge assets into meaningful and semantically coherent chunks of knowledge. These semantically coherent fragments will mirror the original ontologies but they are smaller in size and complexity and thus, easier to process in a operationally useful time frame.

In the next section we describe a novel mechanism for extracting those fragments from ontologies using ontology alignments as the trigger.

3 POAF

POAF aims to increase the usability, tracking of provenance, and portability of ontology alignment products (typically a number of `owl:sameAs` statements) among interested parties. It is a post ontology alignment process. We assume that an ontology alignment or mapping tool has been executed and delivered a set of ontology alignments using the W3C's standard notation: `owl:sameAs`, `owl:equivalentClass` and `owl:equivalentProperty`. These notations enable us to indicate semantic similarity of two OWL constructs (ranging from individuals to classes and properties, depending on the type of OWL language used). Automated reasoners, when encounter `owl:sameAs` or similar statements, make use of that information and access the aligned terms in order to complete their reasoning process. However, there are issues with this approach which call for a lighter and more efficient way of sharing aligned terms:

- Availability and access of the aligned OWL ontologies: if the ontologies where the `owl:sameAs` referenced terms belong are not accessible (i.e.: due to network outage, bandwidth restrictions or interference concerns in a battlespace operational context) or not available at the time the reasoner tries to conduct its inference cycle, this could cause a break in the reasoning and consequently bring the inference process to a halt, causing the reasoner to abandon this task (some advanced DL reasoners could resume at another point but the particular inference will not be concluded);
- Unnecessary reasoning steps: when a reasoner visits the ontologies where the aligned terms originate from - depending on the type of reasoning task - it is likely that unnecessary crawling of the OWL ontologies will occur. Even if the task is to simply resolve the name of the aligned concept, visiting the originating ontologies will add unnecessary time to the to the processing task;
- Fragmented and distributed factual knowledge base: When a reasoner tries to perform a task where multiple `owl:sameAs` statements are involved and point to a number of different ontologies, the reasoner will have to collate information from different URI addresses. Although today's DL reasoners can cope with this task, it is an unnecessary resource load for the system and could affect its performance;

– Difficult to inspect and track provenance information: When a number of `owl:sameAs` statements are used to convey semantic similarity information for the aligned concepts, it is difficult to track their provenance. This is becoming increasingly difficult as more ontologies are involved. At least is not feasible to inspect the origin of the aligned concepts by using eyeball checking on a casual fashion. It is likely that an engineer will need to employ a reasoner to do this task which brings us back to the problems mentioned before: that of unnecessary reasoning steps, overload of time and bandwidth resources and fragmentation of knowledge base.

Based on these observations, and our experiences with designing, developing, deploying and using ontology mapping systems [5,6], we propose a lighter and more portable way for sharing ontology alignment information. We propose a mechanised way for extracting fragments from the underlying ontologies using ontology alignment information as a trigger. We dubbed those, POAF (Portable Ontology Aligned Fragments) to highlight the tight coupling of the ontology alignment and the generated fragment. As the name dictates, we are interested in fragments of ontologies and in particular, we aim to make those portable. By that we mean:

1. Capture a fragment of the ontology that is directly relevant to the aligned term;
2. Use the aligned term and OWL semantics, to identify and capture those fragments;
3. Bundle these fragments in OWL compatible snippets of code that reasoners can use automatically as if they were "mini-ontologies";
4. Include as much provenance information as possible in these fragments, so that tracking and tracing of the original ontology is feasible.

We implemented these steps in a process to operationalise the idea of POAF. We depict this process in the figure below:

The first step of the POAF generation linear process involves the *capture fragment* task: we use OWL semantics and the ontology subsumption relations to identify related fragments. These are fed into the second step, *Use W3C technology to extract fragment* where the actual extraction occurs. We opted for W3C technology so that we increase our interoperability potential with other tools and technologies. We use SPARQL queries, SWRL rules and HP's Jena Framework. The next step is to *Bundle Fragment*. The aim of this task is to construct well formed OWL fragments so that DL reasoners and external tools can use them as a substitute for the original ontologies, depending on the task and scope of application. Finally, the last step is to provide as much provenance information as possible in the *Provenance aggregator* task. We collate the original namespaces with POAF specific ones to distinguish between declared and inferred statements in the POAF files.

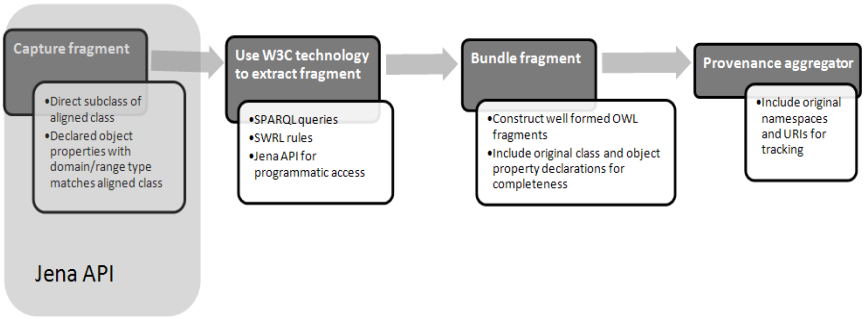

Figure 1. The POAF process.

3.1 An example case

POAF, albeit being at an early stage of development, has been deployed in experimental scenarios in the context of the "Semantic Integration and Collaborative Planning" project of the International Technology Alliance programme. In one of these scenarios, we had to align two OWL ontologies: `terrorism.owl`[6] and `tkb.owl`[7] for the sake of enabling interoperable task between agents using these two domain ontologies. We used an ontology mapping tool to discover potential alignments between the two ontologies. Although this step is not part of POAF, for the sake of completeness we provide brief information on the process: we chose to work with CMS (CROSI Mapping System), an open source comprehensive ontology mapping system[8] . We include in the screenshot below, small fragments of the original ontologies and one of the proposed alignments, as delivered by CMS in `owl:sameAs` format:

We set the CMS system to the following matching criteria: use 1 matcher (`StructurePlus`, an algorithm that uses ontology structure information along with linguistic features), no weight tuning up (not applicable in our case as we used only 1 matcher), output the top 20% of results only and use OWL format for the output[9] .

[6] This is an OWL ontology that describes concepts in the terrorism domain, available from http://counterterror.mindswap.org/2005/terrorism.owl

[7] this is an OWL ontology in the terrorism domain that describes main concepts and facts on terrorism attacks. Available on request.

[8] available from http://sourceforge.net/projects/ontologymapping/

[9] more on the CMS system and the alignment algorithms it uses can be found here: http://www.aktors.org/crosi/

Fragment of terrorism.owl:

```
<owl:Class rdf:about="#City">
    <rdfs:label>City</rdfs:label>
    <rdfs:label>In city</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Location"/>
  </owl:Class>
```

Fragment of tkb.owl:

```
<owl:Class rdf:about="#City">
    <owl:disjointWith>
      <owl:Class rdf:about="#Information_Source"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="Tactic"/>
    </owl:disjointWith>
    <owl:disjointWith>
…
  </owl:Class>
```

```
<rdf:Description rdf:resource="http://counterterror.mindswap.org/2005/terrorism.owl#City">
  <owl:sameAs rdf:resource="http://www.ecs.soton.ac.uk/ita/work/tkb.owl#City"/>
</rdf:Description>
```

Figure 2. Fragments of `terrorism` and `tkb` ontologies and one of the proposed alignments as found by CMS tool.

Going back to the steps outlined before, we now need to "*Capture a fragment of the ontology that is directly relevant to the aligned term*". We chose the following OWL constructs to be directly relevant to any aligned term: subclasses, data and object properties. We identify subclasses by default taxonomic reasoning whereas for object and data type properties we take into account the domain and range type of the property. If the value of the `rdfs:domain` and `rdfs:range` matches the aligned class, then we earmark this property from inclusion in the POAF structure. This basic information is deemed to be enough to capture the direct provenance of the aligned term and eventually will constitute a POAF for that term. Note, that we do not aim to capture a comprehensive fragment that can act as a substitute for an entire ontology. In that sense, POAF structures should only be used as *elaborate summaries* of aligned fragments that include their directly related constructs (immediate subclass and their properties). We chose not to include immediate superclass, if exist, due to the intuitive denotational distance: superclasses are, in general, more abstract and that could introduce semantic discrepancies in the POAF structure. On the other hand, a direct subclass is a safer option as subclasses in general are more specific and, intuitively speaking, we expect them to be semantically closer to the aligned class.

After we decided what to include in a POAF structure, the next step outlined before is to: "*Use the aligned term, and OWL semantics, to identify and capture those fragments*". At this stage we had several options for extracting fragments from an ontology. As our work is not directly targeting ontology modularization, per se, but rather an informed and selective selection of ontology constructs, we decided to stray away from the popular graph traversal techniques used by ontology modularization practitioners [1]. Rather, we focussed on meeting certain operational requirements:

– Be semantic web compatible; by that we mean to use semantic web standards (W3C endorsed), mainly OWL, RDF and SPARQL;

– Use automatic inference, whenever possible, to identify the fragments. This is achieved by using OWL inference, especially when we want to extract taxonomic information;
– Use easily editable automated procedures that enable re-use and portability. We see rules technology as the best candidate here, as rules can be edited, re-defined and re-used in different settings. Also, W3C already backs certain semantic web rules technology: SWRL and RIF, most notably. Another option here, is to use query technology, especially SPARQL, as it meets the criteria set for rules (editable, re-usable) and enjoys the strong backing from W3C standards.

To satisfy these requirements, we opted for a solution where we use a combination of SWRL rules and SPARQL queries to identify and capture the fragments. We decided to provide both as SWRL captures the logic for identifying the fragment, but its executability is cumbersome with only a handful of theorem provers[10] and closed environments offering SWRL reasoning[11]; SPARQL on the other hand, provides the functionality we are after in query format using the CONSTRUCT clause. In addition to this, SPARQL queries are easily re-usable at different settings.

To illustrate the result of our "*capturing the fragments*" task, we include below a screenshot of the eligible constructs from the original ontologies:

As we can see, both `terrorism.owl:City` and `tbl.owl:City` have object properties declared: `locatedInCity` for `terrorism.owl:City` and `incidentOccuredHere` for `tbl.owl:City`. Also, `terrorism.owl:City` has two subclasses: `Town` and `Municipality` whereas `tkb.owl:City` has `CapitalCity` as a subclass. Note that these are only a fraction of the original object property and subclass declarations due to space limitations. This is the basic information (subclasses, properties) we aim to capture to populate the POAF structure for this alignment.

We do that by using SWRL rules and SPARQL CONSTRUCT queries. For example, to capture object property information we use the SWRL rule depicted in figure 4, which can also be expressed as a SPARQL CONSTRUCT query, depicted in figure 5.

Note that the query above captures only the `rdfs:domain` information of an object property statement; a similar query is used for `rdfs:range` statements.

The next step in our process it to: "*Bundle these fragments in OWL compatible snippets of code*". To do that we simply execute the SWRL rule over the original ontologies or fire the SPARQL query via a custom built endpoint. This will capture the information we want to include in the POAF structure (subclass, properties). The results are depicted in figure 6.

As we can see, class `Municipality` from `terrorism.owl` has been associated with class `City` from `tkb.owl`. Similarly, class `Town` from `terrorism.owl` has been associated with `City` from `tkb.owl`. On the other end, class `Capital_City` from `tkb.owl` has been associated with class `City` from `terrorism.owl`. Similar correspondences between object properties have been produced: `IncidentOccuredHere`

[10] like the Hoolet theorem prover: http://owl.man.ac.uk/hoolet/
[11] like the Protege SWRLTab: http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

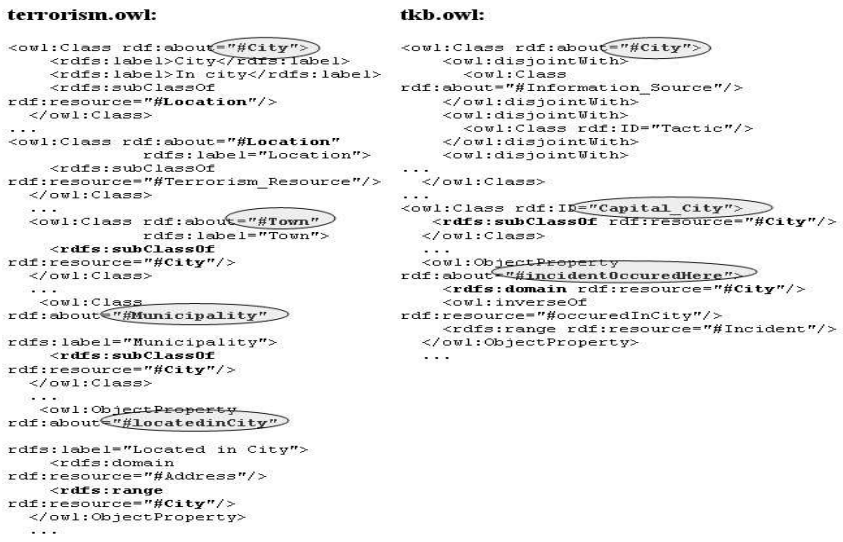

Figure 3. Eligible constructs for extraction.

object property from `tkb.owl` is now associated with class `City` from `terrorism.owl` and `locatedinCity` from `terrorism.owl` is associated with class `City` from `tkb.owl`. Note that all the original declared associations of subclass and object properties are retained for semantic consistency. These associations are driven by the ontology mapping results of `terrorism.owl:City` being declared as `owl:sameAs` to `tkb.owl:City`, and it is legitimate to assume that they share the same properties and subclasses. The above depicted POAF structure merely reflects this and we built it automatically.

The final step in the POAF process is to: "*include as much provenance information as possible in these fragments, so that tracking and tracing of the original ontology is feasible.*". This is essentially a safeguarding mechanism against unwarranted inferences from reasoners attempting to parse the POAF structure. Currently, we use the standard `rdfs:subclassOf` construct to denote subclass relationship. But we also store the POAF inferred subclass relationships with a designated `poaf:inferred` construct to distinguish between declared and inferred information. This will ease tracking and evaluation of POAF structures. We are also planning to collect and package all the original namespaces in a designated construct so that we facilitate tracing provenance for bulk POAF structures processing.

```
<swrl:variable rdf:ID="x"/>
<swrl:variable rdf:ID="y"/>
<swrl:variable rdf:ID="z"/>
<swrl:Imp>
   <swrl:body rdf:parseType="Collection">
         <swrl:IndividualPropertyAtom>
               <swrl:propertyPredicate rdf:resource="&usukitacs;hasObjProperty"/>
                  <swrl:argument1 rdf:resource="#y"/>
                  <swrl:argument2 rdf:resource="#z"/>
         </swrl:IndividualPropertyAtom>
         <swrl:IndividualPropertyAtom>
               <swrl:propertyPredicate rdf:resource="&usukitacs;sameAs"/>
                  <swrl:argument1 rdf:resource="#x"/>
                  <swrl:argument2 rdf:resource="#y"/>
         </swrl:IndividualPropertyAtom>
   </swrl:body>
   <swrl:head rdf:parseType="Collection">
         <swrl:IndividualPropertyAtom>
              <swrl:propertyPredicate rdf:resource="&usukitacs;hasObjProperty"/>
                 <swrl:argument1 rdf:resource="#x"/>
                 <swrl:argument2 rdf:resource="#z"/>
          </swrl:IndividualPropertyAtom>
   </swrl:head>
</swrl:Imp>
```

Figure 4. Capturing object property information using SWRL.

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>

CONSTRUCT { ?x rdf:type owl:Class .
                  ?z rdf:type owl:ObjectProperty .
                  ?z rdfs:domain ?x
                  }
WHERE { ?x owl:sameAs ?y .
            ?z rdf:type owl:ObjectProperty .
            ?z rdfs:domain ?y
            }
```

Figure 5. Capturing and propagating object property information using SPARQL CONSTRUCT.

3.2 Extensions

The current setting in the POAF work can be enhanced with the provision of a customized front for selecting the ontology constructs to be extracted and included in a POAF structure. For example, our default option is to extract the direct subclass of the aligned class. However, depending on the application scenario and inference needed, one might need to extract all the subclasses of the aligned class or even some or all of the super-classes. This sort of custom tuning of the taxonomy hierarchy is feasible with advance Semantic Web APIs, like HP's Jena Framework[12]. Using a customized extraction mechanism we should be able to extract any kind of ontology construct that is somehow related with the

[12] Available from: http://jena.sourceforge.net/

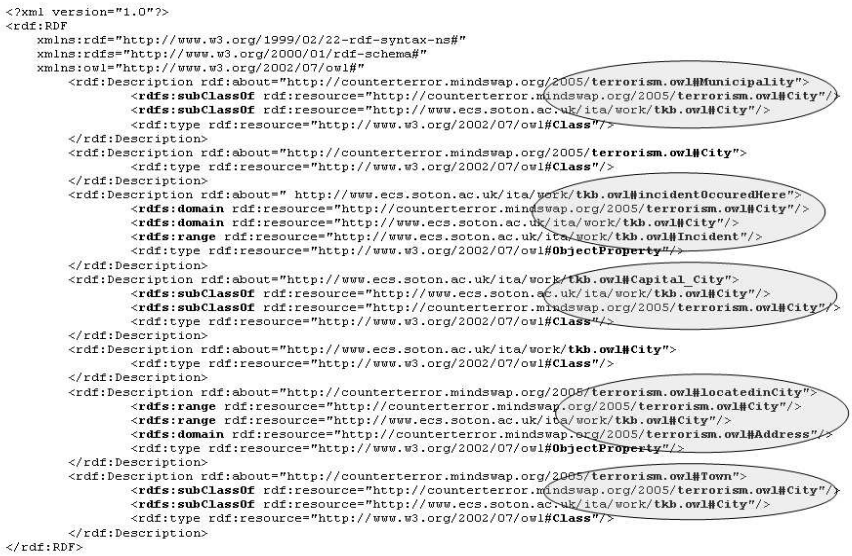

Figure 6. A typical POAF fragment.

aligned class and goes beyond the direct subclass and object/data type property as is the default setting.

We are planning to work on a meta-OWL extension for expressing subclass information that is inferred by the POAF structure. Currently, we simply store such information with a designated construct (`poaf:inferred`). But eventually, we would like to replace the standardized `rdfs:subclassOf` with an equivalent construct to distinguish between declared and inferred information. Any extension of that sort will have to be OWL compatible though and parseable by standard DL reasoners.

Finally, we also plan to work on the SPARQL front by distributing the generated SPARQL CONSTRUCT queries which encapsulate both the fragment extraction logic but also the original semantic similarity information. Currently, we use those in a sandbox application where the POAF program initiates the process. In a distributed scenario though, we want to achieve the same functionality with a series of SPARQL CONSTRUCT clauses that capture the desired information, and build the POAF structure.

4 Related work

POAF is in principle an ontology fragmentation method. However it is distinctly different from typical fragmentation techniques in that it makes use of a strong pre-requisite condition for enacting fragmentation: the availability of ontology

alignments. Another differentiating factor is that POAF aims to increase the potential of sharing, using and distributing ontology alignments across multiple systems in a dynamic networked environment. The ontology fragmentation aspect is a side effect of this functionality and we do not intend to use it as its primary function. Taking into account this distinct characteristics of POAF we identify the following related research: Euzenat and colleagues use the notion of ontology alignments to inform the syntax for expressing ontology modules [4]. Their main aim is use ontology alignments in order to take advantage of the alignment composition features embedded in them. This allows them to express a syntax for ontology modules in a consistent way. They argue that the modules can replace ontologies where they are used. It appears that the role of the alignments is to enable them to include related content in the module, also used to enhance the syntax. This is different from our use of ontology alignment which is to identify the starting points in our fragmentation algorithm which employs standard taxonomic reasoning.

Similarly, taxonomic reasoning is deployed in [2]. The authors propose a modularization algorithm and a set of requirements for ontology modularization. The algorithm takes advantages of the subsumption checking of DLs to identify self-contained pieces in the given ontology.

Others have argued that relaxing the subsumption dependencies between ontology constructs can improve the expressivity of modular languages [1]. They argued for a new modular semantics or extensions to improve the functionality of `owl:import`.

5 Conclusions

In this paper we proposed a light weight mechanism for sharing and distributing enhanced ontology alignment information. We make use of the generated alignments between two ontologies and we expose their immediately related terms in a well defined and concise OWL structure, which we dubbed Portable Ontology Alignment Fragments (POAF). Our work taps on the area of ontology fragmentation, but not directly related to a fragmentation strategy. Rather, we aim to highlight the importance of distributing more information when reporting semantic similarity between two aligned terms. This information could enhance uptake and re-use of ontology alignments in scenarios not easily foreseen today. We also contribute to adoption of semantic web technologies in scenarios where operational time constraints and resource load are important requirements. A POAF based solutions aims to reduce the time needed to process ontologically aligned structures as it encapsulates the aligned concepts and their immediately related concepts in small and easily manageable fragments.

POAF can also be used as a quality assessment tool for ontology alignment. Since the generated POAF structure exposes the related terms of the aligned terms (subclass and properties), this information can be used to semantically sanitize and proof-check the proposed alignment. This process could be a manual or a semi-automated one depending on the assessment task.

Acknowledgements

This research was sponsored by the US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. J. Bao and V. Honavar. Divide and conquer semantic web with modular ontologies - a brief review of modular ontology language formalisms. In *Proceedings of the ISWC'06 International Workshop on Modular Ontologies (WoMo'06), Atlanta, GA, USA*, nov 2006.
2. M. d'Aquin, M. Sabou, and E. Motta. Modularization: a key for the dynamic selection of relevant knowledge components. In *Proceedings of the ISWC'06 International Workshop on Modular Ontologies (WoMo'06), Atlanta, GA, USA*, nov 2006.
3. J. Euzenat. An API for ontology alignment. In *Proceedings of the 3rd International Semantic Web Confernece (ISWC'04), LNCS 3298, Hiroshima, Japan*, pages 698–712, Nov. 2004.
4. J. Euzenat, A. Zimmermann, and F. Freitas. Alignment-based modules for encapsulating ontologies. In *Proceedings of the K-Cap'07 International Workshop on Modular Ontologies (WoMo'07), Whistler, BC, Canada*, oct 2007.
5. Y. Kalfoglou and B. Hu. CMS: CROSI Mapping System - results of the 2005 ontology alignment contest. In *Proceedings of the K-Cap'05 Integrating Ontologies workshop, Banff, Canada*, pages 77–84, Oct. 2005.
6. Y. Kalfoglou and M. Schorlemmer. IF-Map: an ontology mapping method based on Information Flow theory. *Journal on Data Semantics*, 1:98–127, Oct. 2003. LNCS2800, Springer, ISBN: 3-540-20407-5.
7. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.
8. Y. Kalfoglou, M. Schorlemmer, M. Uschold, A. Sheth, and S. Staab. Semantic interoperability and integration. Seminar 04391 - executive summary, Schloss Dagstuhl - International Conference and Research Centre, Sept. 2004.
9. R. Palma, P. Haase, and A. Gomez-Perez. OYSTER: sharing and re-using ontologies in a peer-to-peer community. In *Proceedings of the 15th International World Wide Web Conference (WWW'06), Edinburgh, UK*, may 2006.